POLITECNICO DI TORINO

SCUOLA DI DOTTORATO

Dottorato in Matematica per le Scienze dell'Ingegneria - XV Ciclo

Settore disciplinare: MAT/05 ANALISI MATEMATICA

Tesi di Dottorato

Localization and optimization problems for camera networks

Domenica Borra

Tutore

Prof. Fabio Fagnani

Coordinatore del corso di Dottorato

Prof. Lamberto Rondoni

Marzo 2013

Localization and optimization problems for camera networks

Domenica Borra

Submitted for the degree of Doctor of Philosophy

POLITO - Politecnico di Torino

An equation means nothing to me
unless it expresses a thought of God.
— Srinivasa Ramanujan

To whom believed in me

Acknowledgements

During my Ph.D I have benefited from the help of many people. My first and deepest thank goes to my supervisor Prof. Fabio Fagnani. His guidance, wisdom, pragmatism and patience were essential in these three years.

He has always been forward-looking during my studies and gave me the chance to travel and experience different scientific groups, from the Automatica Department (DEI) in Padova, to UCSB, and useful interesting summer schools and international conferences.

My gratitude also goes to Prof. Sandro Zampieri, that allowed me to experience the inspiring scientific life at the Department of Information Engineering in Padova, giving me many inputs and valuable teaching, being a guide in my research, together with my supervisor Fabio Fagnani. Their direct and wise advice has been precious for me.

I also want to thank Prof. Francesco Bullo, who permitted me to live an enriching experience at UCSB (Santa Barbara, CA), supporting my work, giving me his revealing advise and encouraging me. This experience made me discover a different approach to scientific research, and made my whole Ph.D. an unforgettable experience.

I am grateful to Professors Fabio Fagnani and Luciano Pandolfi for the teaching assistant opportunities they gave to me. This academic didactic experience gave me the chance to face 200 freshmen each year of my PhD course, and it has been formative and enriching to teach them a tiny part of “The Mathematics”.

Several researchers helped me and gave rise to compelling collaborations during my Ph.D., among them I want to thank Sophie Fosson, Enrico Lovisari, Ruggero Carli, Tommaso Lorenzi, and Fabio Pasqualetti. I wish to thank also my collegues PhD students, and the entire research group in our department, thanks Wilbert Samuel Rossi, Paolo Frasca, Chiara Ravazzi, for our scientific discussions and nice coffee breaks and lunches together.

My thanks for my family will never be enough, they always supported, took care and encouraged me. This achievement, and many past other ones, would have never been possible without them by my side.

I also deeply thank my friends, the old and the new ones, for the countless cheerful moments shared together, that made me enjoy these years during my studies. They shared with me up and down moments, always giving me advice, supporting me and my decisions.

Last but not least, I deeply thank Simone that has always been by my side, I thank him for the huge patience, fundamental comforting words and advice, for his belief in me; he has been a bedrock during these years.

Torino, March 2013 D. B.

Ringraziamenti

Durante il Dottorato ho beneficiato dell'aiuto di molte persone. Il primo e piú profondo grazie va al mio relatore, il Prof. Fabio Fagnani.

La sua guida, saggezza, pragmatismo e pazienza sono state essenziali in questi tre anni. É sempre stato lungimirante durante i miei studi, e mi ha dato la possibilitá di viaggiare e sperimentare diversi gruppi scientifici, dal Dipartimento di Automatica (DEI) a Padova, al gruppo di robotica a UCSB, incluse utili e interessanti scuole estive e conferenze internazionali.

La mia gratitudine va anche al Prof. Sandro Zampieri, che mi ha permesso di vivere la stimolante atmosfera scientifica al DEI a Padova, dandomi molti spunti e importanti insegnamenti, rimanendo una guida nella mia ricerca, insieme a Fabio Fagnani. La loro supervisione diretta e saggia é stata preziosa per me.

Vorrei anche ringrazionare il Prof. Francesco Bullo, che mi ha dato la possibilitá di avere una ricca esperianza a UCSB (Santa Barbara, CA), sostenendo il mio lavoro, dandomi consigli rivelatori e incoraggiandomi. Questa esperienza mi ha fatto scoprire un diverso approccio alla ricerca scientifica, e ha reso il percorso come dottoranda un'esperienza indimenticabile.

Sono grata ai Prof. Fabio Fagnani e Luciano Pandolfi per le opportunitá didattiche che mi hanno dato. Svolgere le esercitazioni di Analisi Matematica I per tutti e tre gli anni, avere a che fare con 200 studenti ogni anno diversi é stato formativo e arricchente, in particolare ho avuto il piacere di insegnare loro una piccola parte della "Matematica".

Molti ricercatori mi hanno aiutato e hanno dato luogo alla nascita di stimolanti collaborazioni, tra tutti voglio ringraziare Sophie Fosson, Enrico Lovisari, Ruggero Carli, Tommaso Lorenzi, e Fabio Pasqualetti.

Vorrei anche ringraziare i miei colleghi dottorandi, e l'intero gruppo di ricerca nel Dipartimento di Matematica al Politecnico di Torino, grazie a Wilbert Samuel Rossi, Paolo Frasca, Chiara Ravazzi per le pause caffé e i pranzi insieme.

I ringraziamenti alla mia famiglia non saranno mai abbastanza, mi hanno sempre sostenuto, prendendosi cura di me e incorraggiandomi. Questo traguardo, come tutti i precedenti, non sarebbe mai stato possibile senza la mia famiglia a fianco.

Ringrazio sentitamente anche i miei amici, i vecchi e i nuovi, per gli innumerevoli momenti di svago insieme, mi hanno fatto vivere bene questi anni durante i miei studi. Essi hanno condiviso con me sia i momenti negativi sia quelli positivi, dandomi consigli e sostegno nelle mie decisioni.

Infine, ultimo ma non per questo meno importante, é il mio grazie a Simone, che é sempre stato al mio fianco. Lo ringrazio per l'enorme pazienza, le fondamentali parole di conforto e guida, per la fiducia che riposto in me; é stato la mia roccia durante questi anni.

Torino, Marzo 2013 D. B.

Abstract

In the last years much effort has been devoted by the scientific community to develop leaderless distributed strategies for solving problems involving interacting agents which need to achieve a common goal. The advantages of these strategies, if compared with the centralized ones, are in terms of robustness of the resulting systems with respect to communication and/or agent failures, and in terms of adaptivity to environmental changes and simplicity in the system tuning. In other terms, they do not require the presence of a central unit that has to gather all the information from the network and process huge amounts of data, dealing with reliability of multi-hops, reliability of the agents themselves, both from a security point of view and a physical one. On the other hand, distributed strategies, based on local exchange of information among peer agents, can be more difficult to design and optimize and often exhibit slower convergence to the regime working conditions. We focus on relative localization in sensor networks, investigating how the error due to noisy data propagates through the network, in terms of the relative error on each component of the optimal estimator of the position vector. The relative error is computed as a function of the eigenvalues of the network using the DFT, and these tools are useful especially for the exemplary class of networks called the Abelian Cayley networks. The role of the network topology and dimension has a leading importance in the error characterization, that is investigated in this work. In this framework of networked control systems, we focus on networks of autonomous cameras, in particular we focus on Pan-Tilt-Zoom cameras (PTZ). A large set of cameras communicating each other through a network is a widely used architecture in application areas like video surveillance, tracking and motion capture Aghajan and Cavallaro [2009]. In a network of cameras one of the most crucial problems is calibration. For each camera this consists in understanding what is its position and orientation with respect to a global common reference frame. The importance of this information is for instance if the camera network is used to track an external mobile object, since neighboring cameras have to communicate whenever the target goes from one *field of view* (f.o.v.) to another. Our aim is to propose an algorithm that makes the cameras

complete this task autonomously, in a distributed fashion. This also allows the possibility to re–calibrate periodically, especially if we deal with mobile cameras. The calibration problem over Euclidean spaces has recently been studied by Barooah and Hespanha in a great detail (see Barooah et al. [2006], Barooah and Hespanha [2007]). They achieve the network localization using noisy relative measurements. Well–known methods in computer vision permit to obtain quite easily and efficiently relative positions and orientations of pairs of cameras whose sensing regions overlap. Then our problem is to determine, from these noisy input data, the position and the orientation of the cameras with respect to a common reference coordinate system. Cameras calibration can be casted into an optimization problem (or a consensus) over the manifold $SE(3)$. We decouple the latter in the estimation of the position and the orientation, and we focus on the orientation calibration over the manifold $SO(2)$, under mild assumptions. In Sarlette and Sepulchre [2009b], Tron and Vidal [2009b] consensus algorithms on the manifolds $SO(2)$ and $SE(3)$ based on the gradient flow of a potential defined using the chordal or geodesic distance are studied. The drawback is that the proposed potentials exhibit a great number of local minima.

Our first contribution is the design of a synchronous calibration algorithm, based on a non-convex optimization problem. The set of available relative measurements is described by a graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, where nodes represent cameras, and edges represent the available relative orientation measurements. We break the estimation problem into two parts: first we estimate a combinatorial object, which is a set of integers, each associated with an edge in $\mathscr{E}$. Intuitively, these integers take care of the fact that measurement noises along the cycles in the graph in general do not sum to 0. Then, the original optimization problem over a manifold can be reduced to a quadratic optimization problem, which can be easily solved using classic algorithms. The idea of using cycles has already been proposed in Russell et al. [2010], Piovan et al. [2011b]. In fact, we propose two versions of the algorithm, their difference lies in the choice of the set of cycles, that can be minimal or associated with spanning trees. We gain consistency, since the solution given by the algorithm coincides with the true one, if there is no noise. A worst–case analysis of its performance and numerical experiments are provided, comparing the the two different versions according to the chosen cycle basis. This research has appeared in Borra et al. [2012a, a].

Our second contribution concerns the problem of calibrating a network of cameras with a completely distributed and asynchronous random algorithm. We assume the communication protocol to be random gossip-like, in which at each iteration only one link is updated gathering only the states of neighboring cameras. In the previous setup, we propose an asynchronous gossip version of the algorithm proposed in Piovan et al. [2011b]. The leading idea is to obtain a new set of relative orientations which is ensured to sum up to multiples of 2π over a chosen family of cycles. The new set of relative orientations is then spread along a spanning tree to obtain an estimate of the orientations. The algorithm is proved to converge in the mean square sense for general graphs, with null cycle-error almost surely. If we focus on ring graphs, the proposed algorithm converges for any realization, and the expected value of the limit random variable equals the optimal solution, written in closed form. We also characterize the convergence speed proving that the estimate approaches the limit values exponentially fast. Numerical experiments validating our analysis and investigating non-planar topologies are provided.

Our third and final contribution refers to the design of surveillance trajectories for a network of autonomous calibrated cameras to detect intruders in an environment. Remote surveillance of human activities for civil and military applications is receiving considerable attention from the research, and one of the main challenges consists of developing efficient algorithms for the cameras to autonomously and distributively complete tracking, surveillance, and recognition tasks. Given a network of PTZ cameras installed at important locations, we assume the cameras to move their f.o.v. (subject to physical constraints) to cooperatively self-organize in order to detect *intruders* in the environment, that appear at arbitrary locations and times. We consider both *static* intruders and *dynamic* intruders, and our performance criteria is the *worst-case detection time*, namely the longest time needed for the network to detect intruders. Works related to our camera surveillance problem can be found in the mobile robotics and computer science literatures, see e.g. Alberton et al. [2012], Pasqualetti et al. [2011b]). In this context, the perimeter patrolling problem has recently been studied in Baseggio et al. [2010], Carli et al. [2011], Spindler et al. [2012], and we extend their results to the case of general topologies. Our graph partitioning problem differs from classical setups where the problem is combinatorial and discrete, whereas we formulate continuous graph partitioning problems, in which the partition is obtained by splitting the edges. Our results are applicable to different problems, including dynamic load balancing for multiprocessor networks. We show that our continuous graph partitioning problem is convex and non-differentiable, and we characterize its solutions. Then, we derive an equivalent convex and differentiable partitioning problem, which is amenable to distributed implementation. We exhaustively discuss any possible optimal cameras trajectory against static intruders, showing that for tree and ring roadmaps, it is equivalent to solve a continuous graph partitioning problem. For general cyclic roadmaps, our trajectories based on continuous partitions are proved to be optimal up to a constant factor. For the case of dynamic intruders, we derive a necessary and sufficient condition on the cameras locations for the existence of a trajectory with finite detection time. We design optimal cameras trajectories up to constant factors, for ring and tree roadmaps. Finally, we consider three different communication models (synchronous, asymmetric broadcast, and gossip models), and we propose distributed algorithms for the cameras to partition the graph in all these scenarios. The convergence analysis and a simulation study of these algorithms are provided. These results appeared in Borra et al. [2012b, b].

In short. In the framework of networked control systems, we focus on networks of autonomous PTZ cameras. A large set of cameras communicating each other through a network is a widely used architecture in application areas like video surveillance, tracking and motion. First, we consider relative localization in sensor networks, and we tackle the issue of investigating the error propagation, in terms of the mean error on each component of the optimal estimator of the position vector. The relative error is computed as a function of the eigenvalues of the network: using this formula and focusing on an exemplary class of networks (the Abelian Cayley networks), we study the role of the network topology and the dimension of the networks in the error characterization. Second, in a network of cameras one of the most crucial problems is calibration. For each camera this consists in understanding what is its position and orientation with respect to a global common reference frame. Well-known methods in computer vision permit to obtain relative positions and orientations of pairs of cameras whose sensing regions overlap. The aim is to propose an algorithm that, from these noisy input data makes the cameras complete the calibration task autonomously, in a distributed fashion. We focus on the planar case, formulating an optimization problem over the manifold $SO(2)$. We propose synchronous deterministic and distributed algorithms that calibrate planar networks exploiting the cycle structure of the underlying communication graph. Performance analysis and numerical experiments are shown. Third, we propose a gossip-like randomized calibration algorithm, whose probabilistic convergence and numerical studies are provided. Forth and finally, we design surveillance trajectories for a network of calibrated autonomous cameras to detect intruders in an environment, through a continuous graph partitioning problem.

Keywords. Distributed control, Camera network, Graph Theory, Communication protocols, Decentralized systems, Graph partitioning, Convex constrained optimization.

Contents

Acknowledgements **v**

Abstract **ix**

List of figures **xvii**

1 Introduction **1**

1.1 Literature synopsis 1

1.1.1 Distributed estimation on graphs from relative measurements 3

1.1.2 Distributed calibration 5

1.1.3 Patrolling problem and graph partitions 6

1.2 Contributions of this thesis 7

1.3 Publications 9

2 Mathematical tools **11**

2.1 Graph theory preliminaries 11

2.1.1 Cycle structure of a graph 12

2.1.2 Cayley graphs 16

2.2 Linear algebra tools 16

2.3 Overview on Markov Chains 19

2.3.1 Basic definitions 19

2.3.2 The electrical analogy 22

2.4 Discrete Fourier transform over finite Abelian groups 24

3 Error propagation for relative localization over geometric networks **27**

3.1 Introduction 27

3.2 Relative vector localization 29

3.3 Propagation of errors 31

3.3.1 1-D thorus graph 35

3.3.2 2-D thorus graph 37

3.4 Conclusions 41

4 Synchronous distributed calibration algorithms 43

4.1 Introduction 43

4.2 Problem Formulation 46

4.3 Cameras calibration 48

4.4 Description of the proposed algorithm 52

4.4.1 The regions of convexity of the cost functions 52

4.4.2 Estimation of the convexity region 53

4.4.3 Performance analysis of the proposed algorithm 55

4.5 Distributed algorithms for rotational calibration 57

4.5.1 The Tree-algorithm 59

4.5.2 Minimal cycles-algorithm 59

4.6 Resilience against measurement noise for different graph topologies 62

4.7 Numerical results 63

4.8 Conclusions 66

5 Asynchronous distributed calibration algorithm 69

5.1 Introduction 69

5.2 Problem setup 72

5.3 Description of the proposed algorithm 74

5.4 Perfomance analysis 75

5.4.1 General graphs 76

5.4.2 Ring graphs 82

5.5 Numerical examples 83

5.5.1 Experiments on planar graphs 83

5.5.2 Experiments on Cayley graphs 85

5.6 Conclusions and further work 87

6 Graph partitioning for camera networks surveillance 89

6.1 Introduction 89

6.2 Continuous Partitions of Weighted Graphs 92

6.3 Setup for Camera Surveillance 95

6.3.1 Problem Setup . . . 96
6.3.2 Cameras trajectory . . . 96
6.3.3 Performance criteria . . . 98
6.4 Camera Trajectory for Static Intruders . . . 98
6.4.1 Main results for static intruders . . . 99
6.4.2 Proof of Theorem 6.7 . . . 99
6.5 Cameras Trajectories for Dynamic Intruders . . . 102
6.5.1 Ring roadmap . . . 104
6.5.2 Tree roadmap . . . 106
6.6 Distributed Partitioning Algorithms . . . 107
6.6.1 Synchronous Gradient Partitioning algorithm . . . 108
6.6.2 Asymmetric Broadcast Partitioning algorithm . . . 109
6.6.3 Symmetric Gossip partitioning algorithm . . . 110
6.7 Conclusion . . . 111

7 Conclusion and future work **113**
7.1 Summary . . . 113
7.2 Directions for future research . . . 115

Bibliography **124**

List of Figures

3.1 The figure shows different d-tori, one dimensional (left) and 2-dimensional (center) respectively. In the right figure, we focus on the 2D Cayley grid, and we fix $(\boldsymbol{i},\boldsymbol{j}) = ((0,0),(1,0)) \in \mathscr{E}$. In order to compute the quantity in Eq. (3.12), we consider two different kind of edges, horizontal $e^o(t_1,t_2) = ((t_1,t_2),(t_1+1,t_2)) \in \mathbb{Z}_N^2$, and vertical $e^v(t_1,t_2) = ((t_1,t_2),(t_1,t_2+1))$, where the free parameters are $t_1,t_2 \in \mathbb{Z}$. 38

3.2 3D plot and contour lines of the two variable function f, with $\boldsymbol{t} = (1,4)$ (left), and $\boldsymbol{t} = (1,4)$ (right). It is clear that t_1 and t_2 are the frequency of oscillations along the x-axis and y-axis respectively. 41

3.3 The Figure (left) show how the quantity $\sum_{\boldsymbol{h}\neq\boldsymbol{0}} |C^o_{\boldsymbol{h},\boldsymbol{t}}|/N^2$ goes to infinity as N asymptotically increases in a 2D Cayley grid. As opposite, in right plot it is shown the asymptotic behavior of $x_{(0,0)} - x_{(1,0)}$ as defined in Eq. (3.13), setting the maximum noise $\bar{\epsilon} = 1$. Note that, as N is asymptotically large, the considered quantity remains bounded. 42

4.1 A simple ring with 3 agents. On the right panel, the three regions in which $[-\pi,\pi)$ is partitioned, with the contour lines of the quadratic functions in (4.5). 48

4.2 Description of a network of cameras satisfying the assumptions. It is shown the external reference frame Σ_0 and, for each camera, the “natural” local reference frame Σ_v' (in dotted lines) and the chosen local reference frame Σ_v (in solid lines). 50

4.3 A simple graph to show how the second algorithm works. 62

4.4 Two examples of spanning trees for a line–like graph. The proposed algorithms work in a similar manner for the one on the right, while the *Minimal cycles-algorithm* is far more effective for the one on the left. 62

4.5 On the left a ring graph, for which the two algorithms have the same performance. On the right, a grid graph. 63

4.6 On the left a square grid graph for $n = 4$. On the right the correspondent spanning tree used in simulations. 64

4.7 Average error on the orientations (modulo 2π) and error on $\bar{\boldsymbol{K}}$ in case of $2-D$ grid with $N = 9,\ldots,400$. The circle–marked plot corresponds to the *Minimal cycles-algorithm*, the crossed-marked one to the *Tree-algorithm*. 65

4.8 Comparison with the Frame Localization algorithm. On left panel, average error on the orientations (modulo 2π) in case of $2-D$ grid with $N = 9,\ldots,400$ and small measurement noise. On the right panel, in the case of $N = 9,\ldots,100$ with large measurement noise. The plot compares the *Minimal cycles-algorithm* (cross–marks), the *Tree-algorithm* (circle-marks), the FL algorithm using $\mathcal{T}$-fundamental cycles (square-marks), and the FL algorithm using minimal cycles (diamond-marks). On the left panel, the results of FL algorithm on minimal cycles are not shown since they overlap those of *Minimal cycles-algorithm*. . . . 67

5.1 Consider a planar grid graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$, with $|\mathcal{V}| = N = 5^2$ and thus $|\mathcal{E}| = 2N - 2\sqrt{N} = 42$. This figure shows the $M - N + 1 = 18$ (they may have multiplicity greater than 1) eigenvalues (blue lines) of the substochastic matrix $\mathbb{E}[U_e^2]$ defined in Eq. 5.13. The eigenvalues are functions of the algorithm stepsize $k \in (0,1)$. From this plot, we can compare the spectral radius ρ of the latter matrix, with the upper bound (red line) shown in Theorem 5.4. 79

5.2 This figure shows a ring graph (left), and a grid graph (left) as an example of a planar graph. Algorithm 3 is analyzed on these different communication networks in Section 5.4.1 and 5.4.2 respectively. 82

5.3 In the left figure we consider a ring graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$ with $N = 20$, the time horizon is $\tau = 300$, and the number of samples is $n_s = 50$. Given Eq. (5.20), the figure shows respectively the initial values $E_1(0)$ (blue dash-star line), $E_2(0)$ (black dash-dot line), $E_3(0)$ (red dash-square line), and asymptotic values $E_1(\infty), E_2(\infty), E_3(\infty)$ (solid lines), while we vary the stepsize $k \in (0,1)$. The projection error E_2 goes to zero for every k, while E_1, E_3 decrease only for k smaller than a certain threshold related to $\lambda_{\max}(C)$ (cfr. Theorem 5.2 for the synchronous algorithm threshold). In the right figure we fix the stepsize $k = 0.3$ and time horizon $\tau = 250$. The figure shows the asymptotic values $E_1(\infty)$ (blue dash-star line), $E_2(\infty)$ (black dash-dot line), $E_3(\infty)$ (red dash-dot line), and $V(\infty)$ (green dashed line) defined in Eq. (5.20), while we vary $N \in [5,100]$ in Algorithm 3. 84

5.4 Given a grid $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ with $N = n^2$ nodes, we fix $n = 5$, the time horizon $\tau = 10^3$ and the number of samples $n_s = 50$. The figure shows the asymptotic values $E_1(\infty)$ (blue dash-star line) and $E_2(\infty)$ (black dash-dot line), defined in Eq. (5.20), while we vary $k \in (0,1)$ in Algorithm 3. If we run the synchronous deterministic algorithm in Piovan et al. [2011a], we obtain the corresponding asymptotic values $E_1^S(\infty)$ (red dash-star line) and $E_2^S(\infty)$ (red dash-dot line). 85

5.5 This figure shows a 2-dimensional Cayley graph (left). Fix $N = 9$ (right) and consider a minimal cycle basis made of the $N - 1 = 8$ cycles of length 4 drawn with counter clockwise orientation, and the two longitudinal cycles $c_9 = \{1,2,13\}$ (blue) and $c_{10} = \{7,8,16\}$ (red), where the numbers denote the edge labels. . . . 86

5.6 Given a Cayley grid $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ with $N = n^2$ nodes, we fix $n = 3$, the time horizon $\tau = 10^5$ and the number of samples $n_s = 10$. The left figure shows the asymptotic values $E_1(\infty)$ (blue dash-star line) and $E_2(\infty)$ (black dash-dot line), defined in Eq. (5.20), while we vary $k \in (0,1)$ in Algorithm 3. The plot shows that if k is greater than a certain threshold, the time required to lower the considered indices becomes greater than 10^5. If we run the synchronous deterministic algorithm in Piovan et al. [2011a], we obtain the corresponding asymptotic values $E_1^S(\infty)$ (red dash-star line) and $E_2^S(\infty)$ (red dash-dot line). The right figure refers to the same graph, with $k = 0.2$, and time horizon $\tau = 10^3$. It shows the evolution in time of $E_1(t)$ (blue line) and $E_2(t)$ (black line), running Algorithm 3, and $E_1^S(t)$ (blue dashed line), $E_2^S(t)$ (black dashed line) running the synchronous deterministic algorithm in Eq. 5.2. 86

5.7 Consider the same setup of Fig. 5.6, with stepsize $k = 0.35$. The left figure shows that the synchronous deterministic algorithm in Eq. 5.2 does not converge (a sufficient condition that guarantees convergence is $k < 2/(1 + \lambda_{max}(C))$, cfr. Piovan et al. [2011a]), while our Algorithm 3 converges and lowers $E_1(0), E_2(0)$. In the right figure we consider a planar grid with $N = n^2 = 9$, and $k = 0.35$. In both cases $\lambda_{max}(C) = 6$, therefore the latter threshold for the deterministic algorithm is $k < 1/8$, whereas it is sufficient to have $k < 1$ for the gossip algorithm (see Theorem 5.3). 87

6.1 This figure shows an environment to be surveilled by a camera network. The environment is represented by a roadmap $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{v_1, \ldots, v_{14}\}$. Edges $\mathcal{E}$ are denoted with solid black lines. Cameras are installed at the locations $\mathcal{V}_c = \{v_1, \ldots, v_7\}$. White rectangles along the edges represent cameras visibility constraints, and the parameters $\boldsymbol{\alpha}$ define a continuous partition of $\mathcal{G}$. Finally, the DF-Trajectory associated with the partition given by $\boldsymbol{\alpha}$ is identified by the closed paths around the cameras. 97

6.2 This figure shows a tree roadmap, and it illustrates the partitioning procedure in the proof of Theorem 6.9. 101

6.3 Optional caption for list of figures 104

6.4 Optional caption for list of figures 106

6.5 This figure shows the convergence of the Synchronous Gradient Partitioning (SGD, blue solid line), the Asymmetric Broadcast Partitioning (AB, black dash-dot line), and the Symmetric Gossip Partitioning algorithms (SG, green dashed line) towards a solution of the continuous min-max partitioning problem. For the simulation we use the configuration in Fig. 6.2, with 111

1 Introduction

The purpose of this Chapter is to give an overview of the dissertation. First, we review the existing literature on cooperative multiagent agent systems, distributed estimation and consensus computation, with a particular focus on network of cameras, patrolling problems and related graph partitioning issues in Section 1.1. Second, we illustrate the organization of next chapters in Section 1.2. We conclude the Chapter in Section 1.3 with a list of publications partly containing the results presented in this thesis.

1.1 Literature synopsis

In this Section we review the existing literature regarding the topics treated in this thesis. This will be instrumental to state more concretely the contributions here presented. In the framework of cooperative multiagent systems, the considered agents may strongly vary, spacing from robots, wireless sensors, computers linked by web connections, to swarms of animals or human beings in the context of opinion dynamics. Nevertheless, all these networked dynamical systems share the same distinctive features, characterizing complex systems.

(i) There is a global goal for the network to be achieved, that could be the estimation of a global quantity of interest in a (wireless) sensor network, or formation control for robotic networks, optimization of a given cost function based on local data, and so on.

(ii) Each agent is able to process only a limited amount of data and computations, due to memory and capability limitations, therefore a centralized algorithm can not be conceived, in order to avoid overload.

(iii) Only local information can be processed, namely only neighboring agents can communicate, for several reasons, including bandwidth limitations, physical constraints, huge number of agents and unreliability of multihops.

(iv) For the previous fundamental features, it is not feasible to provide a central unit, able to collect the necessary data from the whole network to achieve the global goal. In other words, there is no chance to have a single “super-agent”, since the network is often dynamic, changing in time and/or space, and such an iterative centralized decision-making procedure would result too slow and badly performing in terms of noise/error spreading.

Despite the fact that the latter characteristics are limiting constraints on the control of such systems, the increasing interest of the scientific community towards these systems is due to the global behavior that the networks have from simple local laws applying to some nodes, unpredictable when looking at the single agent. Another reason is the real-life application of these systems, since we take part and we are increasingly surrounded by such networks, in several fields, from economics to social networks, from robotics to video surveillance networks. In the last years much effort has been devoted by the scientific community to develop leaderless distributed strategies to solve problems involving interacting agents which need to achieve a common goal. The advantages of these strategies, if compared with the centralized ones, are in terms of robustness of the resulting systems with respect to communication and/or agent failures, and in terms of adaptivity to environmental changes and simplicity in the system tuning. On the other hand, distributed strategies, based on local exchange of information among peer agents, can be more difficult to design and optimize and often exhibit slower convergence to the regime working conditions.

In the remainder of this Section, we first focus on distributed estimation from relative measurements over a network of cooperative multiagent systems in Section 1.1.1. Second, in Section 1.1.2 we focus on cooperative autonomous camera networks with a particular stress on distributed calibration algorithms.Third and finally, we give an overview on surveillance and patrolling problems, and related graph partitioning problems in Section 1.1.3.

1.1.1 Distributed estimation on graphs from relative measurements

Natural candidates for distributed estimation algorithms are iterative consensus algorithms, where each node maintains a local estimate of a parameter of interest, which is updated distributively with a weighted average of the estimates from the local neighbors. Starting from the pioneering work Tsitsiklis [1984], many variations can be found in herein cited literature. In particular further study of this problem has been reported in Moreau [2004], Olfati-Saber et al. [2007], and several applications can be found, among them in the field of management science and statistics DeGroot [1974], in robotics Jadbabaie et al. [2003], in computer science and parallel computing Lynch [1997], Bertsekas and Tsitsiklis [1997], in flocking phenomena Savkin [2004]. The main advantage of consensus algorithms is that they converge exponentially to the centralized solution under very mild communication assumptions, even in the case of a time-varying network topology. Nevertheless, the quantities considered in literature generally live in Euclidean spaces.

In literature, a wide spectrum of works propose distributed estimation algorithms with asynchronous randomized mode of operation, see e.g. Tsitsiklis [1984], Fagnani and Zampieri [2008], and the herein references. It should be pointed out that in many practical applications a node can not simultaneously receive data from two different adjacent nodes (collision can delete messages in wireless environment) and in general it cannot simultaneously transmit to more than one node (for instance for processors nets). Hence, these fundamental limitations must be taken into account even if the networks we consider are quite dense. The use of randomized algorithms turns to be appealing, since they allow to achieve better performance than deterministic ones with comparable complexity. These are some of the reasons why, in Chapter 5, we provide a randomized distributed estimation algorithm. Random linear schemes have been studied for instance in Boyd et al. [2005], Kempe et al. [2003], known as gossip algorithms, in which the evolution matrix of the algorithm changes randomly at every time step. Convergence has to be considered in a probabilistic sense and performance is studied in mean square sense.

In the more general framework of distributed estimation problems on graphs Borkar and Varaiya [1982], Notarstefano and Bullo [2011], Pasqualetti et al. [2010, 2011a], our focus is on the problem of estimating a number of vector valued variables from a number of noisy relative measurements, namely measurements of the difference between certain pairs of these variables. This problem is also known in literature as relative localization, and has attracted much attention in the last decades. The main reason of this increasing interest is due to its multiple applications in the area of sensor networks, among them position localization (Barooah and Hespanha [2005, 2007, 2009]), or time synchronization in which each node has a local clock and the goal is to globally synchronize the whole network clocks or non linear oscillators Kuramoto [1975], R. Karp and Shenker [2003], Sundararaman et al. [2005], Barooah et al. [2006], Giridhar and Kumar [2006], Moreno and Pacheco [2004]. Several localization algorithms have been designed assuming only relative range information L. Doherty and Ghaoui [2001], N. Patwari and Perkins [2003], Moore et al. [2004], and assuming only angle measurements through the use of multiple ultrasound receivers N. B. Priyantha and Teller [2001] or acoustic signals and corresponding time of arrivals Caffery and Stber [1998]. In K. Chintalapudi and Sukhatme [2004] location estimation is performed using both relative distance and angle measurements, which can substantially improve the accuracy of estimates. A certain number of well placed *anchor* nodes that know their position and broadcast that to the network is a usual requirement for many localization schemes.

In the context of position localization in Euclidean spaces, the first papers dealing with noise in the relative measurements, and investigating the effect of network properties on the estimation error applied the Cramer-Rao lower bound with Gaussian assumptions on noise Niculescu and Nath [2004], N. Patwari and Perkins [2003], or provided numerical studies focusing on node density and network size in A. Savvides [2005].

To our knowledge, the first and complete analytical results appeared in Barooah and Hespanha [2005, 2007, 2009], in which they provide an unbiased estimator with minimum variance, that can be computed in a decentralized fashion. They investigated the question of how the estimation error of the optimal estimate scales with distance, and how it is affected by the network topology. In Chapter 3, our concern is to investigate how the error on a fixed edge is affected by the error corresponding to edges “far”in terms of length on minimum path along the graph. We focus on a particular and exemplary class of graphs, the so-called Cayley graphs, crf. Section 2.1.2, which can be embedded in a proper d-dimensional Euclidean space. Cayley graphs show interesting geometric properties and easy spectral characterization of the fundamental matrices describing such graphs.

1.1.2 Distributed calibration

In the core of this dissertation (Chapter 4, 5, and 6) we focus on camera networks. Networks of this kind are a widely used architecture, and could be used in a variety of applications, such as 3-D reconstruction of large environments, tracking of mobile targets, video surveillance and motion capture Aghajan and Cavallaro [2009].

In a network of cameras one of the most crucial problems is calibration. For each camera this consists in understanding what is its position and orientation with respect to a global and common reference coordinate system.

The calibration problem over Euclidean spaces has recently been studied by Barooah and Hespanha in a great detail (see Barooah and Hespanha [2005], Barooah et al. [2006], Barooah and Hespanha [2007]), as already explained in Section 1.1.1. By means of well–known methods in computer vision (see Ma et al. [2003]), we can tackle the calibration issue using as input data the relative positions and orientations of pairs of cameras whose sensing regions overlap. The latter can be formulated as a minimization problem on the manifold $SE(3)$, since each camera pose is mathematically formalized as the couple (R,T), where $R \in SO(3)$, $T \in \mathbb{R}^3$ with respect to a fixed external reference frame. In particular, one of the most interesting subproblems of the latter is the calibration of the orientations, which can be seen as an optimization problem in the manifold $SO(3)$, Tron and Vidal [2009b], Tron et al. [2011], Scardovi et al. [2007], Sarlette and Sepulchre [2009b]. In particular, this optimization problem consists in minimizing a piece-wise convex cost function defined on $SO(3)$, characterized by several local minima. A straightforward consequence is that the standard gradient descent procedure usually proposed in literature must be initialized correctly in order to avoid trajectories of the estimate that get stuck in some local minima that are not global minima. A distributed method for camera localization and calibration based on Belief Propagation (as opposed to consensus) is proposed in Devarajan and Radke [2007]. However, no conditions for a consistent solution are imposed and the non-Euclidean structure of the rotations is not rigorously exploited.

In Piovan et al. [2011b] the problem of calibration on $SO(2)$ is considered when measurements of relative orientations are affected by additive noise. the main idea is to exploit the cycles in the graph, since the input noisy relative orientations my sum to zero on closed paths, up to integer multiples of 2π. Chosen a particular family of cycles in the graph, they propose a novel iterative algorithm solving a non convex constrained minimization problem. Such procedure is cycle-distributed, and converges exponentially to the manifold of relative orientations with null cycle errors.

Finally, in Singer [2011] an estimation algorithm is proposed for a measurement model in which some measures are ideal, while others are completely random. The estimate of the orientations is obtained via the computation of the eigenvector associated with the largest eigenvalue of a suitable Hermitian matrix.

1.1.3 Patrolling problem and graph partitions

Remote surveillance of human activities for civil and military applications is receiving considerable attention from the research community. Many tasks requiring repetitive execution can be achieved by cooperative multirobot and multicamera networks, including the monitoring of oil spills Clark and Fierro [2007], detection of fires Kingston et al. [2008], patrolling of environments Susca et al. [2008]. Public areas such as banks, art galleries, private houses, prisons, department stores, and parking lots, are now equipped with camera networks to detect important activities, O'Rourke [1987], Susca et al. [2008]. From a technological perspective, one of the main challenges consists of developing efficient algorithms for the cameras to autonomously and distributively complete tracking, surveillance, and recognition tasks.

In Chapter 6, we focus on the patrolling problem, considering one dimensional environments embedded in the Euclidean space, and we tackle the issue of performing the surveillance task by means of a fixed camera network. In mobile robotics, the patrolling problem consists of scheduling the motion of a team of autonomous agents in order to detect intruders or important events, e.g., see Alberton et al. [2012], Baseggio et al. [2010], Machado et al. [2003], Pasqualetti et al. [2011b]. Nevertheless, when dealing with camera networks, there are several differences, first our video devices are placed at fixed locations in the environment and therefore there are additional conceivable physical constraints, in general not considered in mobile robot networks. Our approach is also related to graph-clearing and graph-search problems, see e.g. A. Kehagias and Singh [2009], Kolling and Carpin [2010], Parsons [1978], in which agents are usually mobile robots, and therefore these results do not apply to the setup we considered.

In the context of perimeter patrolling, there are several papers providing distributed strategies for line topologies Baseggio et al. [2010], Carli et al. [2011], Spindler et al. [2012], Pasqualetti et al. [2013]. These works tightly resembles our setup, since they provide optimal cameras trajectories that minimize the detection time for *smart* intruders, both static and dynamic with a complete knowledge of the cameras trajectories. These optimal strategies are designed by means of graph partitions, where the goal is to minimize the maximum load of all the cameras, in terms of size of their patrolled region. In Chapter 6, our concern is to generalize such approach to more general topologies.

Graph partitioning problems are classically combinatorial, see e.g. Andreev and Racke [2006], Arkin et al. [2006], Even et al. [2004], G. Even and Schieber [1997] and the references therein, where the goal is to partition the nodes set in a suitable way, depending on the size of the partition. Our model has a novel formulation, namely we define the continuous partition of a graph, meaning that, once each edge is parametrized by a continuous variable, the aim is to provide a cut on each edge, in order to mathematically split the environment among the cameras.

1.2 Contributions of this thesis

The main contributions of each chapter are as follows.

Chapter 2. In Chapter 2 we introduce the notations, and mathematical tools that will be intensively used in the remainder of the thesis. We start recalling some standard results in graph theory, with an emphasis on algebraic and topological characterization of the cycle structure of a network in Section 2.1.1, and the Abelian Cayley graph class in Section 2.1.2. Afterwards, we report some linear algebra basics in Section 2.2. An overview of Markov Chains is provided in Section 2.3, focusing on their analogy with electrical resistive networks in Section 2.3.2. Finally, Section 2.4 reports the main properties of the DFT (Discrete Fourier Transform) used in Chapter 3.

Chapter 3. In this Chapter, we consider relative localization in sensor networks and its intrinsic performance limitations, in terms of the mean error on each component of the optimal estimator of the position vector. The relative error is computed as a function of the eigenvalues of the network: using this formula and focusing on an exemplary class of networks (the Abelian Cayley networks), we study the role of the network topology and the dimension of the networks in the error characterization. If we assume to have sequences of graphs of increasing size (in terms of number of nodes), we investigate the asymptotic behavior of the relative error and how ”far“(in terms of path along the graph) relative errors due to noise in measurements affects the local relative error. The analysis, with a complementary numerical study, is provided for different dimensions of the Cayley grid.

Chapter 4. In this Chapter we face the problem of calibrating in a distributed way a network of autonomous cameras, namely the problem of estimating a common orientation reference frame. We assume each camera obtains noisy measurements of its relative orientation with respect to some other cameras. The set of measurements can be described by a graph having the cameras as nodes and having an edge between two communicating cameras. We propose a two-step estimation algorithm based on a choice of a basis for the set of graph cycles. First we compute a set of integer numbers, which provides a first rough estimate of the orientations. Second, this information is used to cast the estimation problem in a suitable quadratic minimization problem. We propose two different implementations of this main idea, corresponding to two different basis of cycles. Analytical performances are described and compared in terms of the worst–case scenario. The final part of this Chapter is devoted to a numerical study that deepen the comparison between the two versions of our algorithm, and a third algorithm proposed in the literature for solving the same problem.

Chapter 5. This chapter focuses on the same problem of Chapter 4, with the difference that the camera network is based on a different communication model. We design a distributed algorithm for the cameras to autonomously calibrate adopting an asynchronous gossip-like communication protocol. The idea is to exploit the cycles in the graph, along which all relative measurements must sum to zero, in order to reduce the noise. Probabilistic convergence and numerical experiments are provided. For general planar graph topologies the algorithm is proved to converge to the set of angles with zero cycle error, almost surely and in the mean square sense. Numerical experiments investigate the performance of the proposed randomized algorithm on non-planar graphs.

Chapter 6. In this final part, we deal with a calibrated camera network that tackles the issue of patrolling an environment modeled by a graph embedded in the Euclidean plane. The goal is to design surveillance trajectories for the camera network to detect intruders in the environment, through a continuous graph partitioning problem. The proposed partitioning algorithms are decentralized, based on different communication protocols both deterministic and probabilistic, and they rely on convex optimization tools. Analytical and numerical results are provided.

Chapter 7. In this Chapter we draw the conclusions of this work, and we describe a wide spectrum of possible research directions for all the treated topics.

1.3 Publications

The results of the present work are based on the following papers.

Journal Papers

- D. Borra, Fabio Pasqualetti, F. Bullo, *Continuous graph partitioning for camera network surveillance*, Automatica, Note: submitted, July 2012.
- D. Borra, R. Carli, F. Fagnani, E. Lovisari, S. Zampieri, *Autonomous calibration algorithms for planar networks of cameras*, Automatica, Note: submitted, July 2012.

Conference Papers

- D. Borra, F. Fagnani, *Asynchronous distributed calibration of camera networks.* In ECC'13, Zurich, Switzerland, July 2013. Note: Accepted.
- D. Borra, Fabio Pasqualetti, F. Bullo, *Continuous graph partitioning for camera network surveillance*, IFAC NecSys 2012, Santa Barbara, September 2012.
- D. Borra, R. Carli, F. Fagnani, E. Lovisari, S. Zampieri, *Autonomous calibration algorithms for networks of cameras*, ACC 2012, Montréal, June 2012.

The research pursuit developed during my PhD course includes also opinion dynamics issues. We studied a hybrid bounded confidence model for opinion formation in a large group of agents exposed to the persuasive action of a small number of strong opinion leaders. In this thesis the focus is on localization and detection problems dealing with camera networks, thus the following work has not been included in order to maintain a consistent dissertation avoiding off-topic digressions. The contents and results achieved can be found in the following articles.

Journal Papers

- D. Borra, T. Lorenzi, *A hybrid model for opinion formation*, Z. angew. Math. Phys., DOI: 10.1007/s00033-012-0259-z, 2012.
- D. Borra, T. Lorenzi, *Asymptotic analysis of continuous opinion dynamics models under bounded confidence*, Commun. Pure Appl. Anal., 12, 1487-1499, 2013.
- D. Borra, F. Fagnani, S. Fosson, *Analysis of a Krause-based one-dimensional model in the presence of stubborn agents*, Note: in preparation.

All publications are available online at: http://calvino.polito.it/~borra/.

2 Mathematical tools

Before presenting the core of the work, this Chapter defines recurrent notation and collects some specific tools that are important ingredients in the sequel. This includes, first, a brief review of graph theory in Section 2.1, in which a deep characterization of the cycle structure of a graph is provided (Section 2.1.1), and the definition and properties of Cayley graphs are recalled (Section 2.1.2). Second, linear algebra tools and definitions are presented in Section 2.2. Third, in Section 2.3 an overview on Markov Chains is reported, with a particular focus on the electrical analogy in Section 2.3.2. Forth and finally, Section 2.4 contains a brief overview of the Discrete Fourier Transform, that will be extensively used in Chapter 3.

2.1 Graph theory preliminaries

In this Section we recall some known facts from algebraic graph theory which will be instrumental in the development of this work. Some basic definitions are here recalled. For further details, see e.g. Diestel [2005], Gutin [2003].

An undirected graph is a couple $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, N\}$ is the set of nodes, and $\mathcal{E}$ is a subset of unordered pairs of elements of $\mathcal{V}$ called edges[1]. We let $M := |\mathcal{E}|$. An *orientation* on $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a pair of maps $s : \mathcal{E} \to \mathcal{V}$ and $t : \mathcal{E} \to \mathcal{V}$ such that $e = \{s(e), t(e)\}$ for every $e \in \mathcal{E}$. According to this definition, $s(e)$ and $t(e)$ are called the source and terminal node of the edge e, respectively. Assume from now on that we have fixed an orientation (s, t) on $\mathcal{G}$. The *incidence matrix* $B \in \{\pm 1, 0\}^{\mathcal{E} \times \mathcal{V}}$ of $\mathcal{G}$ is defined by putting $B_{e,s(e)} = 1$, $B_{e,t(e)} = -1$, and $B_{e,v} = 0$

[1] More precisely an edge is a subset of $\mathcal{V}$ with two elements.

if $v \neq s(e), t(e)$.

A *weighted Laplacian* of $\mathcal{G}$ is given by $L = B^T \mathcal{C} B$ where $\mathcal{C} \in \mathbb{R}_+^{M \times M}$ is a diagonal matrix whose (e, e) entry can be seen as the weight of edge e. We denote by Z the *Green matrix* of L, which is the unique matrix such that $Z\mathbb{1} = 0$ and $ZL = I - \frac{1}{N}\mathbb{1}\mathbb{1}^T$. This definition can be generalized to non symmetric Laplacians, but we will restrict to this case for simplicity. A graph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ is a subgraph of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ if $\mathcal{V}' \subseteq \mathcal{V}$ and $\mathcal{E}' \subseteq \mathcal{E}$. A connected acyclic graph is called *tree.* A *spanning tree* of a graph is a subgraph with N vertices and $N-1$ edges that form a tree. To conclude, if $\boldsymbol{v} \in \mathbb{R}^N$ is a vector, then $|\boldsymbol{v}| \in \mathbb{R}^N$ is a vector whose i-th component is $|v_i|$. The symbol $|\boldsymbol{v}| \leq \boldsymbol{p}$, where both $\boldsymbol{v}$ and $\boldsymbol{p}$ are vectors of the same dimension n, means $|v_i| \leq p_1$ for all $i = 1, \dots, n$.

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a spanning tree $\mathcal{T} = (\mathcal{V}, \mathcal{E}_{\mathcal{T}})$ of $\mathcal{G}$ is a connected subgraph of $\mathcal{G}$ which is a tree. Notice that $|\mathcal{E}_{\mathcal{T}}| = N - 1$.

A path h of length n is an ordered sequence of nodes $h = (v_1, v_2, \cdots, v_{n+1})$ such that $\{v_i, v_{i+1}\} \in \mathcal{E}$ for all $i = 1, \dots, n$. A path $h = (v_1, v_2, \cdots, v_{n+1})$ is said to be *closed* if $v_1 = v_{n+1}$. A closed path $h = (v_1, v_2, \cdots, v_n, v_1)$ is said to be a *cycle* if $n \geq 3$ and $v_i \neq v_j$ for all $i, j \in \{1, \dots, n\}$ with $i \neq j$. The support of a path is given by the set of its edges, namely, if $h = (v_1, v_2, \cdots, v_{n+1})$, then $\text{supp}(h) := \{e \in \mathcal{E} \,|\, e = \{v_i, v_{i+1}\}, \exists i = 1, \dots, n\}$.

A cycle is closed path, and let us denote the set of cycles of $\mathcal{G}$ as $\mathcal{H}$. A graph $\mathcal{G}$ is said to be *planar* if its nodes and edges can be embedded in the Euclidean plane without intersection between edges (see Diestel [2005]). Given a planar graph $\mathcal{G}$, its *dual graph* $\tilde{G} = (\mathcal{H}, \tilde{\mathcal{E}})$ is defined by putting $\{h, h'\} \in \tilde{\mathcal{E}}$ if and only if h, and h' are adjacent cycles.

In the following, we first focus on the cycle structure of a graph in 2.1.1, that will be extensively used in Chapter 4 and 5. Second, we recall the definition and main properties of Cayley graphs in 2.1.2, studied in Chapter 3.

2.1.1 Cycle structure of a graph

Consider now $\mathbb{Z}^{\mathcal{E}}$, the $\mathbb{Z}$-module of $\mathbb{Z}$-valued row vectors whose components are labeled by $\mathcal{E}$. Given $\boldsymbol{r} \in \mathbb{Z}^{\mathcal{E}}$, we define its support as

$$\text{supp}(\boldsymbol{r}) := \{e \in \mathcal{E} \,|\, \boldsymbol{r}(e) \neq 0\}$$

We now associate with every path h, an element $\boldsymbol{r}_h \in \mathbb{Z}^{\mathscr{E}}$ as follows. First, if $h = (v_1, v_2)$, we put

$$\boldsymbol{r}_h(e) = \begin{cases} B_{ev_1}, & \text{if } e = \{v_1, v_2\} \\ 0, & \text{otherwise.} \end{cases}$$

Then, for a generic path $h = (v_1, v_2, \cdots, v_{n+1})$ we define $\boldsymbol{r}_h(e)$ as $\boldsymbol{r}_h(e) = \sum_{i=1}^{n} \boldsymbol{r}_{(v_i, v_{i+1})}(e)$. In particular, for paths h with non-repeating edges, $\boldsymbol{r}_h$ is built by assigning $\boldsymbol{r}_h(e) = 1$ if the edge e appears in h and it is crossed with the same orientation of $\mathscr{G}$, assigning $\boldsymbol{r}_h(e) = -1$ if the edge e appears in h and it is crossed with the reverse orientation of $\mathscr{G}$ and finally assigning $\boldsymbol{r}_h(e) = 0$ if the edge e does not appear in h.

We give some algebraic properties of the row vectors $\boldsymbol{r}_h$ associated with the paths h. We start by observing that,

$$(\boldsymbol{r}_{(v_1,v_2)} A)_v = \sum_{e \in \mathscr{E}} \boldsymbol{r}_{(v_1,v_2)}(e) A_{ev} = B_{\{v_1,v_2\},v} B_{\{v_1,v_2\},v} = \begin{cases} 1, & \text{if } v = v_1, \\ -1, & \text{if } v = v_2, \\ 0, & \text{if } v \neq v_1, v_2. \end{cases}$$

In other words, $\boldsymbol{r}_{(v_1,v_2)} B = \mathbb{1}_{v_1} - \mathbb{1}_{v_2}$, where the symbol $\mathbb{1}_v$ means the column vector in $\mathbb{R}^{\mathscr{V}}$ with the entry of position v equal to 1 and all the other entries equal to 0. Observe moreover that, if $h = (v_1, v_2, \cdots, v_n, v_{n+1})$, then

$$\boldsymbol{r}_h B = \sum_{i=1}^{n} \boldsymbol{r}_{(v_i, v_{i+1})} B = \sum_{i=1}^{n} \mathbb{1}_{v_i} - \mathbb{1}_{v_{i+1}} = \mathbb{1}_{v_1} - \mathbb{1}_{v_{n+1}}.$$

The latter equality proves that $\boldsymbol{r}_h B = 0$, if h is a closed path.

Denote now by Γ the $\mathbb{Z}$-submodule of $\mathbb{Z}^{\mathscr{E}}$ generated by all the vectors $\boldsymbol{r}_h$ as h varies in the set of closed paths. It holds true that Γ has dimension equal to $M - N + 1$, i.e. there exist $\boldsymbol{r}_{h_1}, \ldots, r_{h_{M-N+1}}$ forming a $\mathbb{Z}$-basis of Γ. This fact is well known in the slight different context where no orientation is considered and where vector spaces are used Diestel [2005]. The following proposition clarifies the situation in our setting.

Proposition 2.1. ***(Cycle space characterization)*** *We have that*

(i) $\Gamma = \{\boldsymbol{r} \in \mathbb{Z}^{\mathscr{E}} \mid \boldsymbol{r} B = 0\}$*;*

(ii) Γ *has dimension equal to* $M - N + 1$.

Proof. (a) The fact that $\Gamma \subseteq \{\boldsymbol{r} \in \mathbb{Z}^{\mathscr{E}} \,|\, \boldsymbol{r}B = 0\}$ follows from the previous arguments.

(b) Fix a spanning tree $\mathscr{T} = (\mathscr{V}, \mathscr{E}_{\mathscr{T}})$ of $\mathscr{G}$ and let $e_1, \dots, e_{M-N+1}$ be the edges in $\mathscr{E} \setminus \mathscr{E}_{\mathscr{T}}$. Select the family cycles $h_1, \dots, h_{M-N+1}$ by taking h_i to be the only cycle in $\mathscr{G}$ with edges in $\mathscr{E}_{\mathscr{T}} \cup \{e_i\}$ and such that $\boldsymbol{r}_{h_i}(e_i) = 1$. We now prove that the $\boldsymbol{r}_{h_i}$'s generate $\{\boldsymbol{r} \in \mathbb{Z}^{\mathscr{E}} \,|\, \boldsymbol{r}A = 0\}$. Let $\boldsymbol{r} \in \mathbb{Z}^{\mathscr{E}}$ such that $\boldsymbol{r}B = 0$ and let $\tilde{\boldsymbol{r}} := \boldsymbol{r} - \sum_{i=1}^{M-N+1} \boldsymbol{r}(e_i)\boldsymbol{r}_{h_i}$. Notice that $\tilde{\boldsymbol{r}}B = 0$ and that $\operatorname{supp}(\tilde{\boldsymbol{r}}) \subseteq \mathscr{E}_{\mathscr{T}}$. We show now that these facts imply $\tilde{\boldsymbol{r}} = 0$. Indeed, if $\operatorname{supp}(\tilde{\boldsymbol{r}}) \subseteq \mathscr{E}_{\mathscr{T}}$ and $\operatorname{supp}(\tilde{\boldsymbol{r}}) \neq \emptyset$, then $\operatorname{supp}(\tilde{\boldsymbol{r}})$ would include at least one leaf, namely a node v^* such that there exists only one edge $e^* \in \operatorname{supp}(\tilde{\boldsymbol{r}})$ containing v^*. In this case

$$0 = (\tilde{\boldsymbol{r}}B)_{v^*} = \sum_e \tilde{\boldsymbol{r}}(e)B_{ev^*} = \boldsymbol{r}(e^*)B_{e^*v^*},$$

which yields $\tilde{\boldsymbol{r}}(e^*) = 0$, a contradiction.

(c) It remains to prove the $\mathbb{Z}$-independence of the row vectors $\boldsymbol{r}_{h_1}, \dots \boldsymbol{r}_{h_{M_N+1}}$. Assume that $\alpha_i \in \mathbb{Z}$ are such that $\sum_i \alpha_i \boldsymbol{r}_{h_i} = 0$. Then, for any $\ell = 1, \dots, M - N + 1$, we have that

$$0 = \sum_i \alpha_i \boldsymbol{r}_{h_i}(e_\ell) = \alpha_\ell \boldsymbol{r}_{h_\ell}(e_\ell) = \alpha_\ell,$$

which shows that $\alpha_i = 0$.

□

Proposition 2.2. ***(Connection between the cycle matrix and the incidence matrix)*** *Let $\boldsymbol{r}_{h_1}, \dots \boldsymbol{r}_{h_{M-N+1}}$ be a $\mathbb{Z}$-basis of Γ. Defined moreover $R \in \mathbb{Z}^{(\mathscr{E} \setminus \mathscr{E}_{\mathscr{T}}) \times \mathscr{E}}$ to be the matrix having $\boldsymbol{r}_{h_1}, \dots, r_{h_{M-N+1}}$ as rows. Then*

(i) $\ker R = \operatorname{Im} B$, *where* $\ker R := \{\boldsymbol{K} \in \mathbb{Z}^{\mathscr{E}} | R\boldsymbol{K} = 0\}$ *and* $\operatorname{Im} A := \{\boldsymbol{K} \in \mathbb{Z}^{\mathscr{E}} | \boldsymbol{K} = B\boldsymbol{h}, \ \exists \boldsymbol{h} \in \mathbb{Z}^{\mathscr{V}}\}$.

(ii) there exists $X \in \mathbb{Z}^{\mathscr{E} \times (\mathscr{E} \setminus \mathscr{E}_{\mathscr{T}})}$ such that $RX = I$ where I is the identity matrix.

Proof. 1) First observe that, if we have two matrices $R, R' \in \mathbb{Z}^{(\mathscr{E} \setminus \mathscr{E}_{\mathscr{T}}) \times \mathscr{E}}$ formed by two different $\mathbb{Z}$-bases of Γ, then $\ker R = \ker R'$ easily follows from the algebra of matrices over $\mathbb{Z}$ (see Newman [1972]). It remains to prove the assertion for a particular choice of the $\mathbb{Z}$-basis of Γ. Consider the $\mathbb{Z}$-basis $\boldsymbol{r}_{h_1}, \dots \boldsymbol{r}_{h_{M-N+1}}$ of Γ, built from a spanning tree $\mathscr{T}$ of $\mathscr{G}$ as described in point b) of the proof of Proposition 2.1. The fact that $\operatorname{Im} B \subseteq \ker R$ follows from Proposition 2.1. At this point, the only thing to be shown is that $\ker R \subseteq \operatorname{Im} B$. Let $\boldsymbol{K} \in \ker R$. Since rows in R form a basis of Γ, we have that $\boldsymbol{r}_h \boldsymbol{K} = 0$ for every closed path h. Let us fix a node $v_0 \in \mathcal{V}$. Then for each node $v \in \mathcal{V}$, let γ_v be the path in the tree $\mathcal{T}$ connecting v_0 to v. Define now the column vector $\boldsymbol{h} \in \mathbb{Z}^{\mathcal{V}}$ with components

$$h_v := \boldsymbol{r}_{\gamma_v} \boldsymbol{K} = \sum_{e \in \mathcal{E}} \boldsymbol{r}_{\gamma_v}(e) \boldsymbol{K}_e \,,$$

where $\boldsymbol{r}_{\gamma_v}$ is the row vector in $\mathbb{Z}^{\mathcal{E}}$ associated with the path γ_v. We now show that $\boldsymbol{K}_e = h_{t(e)} - h_{s(e)}$, for each $e \in \mathcal{E}$. A straightforward consequence is that $\boldsymbol{K} = B\boldsymbol{h}$, i.e. $\boldsymbol{K} \in \mathrm{Im}B$. Consider the closed path h defined merging the paths $\gamma_{s(e)}$, $(s(e), t(e))$ and $-\gamma_{t(e)}$, where we recall that $-\gamma_{t(e)}$ denotes the path obtained by reversing $\gamma_{t(e)}$. Observe that $\boldsymbol{r}_h = \boldsymbol{r}_{\gamma_{s(e)}} + \boldsymbol{r}_{(s(e),t(e))} - \boldsymbol{r}_{\gamma_{t(e)}}$. It follows that

$$0 = \boldsymbol{r}_h \boldsymbol{K} = \boldsymbol{r}_{\gamma_{s(e)}} \boldsymbol{K} + \boldsymbol{r}_{(s(e),t(e))} \boldsymbol{K} - \boldsymbol{r}_{\gamma_{t(e)}} \boldsymbol{K} = h_{s(e)} + K_e - h_{t(e)}$$

whence the thesis holds.

b) This follows from the fact that $\mathbb{Z}^{\mathcal{E}}/\Gamma$ is a torsion free module, and consequently it is free, being $\mathbb{Z}$ a principal ideal domain (see Hungeford [1980]). □

Define the *essential cycle matrix* as the square $(M - N + 1)$-dimensional matrix $C := RR^T$.

Given a planar graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, its *dual graph* $\tilde{\mathcal{G}} = (\mathcal{F}, \tilde{\mathcal{E}})$ is defined by putting $\{c, c'\} \in \tilde{\mathcal{E}}$ if and only if c and c' are adjacent minimal cycles.

Let $\mathcal{V} = \{1, \dots N\}$. Given a symmetric matrix $P \in \mathbb{R}^{N \times N}$ we define the *graph associated with P* as $\mathcal{G}_P = (\mathcal{V}, \mathcal{E}_P)$ where $\mathcal{E}_P := \{\{i, j\} \mid P_{ij} = P_{ji} \neq 0\}$.

The complementary part of the graph in the plane is a union of disconnected bounded domains (called faces) and one unbounded one. In this case, a basis of Γ can be easily obtained considering all the vectors $\boldsymbol{r}_h$ as h varies in the set $\mathcal{F}$ of closed paths which are the boundaries of the (bounded) faces all run in a clockwise fashion. Notice that if c and c' are two such closed paths sharing edge e, then $r_c(e) r_{c'}(e) = -1$. We will refer to the closed paths in $\mathcal{F}$ as to the *minimal cycles* of $\mathcal{G}$.

Given a cycle $c \in \mathcal{F}$, let $|c|$ denote its length, and d_c the number of *adjacent* minimal cycles, namely the elements in $\mathcal{F}$ that share with c one edge. Clearly $d_c \leq |c|$. We define a *border* *cycle*, any element in $\mathscr{F}$ such that $d_c < |c|$. A *border edge* will consequently be any edge in the support of only one minimal cycle in $\mathscr{F}$.

2.1.2 Cayley graphs

We introduce the concept of Cayley graph defined on Abelian groups, see Alon and Roichman [1994], Babai [1979] for further details.

Let G be any finite Abelian group, denoted by $(G,+)$ of order $|G| = N$, and let S be a subset of G containing zero. The *Cayley graph* $\mathscr{G}(G,S)$ is the directed graph with vertex set $\mathscr{V} = G$ and arc set $\mathscr{E} = \{(g,h) \mid g-h \in S\}$. A Cayley graph is always in-regular and out-regular with degree $d = |S|$, and it is strongly connected if and only if $< S >= G$, namely the set S generates the group G. The graph is undirected if S is such that $-S = S$, and we say that S is inverse-closed. A matrix $p \in \mathbb{R}^{G\times G}$ is said to be a *Cayley matrix over the group* G if

$$P_{i,j} = P_{i+h,j+h}, \forall i,j,h \in G$$

and clearly it exists a function $\pi : G \to \mathbb{R}$, called *generator of the Cayley matrix*, such that $P_{i,j} = \pi(i-j)$. Notice that if π_1 and π_2 are generators of the Cayley matrices P_1 and P_2 respectively, then $\pi_1 + \pi_2$ is the generator of $P_1 + P_2$ and $\pi_1 * \pi_2$ is the generator of $P_1 \cdot P_2$, where

$$(\pi_1 * \pi_2)(i) := \sum_{j\in G} \pi_1(j)\pi_2(i-j), \forall i \in G.$$

It implies that P_1 and P_2 commute.
Denote with $\mathscr{G}_P$ the Cayley graph associated to P, whom generator is π, then $S = \{h \in G : \pi(h) \neq 0\}$. Moreover a Cayley stochastic matrix is automatically doubly stochastic, in fact $P\mathbb{1} = \mathbb{1}$ if and only if $\mathbb{1}^T P = \mathbb{1}^T$, and in this case the function π is a probability distribution on the group G. The simpler distribution is $\pi(i) = \frac{1}{|S|}, \forall i \in S$.

2.2 Linear algebra tools

Well known basic concepts from spectral theory are here recalled, see e.g. Hungeford [1980], Gantmacher [1990]. Given a matrix $A \in \mathbb{R}^{N\times N}$, we denote by $\sigma(A) = \{\lambda_1,\ldots,\lambda_N\}$ the set of all the eigenvalues of A, called *spectrum* of A. Set $\lambda_{max} = \max_{i=1,\ldots,N} |\lambda_i|$, then let us define the *spectral radius* as

$$\rho(A) := \sup\{|\lambda| : \lambda \in \sigma(A), \lambda \neq \lambda_{max}\}.$$

Given a matrix $A \in \mathbb{R}^{M\times N}$, we define the *Moore-Penrose pseudo-inverse* of A denoted by $A^\dagger$ as a general way to find the solution of the linear equations

$$A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{b} \in \mathbb{R}^m, \boldsymbol{x} \in \mathbb{R}^N.$$

The Moore-Penrose solution is of the form $\boldsymbol{x} = A^\dagger \boldsymbol{b}$, where $A^\dagger$ satisfies the following

$$\begin{aligned} AA^\dagger A &= A, \\ A^\dagger AA^\dagger &= A^\dagger, \\ (AA^\dagger)^T &= AA^\dagger, \\ (A^\dagger A)^T &= A^\dagger A. \end{aligned}$$

The following properties hold.

(i) If $M = N$, and if A is full rank, then $A^\dagger = A^{-1}$.

(ii) If $M > N$, the solution is a minimizer of

$$\|A\boldsymbol{x} - \boldsymbol{b}\|^2.$$

Hence, in general it is not possible to find a solution to these equations, and $\boldsymbol{x}$ is such that $A^\dagger \boldsymbol{x}$ is the closest to $\boldsymbol{b}$.

(iii) $M < N$, then the Moore-Penrose solution minimizes the 2-norm $\|\boldsymbol{x}\|$. In other words, there are more constraining equations than unknowns in $\boldsymbol{x}$, thus some free variables remain.

When A is full rank, the Moore-Penrose pseudo-inverse can be directly computed as

$$\begin{aligned} M < N &: \ A^\dagger = A^T(AA^T)^{-1}, \\ M > N &: \ A^\dagger = (A^T A)^{-1} A^T. \end{aligned}$$

In general, $A^\dagger$ can be calculated using the SVD (Singular Value Decomposition).

In order to estimate the spectrum of a matrix, let us recall the following well-known result, cfr. Hungeford [1980].

Theorem 2.3. ***(Gershgorin Circle Theorem)*** *Given a matrix $A = (a_{ij}) \in \mathbb{R}^{N\times N}$, define*

$$\begin{aligned} r_i &= \sum_{j=1, j\neq i}^{N} a_{ij}, \quad R_i = \textstyle\sum_{j\neq i} |a_{ij}|, i = 1, \dots, N \\ c_j &= \sum_{i=1, i\neq j}^{N} a_{ji}, \quad C_j = \textstyle\sum_{i\neq j} |a_{ij}|, j = 1, \dots, N \end{aligned}$$

then the eigenvalues of A lie within $\bigcup_{i=1}^{N} D(a_{ii}, R_i)$, where $D(a_{ii}, R_i)$ is the so-called Gershgorin disc*, defined as $D(a_{ii}, R_i) = \{x \in \mathscr{C} \mid |x - a_{ii}| \leq R_i\}$.*

In the remainder of this Section we focus on a particular class of matrices, called circulant, for which the eigenvalues can be explicitly computed in a straightforward way. We define $C \in \mathscr{M}(n, \mathbb{R})$ as a *circulant matrix*, if it has the following form

$$C = \begin{pmatrix} c_0 & c_{n-1} & \dots & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & & c_2 \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ c_{n-2} & & \ddots & \ddots & c_{n-1} \\ c_{n-1} & c_{n-2} & \dots & c_1 & c_0 \end{pmatrix}$$

hence it is identified by the vector $(c_0\ c_1\ \dots\ c_{n-1})$ of length n, which is the first column. The other columns are cyclic permutations of the first one, with offset equal to the column index. These matrices have several properties.

(i) The determinant of a circulant matrix C can be computed as

$$\det(C) = \prod_{j=0}^{n-1} (c_0 + c_1 w_j + c_2 w_j^2 + \dots + c_{n-1} w_j^{n-1}) = \prod_{j=0}^{n-1} \left(\sum_{i=0}^{n-1} c_i w_j^i \right)$$

where w_j are the n-th roots of unity

$$w_j = e^{\frac{2\pi\imath j}{n}}, j = 0, \dots, n-1.$$

(ii) The eigenvalues of a circulant matrix can be calculated explicitly. The eigenvectors are the following

$$\boldsymbol{v}_j = (1, \omega_j, \dots, \omega_j^{n-1})^T, j = 0, \dots, n-1$$

and the corresponding eigenvalues are

$$\lambda_j = c_0 + c_{n-1}\omega_j + c_{n-2}\omega_j^2 + \dots + c_1\omega_j^{n-1}, j = 0, \dots, n-1.$$

2.3 Overview on Markov Chains

The basics on this topic can be found in Aldous and Fill, Norris [1998], Grimmett and Stirzaker [2001]. First, we recall some basic definitions and properties in Section 2.3.1. Second, Section 2.3.2 presents the analogy between electrical networks and Markov Chains, and the well-known notion of effective resistance.

2.3.1 Basic definitions

Let I be a countable, set called *state space*, and each $i \in I$ is called *state*. We say that $\lambda = \{\lambda_i : i \in I\}$ is a *measure* of I, if $0 \leq \lambda_i < \infty$ for all $i \in I$. If the *total mass* $\sum_{i\in I} \lambda_i = 1$, we call λ *distribution*. Consider a probability space $(\Omega, \mathscr{F}, \mathbb{P})$, where Ω is the sample space, $\mathscr{F}$ is the set of events, and $\mathbb{P}$ is a probability measure function (see Shiryaev [1989], Grinstead and Snell [1997] for further details). Recall that a *random variable* X is a function $X : \Omega \to I$, and suppose λ defines a distribution of X, that is $\lambda_i = \mathbb{P}(X = i) = \mathbb{P}(\{\omega \in \Omega : X(\omega) = i\})$ for each $i \in I$. Therefore X models a random state that takes value i with probability λ_i.

A matrix $P = (p_{ij})_{i,j\in I}$ is called *stochastic* if $p_{ij} \geq 0$ for all $i, j \in I$, and $\sum_j p_{ij} = 1$ for all i. In other words, every row $(p_{ij} : j \in I)$ is a distribution. If we denote by $\mathbb{1}$ the vector with all components equal to 1, the last row condition on P can simply be rephrased as $P\mathbb{1} = \mathbb{1}$. The matrix P is instead called *sub-stochastic* if $(P\mathbb{1})_i \leq 1$ for all i, and there exists at least one index i_0 for which the inequality is strict. We can associated the matrix P with a directed graph $\mathscr{G}_P$ with $|I|$ nodes, and weighted edges, namely edge (i, j) has weight p_{ij}. Whenever the weight is null, there is no edge. If P is symmetric, the associated graph is undirected. We shall now formalize the properties for a Markov Chain by means of the corresponding matrix P.

We define $(X_n)_{n\geq 0}$ as a *Markov Chain* (MC for short) with initial distribution λ and *transition* *matrix P*, if

(i) X_0 has distribution λ;

(ii) for any $i_0,\dots,i_{n+1} \in I$, the so-called *Markovian property* holds

$$\mathbb{P}(X_{n+1} = i_{n+1} \mid X_0 = i_0,\dots,X_n = i_n) = p_{i_n,i_{n+1}}.$$

Intuitively, condition (ii) states that, given the present state, the future and past states are independent. We now address the problem of finding what is the probability that after n steps, our MC is in a given state. It reduces to calculate the entries of the n-th power of the transition matrix $P^n = (p_{ij}^{(n)})_{i,j\in I}$, in particular $p_{ij}^{(n)}$ is the n-step transition probability from state i to state j. We can describe the MC, identifying the communicating classes of the chain, namely the subset of states “connected”by positive transition probabilities. A class $C \subseteq I$ is said to be *closed* if, given $i \in C, p_{ij} > 0$, then $j \in C$. The MC associated with P is said to be *irreducible* if I is a single class. In other words, the associated graph $\mathcal{G}_P$ is strongly connected, namely, for all $i, j \in I$, there exists a path in $\mathcal{G}_P$ connecting i to j. P is said to be *aperiodic* if the greatest common divisor of the lengths of all cycles in $\mathcal{G}_P$ is one. Note that if there exists at least one loop, i.e. $p_{ii} > 0$ for some $i \in I$, is a sufficient condition to have aperiodicity. Irreducibility and aperiodicity are two independent properties.

Note that one of the standard examples of MC is the simple random walk on countable sets. Given a weighted graph $\mathcal{G} = (\mathcal{V},\mathcal{E},W)$ with weight matrix W, we can define a discrete-time *random walk* on $\mathcal{G}$ to be the MC with transition matrix $p_{ij} = w_{ij}/w_i$, for all $j \neq i$, where $w_i = \sum_j w_{ij}$. If W is the identity matrix, then $P = AD^{-1}$, where A is the adjacency matrix and D the degree matrix, see Section 2.1. Let $(X_n)_{n\geq 0}$ be a MC with transition matrix P. The *hitting time* of a subset A of I is the random variable $H^A : \Omega \to \mathbb{N} \cup \{\infty\}$ given by

$$H^A(\omega) = \inf\{n \geq 0 : X_n(\omega) \in A\}$$

where $\inf\emptyset = \infty$ by convention. Furthermore $h_i^A = \mathbb{P}_i(H^A < \infty)$ is the probability that, starting from i, the MC ever hits A. If A is a closed class, h_i^A is called *absorption probability*. The mean time taken for $(X_n)_{n\geq 0}$ to reach A, starting from $i \in I$, is given by $E_i^A := \mathbb{E}_i(H^A)$. Many of the long-time properties of Markov Chains are connected with the notion of an invariant distribution or measure. We say that the measure $\lambda = (\lambda_i : i \in I)$ is *invariant* (or stationary) if

$\lambda P = \lambda$. A discrete-time MC $(X_n)_{n\geq 0}$, with transition matrix P, is *reversible* if

$$\pi_i p_{ij} = \pi_j p_{ji}, \text{for all } i, j \in I,$$

where $\boldsymbol{\pi}$ is the stationary distribution. For this object a cyclic tour property holds, as enlighten in the proof of Theorem 3.2. The theory of continuous-time MC closely parallels that of the discrete-time chains, see [Norris, 1998, Chapter 2, 3]. There can be defined also continuous-state MC, but this topic goes beyond the purpose of this work. A continuous-time chain is specified by non-negative *transition rates* $(q_{ij})_{i\neq j\in I}$, without constraints on the sum. Let us define $q_i = \sum_{j\neq i} q_{ij}$and extend (q_{ij}) to a matrix Q by putting $q_{ii} = -q_i$. The continuous-time MC $(X_t : t \geq 0)$ can be defined by the infinitesimal description: given $X_t = i$, the chance that $X_{t+dt} = j$ is $q_{ij}dt$ for each $j \neq i$. A chain is said to be *ergodic* if it is aperiodic and positive recurrent, i.e. if the mean recurrence time is finite. We define the *commuting time* between two states $i, j \in I$ of an ergodic Markov chain as the expected time, starting from i, to go to j and then back to i. Recall this result, that will be used in Chapter 3, Theorem 3.2.

Theorem 2.4. ***([Aldous and Fill, Corollary 10, Chapter 2])*** *Denote with $P_k(T_i < T_j)$ the probability to hit node i before node j, starting from k.*

$$P_k(T_i < T_j) = \frac{E_{ki} + E_{ji} - E_{ki}}{E_{ji} + E_{ij}}$$

where E_{ij} denotes the average time to reach state j starting from i.

A classical result is the following. If P is a sub-stochastic matrix with $\mathcal{G}_P$ connected, then $P^t \to 0$ for $t \to +\infty$. This fact as a deeper meaning if the considered matrix is *primitive*, that is irreducible and aperiodic.

In case the matrix is primitive, the left invariant measure π has a straightforward interpretation as the asymptotic probability of the states in the chain. In other terms, starting from any state and waiting for a sufficiently large time slot, the value π_i gives the probability that the chain is in the state i. More formally, from the well known Frobenius-Perron theory D.A. Levin and Wilmer, if P is primitive, it has eigenvalue 1 with algebraic multiplicity 1, and all the other eigenvalues have modulus smaller than 1, thus

$$P^n \overset{n\to\infty}{\longrightarrow} \mathbb{1}\pi^T \tag{2.1}$$

where π is the invariant measure of P, namely the left eigenvector of P associated with 1, and normalized such that $\sum_i \pi_i = 1$. Moreover, for each $i \in I$, $\pi_i > 0$.
If P is a stochastic and irreducible matrix, we call it *consensus matrix*. Given a consensus matrix P, with invariant measure $\boldsymbol{\pi}$, the *Green matrix* Z of P is defined as

$$Z := \sum_{t \geq 0} (P^t - \mathbb{1}\boldsymbol{\pi}^T). \tag{2.2}$$

From direct computations, the Green matrix G can be considered as a pseudoinverse of the Laplacian $L := I - P$ of the graph associated to the MC defined by P. For the sake of completeness, let us recall the following Lemma.

Lemma 2.5. ***Green Matrix properties*** *Given a consensus matrix P with invariant measure π, and Z be the associated Green matrix, then Z is the unique solution of*

$$\begin{cases} ZL = I - \mathbb{1}\boldsymbol{\pi}^T, \\ Z\mathbb{1} = \mathbf{0}. \end{cases}$$

2.3.2 The electrical analogy

In this Section we present how the well-known notion of effective resistance may be exploited to give bounds on errors in estimating quantities in a decentralized network. The analogy between Markov Chains, and resistive electrical networks first appeared in the work Doyle and Snell. This is a powerful tool to lead our intuitions on the behavior of the chain on the basis of the physics underlying the corresponding electrical network. Furthermore, it allows to give simple and linear proofs to many results. Out interest is to focus on the average effective resistance, as already studied in Barooah and Hespanha [2005], Barooah et al. [2006], Barooah and Hespanha [2007], from which we took many results, stated in Chapter 3 (Section 3.2) without proof. Effective resistance is also a performance metrics as studied in Aldous and Fill, Lovisari et al. [2012], Lovisari [2012], Ghosh et al. [2008], used e.g. for clock synchronization algorithms, see Giridhar [2006], Giridhar and Kumar [2006]. In the remainder of this Section we recall some basic definitions and known results, that are instrumental for Chapter 3, and Chapter 4 (especially Section 4.4.3).

Let us define a *resistive electrical network* as a weighted graph $\mathcal{G}_C = (\mathcal{V}, \mathcal{E}, C)$, where $C \in \mathbb{R}_{\geq 0}^{\mathcal{V} \times \mathcal{V}}$ is a symmetric matrix defining the conductance of the resistor connecting any two nodes. For each edge, the pair of nodes are connected by resistors, thus $\{i, j\} \in \mathscr{E}$ if and only if $C_{ij} \neq 0$, for some $i, j \in \mathscr{V}$. In other words, each wire connecting i to j has conductance C_{ij}, i.e. resistance $1/C_{ij}$. The easiest design is to set $C_{ij} = 1$ whenever there is a link between two nodes. Mathematically speaking, an electrical resistive network is defined by a weighted graph $\mathscr{G}_C$, together with a function on nodes called *voltage*, and a flow called *current*, satisfying some physical constraints (see Eq. (2.3)). A *flow* $\boldsymbol{f} \in \mathbb{R}^{\mathscr{E}}$ on a graph is required to satisfy $f_{ij} = -f_{ji}$ if (i, j) is an edge, and $f_{ij} = 0$ otherwise. A conventional arbitrary orientation of edges is chosen, in order to describe the current flow through the network, by means of Kirchhoff and Ohm Laws. By abuse of notation, let us denote by $\mathscr{E}$ the oriented edges, given the initial undirected edges and the two arbitrary functions $s, t : \mathscr{E} \to \mathscr{V}$ (starting and terminating node). Coherently with Section 2.1, let B denote the incidence matrix respectively. We now define the diagonal conductance matrix $\bar{C} \in \mathbb{R}^{\mathscr{E}\times\mathscr{E}}$, as follows

$$B^T \bar{C} B = \mathrm{diag}(C\mathbb{1}) - C,$$

that is $[B^T C B]_{ij} = C_i := \sum_{j\in\mathscr{V}} C_{ij}$, if $i = j$, $[B^T C B]_{ij} = -C_{ij}$ if $(i, j) \in \mathscr{E}$, and it is null otherwise. First, let us formalize the current flow, defining $\boldsymbol{i} \in \mathbb{R}^{\mathscr{V}}$, satisfying $\boldsymbol{i}^T\mathbb{1} = 0$, where the k-th component of $\boldsymbol{i}$ is the current injected (extracted) if the sign is positive (negative) in node k. Second, we define $\boldsymbol{j} \in \mathbb{R}^{\mathscr{E}}$ and $v \in \mathbb{R}^{\mathscr{V}}$ to be the current flows on edges and potentials at the nodes. Therefore, we can write Kirchhoff and Ohm Laws in matrix form, as follows

$$\begin{cases} B^T \boldsymbol{j} = \boldsymbol{i}, \\ \bar{C} B \boldsymbol{v} = \boldsymbol{j}. \end{cases} \tag{2.3}$$

If we consider the latter, with $\boldsymbol{j}, \boldsymbol{v}$ as unknowns, we can find $\boldsymbol{v}$ by solving

$$B^T \bar{C} B \boldsymbol{v} = \boldsymbol{i}. \tag{2.4}$$

For connected networks, a solution can be found, from well-known works Doyle and Snell, and it is unique up to additive constants. Given a connected electrical network $\mathscr{G}_C$, let us define the *effective resistance* between nodes i and j as

$$\mathscr{R}_{ij}(C) := v_i - v_j,$$

where we set $\boldsymbol{i} = e_i - e_j$,[2] and $\boldsymbol{v}$ is the solution of Eq. (2.4). Furthermore, let us define the *average effective resistance* as

$$\mathscr{R}_{\text{ave}}(C) = \frac{1}{2N^2} \sum_{i,j \in \mathcal{V}} \mathscr{R}_{ij}(C),$$

where N is the number of nodes. Whenever $\mathscr{G}$ is an undirected graph, we fix $C = A$ as the adjacency matrix. Given a reversible MC with transition matrix P and invariant measure π, we can define the associated electrical resistive network, setting the conductance matrix as

$$C := N\text{diag}(\pi)P,$$

where $\text{diag}(\pi) \in \mathbb{R}^{\mathcal{V} \times \mathcal{V}}$ denotes the diagonal matrix with π on the main diagonal. As it is described in Section 3.2, the effective resistance is tightly related to the Linear Quadratic cost (LQ for short), an index to measure the performance of a consensus algorithm, widely used in the control community. Consider a reversible MC with transition matrix P converging to $\mathbb{1}\boldsymbol{\pi}^T$, as stated in Eq. (2.1), then we define the LQ-cost J, by means of the Frobenius norm[3], as follows

$$J(P) := \frac{1}{N} \sum_{t \geq 0} \|P^t - \mathbb{1}\boldsymbol{\pi}^T\|_F^2.$$

In Barooah and Hespanha [2008, 2009], Lovisari [2012], they prove how this index is related to the effective resistance of a suitable electrical network. It depends on the topology of $\mathscr{G}_P$ only, and not on the values of P. We deepen these aspects in Section 3.2.

2.4 Discrete Fourier transform over finite Abelian groups

We briefly recall the basic concepts related to the theory of the Discrete Fourier transform (DFT for short) over finite Abelian groups, see Terras [1999] for further explanations. Let χ be any *character* of G, that is, a map $\chi : G \to \mathscr{C}^*$ such that $\chi(g+h) = \chi(g)\chi(h)$, for any $g, h \in G$. It follows that $\chi(g)^N = \chi(Ng) = \chi(0) = 1, \forall g \in G$, which means that χ takes values on the N-th roots of unity. The set of all characters of the group G forms an Abelian group with respect to the point wise multiplication. It is called the *character group*, denoted by $\hat{G}$. Since G is

[2] In our convention, $e_i \in \mathbb{R}^N$ denotes the canonical basis vector, with $(e_i)_j = 1$ if $i = j$, and zero otherwise.
[3] The Frobenius norm is defined as $\|A\|_F^2 := \text{tr}(A^T A)$.

isomorphic to its associated group of characters $\hat{G}$, i.e. $G \cong \hat{G}$ if G a locally compact abelian group, there is a bijective correspondence $\phi : G \to \hat{G}$ such that $\phi(h) = \chi_h : G \to \mathscr{C}^*$. Notice that $\chi_0(g) = 1, \forall g \in G$ is the trivial character, and the zero of $\hat{G}$. Now consider the vector space $\mathscr{C}^G$ with the canonical Hermitian form defined as

$$< f_1, f_2 >:= \sum_{g \in G} f_1(g) f_2^*(g), g \in G.$$

It can be proven that $\mathscr{B} = \left\{N^{-1/2}\chi \mid \chi \in \hat{G}\right\} = \left\{N^{-1/2}\chi_h \mid h \in G\right\}$ is an orthonormal basis of $\mathscr{C}^G$. The *Fourier transform* of a function $f : G \to \mathscr{C}^*$ is defined as $\hat{f} : \hat{G} \to \mathscr{C}^*$ such that

$$\hat{f}(\chi) = \sum_{g \in G} \chi(-g) f(g), \forall \chi \in \hat{G}. \tag{2.5}$$

The cyclic case is instrumental to study characters for any finite Abelian group, which is isomorphic to a finite direct sum of cyclic groups. Assume that $G = Z_{N_1} \oplus \ldots \oplus \mathbb{Z}_{N_r}$, the characters of G are precisely the maps $\chi : G \to \mathscr{C}$ such that $\chi((g_1, \ldots, g_r)) = \chi^{(1)}(g_1) \ldots \chi^{(r)}(g_r)$, where $\chi^{(i)} \in \hat{\mathbb{Z}}_{N_i}, i = 1, \ldots, r$. A Cayley matrix P can be seen as an endomorphism of $\mathscr{C}^G$, namely $P : \mathscr{C}^G \to \mathscr{C}^G$, such that

$$(Pf)(g) = \sum_{h \in G} P_{hg} f(g), g \in G.$$

It can be shown that each character χ is an eigenfunction of P, more precisely

$$P\chi_h = \hat{\pi}(\chi_h)\chi_h, h \in G$$

where $\hat{\pi}(\chi_h)$ is the corresponding eigenvalue. Explicitly

$$\hat{\pi}(\chi_h) = \sum_{g \in G} \pi(-g)\chi_h(g), \forall h \in G.$$

Since $\mathscr{B}$ is a basis, P is diagonalizable and its spectrum is

$$\sigma(P) = \left\{\hat{\pi}(\chi) : \chi \in \hat{G}\right\}.$$

A character can be interpreted as a linear function $\chi : \mathscr{C} \to \mathscr{C}^G$, such that $\chi(z) = z\chi$, and its *adjoint linear functional* is $\chi^* : \mathscr{C}^G \to \mathscr{C}$, such that

$$\chi^*(f) =< f, \chi > .$$

Thus $N^{-1}\chi\chi^* : \mathscr{C}^G \to \mathscr{C}^G$ is a linear function that projects $\mathscr{C}^G$ onto the eigenspace associated with χ. Hence P can be rewritten as

$$P = N^{-1} \sum_{\chi \in \hat{G}} \hat{\pi}(\chi)\chi\chi^* .$$

If P represents the transition matrix of a simple random walk on an undirected graph, the spectrum of the associated laplacian L is

$$\sigma(L) = \left\{1 - \hat{\pi}(\chi) : \chi \in \hat{G}\right\}$$

and the corresponding spectrum of Z, Green matrix of L will be

$$\sigma(Z) = \left\{\frac{1}{1 - \hat{\pi}(\chi)} : \chi \in \hat{G}\right\}.$$

Notice that if $Z^{\mathscr{G}}$ denotes the Green matrix of the graph, and Z the Green matrix of the Markov Chain associated to the simple random walk, it holds that $Z^{\mathscr{G}}_{vw} = Z_{vw}/d_w$.

3 Error propagation for relative localization over geometric networks

In this Chapter we present a general framework in which we aim to estimate the error of relative localization algorithms in sensor networks, focusing on how it propagates through the network. We consider the intrinsic performance limitations, in terms of the mean error on each component of the optimal estimator of the position vector. The relative error is computed as a function of the eigenvalues of the network: using this formula and focusing on an exemplary class of networks (the Abelian Cayley networks, defined in Section 2.1.2), we study the role of the network topology and the dimension of the networks in the error characterization. If we assume to have sequences of graphs of increasing size (in terms of number of nodes), we investigate the asymptotic behavior of the relative error and how ”far“(in terms of minimal paths along the graph) relative errors due to noise in measurements affect the local relative error. The analysis is provided for different dimensions of the Cayley grid.

3.1 Introduction

In the more general framework of distributed estimation problems on graphs Borkar and Varaiya [1982], Notarstefano and Bullo [2011], Pasqualetti et al. [2010, 2011a], we want to estimate vector valued variables from a certain number of noisy relative measurements, namely measurements of the difference between certain pairs of these variables. This issue is called in literature *relative localization*, and has attracted increasing interest in the last years. The main causes of this attention are the multiple applications in various areas, including sensor networks for position localization (Barooah and Hespanha [2005, 2007, 2009]), time synchronization in network clocks or non linear oscillators Kuramoto [1975], R. Karp and

Shenker [2003], Sundararaman et al. [2005], Barooah et al. [2006], Giridhar and Kumar [2006], Moreno and Pacheco [2004]. Several localization algorithms have been designed assuming only relative range information L. Doherty and Ghaoui [2001], N. Patwari and Perkins [2003], Moore et al. [2004], and assuming only angle measurements through the use of multiple ultrasound receivers N. B. Priyantha and Teller [2001] or acoustic signals and corresponding time of arrivals Caffery and Stber [1998]. In K. Chintalapudi and Sukhatme [2004] location estimation using both relative distance and angle measurements can substantially improve the accuracy of estimates. A certain number of well placed *anchor* nodes that know their position and broadcast this information to the network is a usual requirement for many localization schemes.

The first papers investigating the effect of network properties on the estimation error applied the Cramer-Rao lower bound with Gaussian assumptions on noise Niculescu and Nath [2004], N. Patwari and Perkins [2003], or provided numerical studies focusing on node density and network size in A. Savvides [2005].

To our knowledge, the most complete analytical results appeared in Barooah and Hespanha [2005, 2007, 2009], in which they provide an unbiased estimator with minimum variance, that can be computed in a distributed way.

In a relative localization problem, we have a group of sensors, and each of them is endowed with a real value state, that we refer to as position, but the value may represent different quantities, such as temperature, humidity, time, velocity, percentage of specific substances in air, water or soil and so on. The measurement model can be expressed in terms of a graph. Every node is allowed to take noisy measurements of the difference between its own state and its neighbors states. The goal of the algorithm is to reconstruct the absolute position of each sensor, with respect to a global reference frame, exploiting the noisy data, up to additive constants. Inspired by Barooah and Hespanha [2005], we consider the optimal linear estimate for the differences between the remaining variables and the reference, that is one particular variable called *anchor*. This is called the least squares optimal estimator, and it is unbiased with minimum variance among all possible estimators. In deriving these results, the authors establish and exploit an analogy between the minimum variance optimal estimator of the network and the effective resistance in the associated electrical network. We focus on an exemplary class of graphs the Abelian Cayley graphs: this assumption turns out to be very suitable to obtain sharp results on the role of the network topology (and specifically of the graph dimension) in determining how the relative error on a fixed edge is affected by the noises along edges that ”far“in the graph, in terms of shortest path.

Our contribution is organized as follows. In Section 3.2, we present our model and we recall the main achievements already known (Barooah and Hespanha [2005, 2007]) about the optimal estimator of the global positions of the agents, which is unbiased, and it has minimum variance, that shows asymptotic properties depending on the size of the geometric grid, as the number of nodes asymptotically increases. Second, in Section 3.3 we provide analytical and numerical results on the single component of the relative error in the case of Abelian Cayley networks, according to different dimensions of the considered grid. Finally, in Section 3.4, we draw the conclusion of this Chapter, summarizing the results.

3.2 Relative vector localization

In this Section, we formalize the problem of *relative position localization* in large networks, recalling the main facts known for the Euclidean space case, cfr. Barooah and Hespanha [2005, 2007]. Let the sensor network be modeled by a graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, with N nodes and M edges, and fix an arbitrary orientation, defined by the functions $s, t : \mathscr{E} \to \mathscr{V}$. For each edge $e \in \mathscr{E}$, $s(e), t(e)$ are called starting and terminating node respectively (see Section 2.1). Suppose to have relative noisy measurements on each edge, stacked in a vector $\boldsymbol{\eta} \in \mathbb{R}^M$, defined as follows

$$\eta_e := \bar{\theta}_{s(e)} - \bar{\theta}_{t(e)} - \epsilon_e, \text{ for each } e \in \mathscr{E}$$

where the vector $\boldsymbol{\epsilon} \in \mathbb{R}^M$ casts the noise that affects each relative measurement and $\bar{\boldsymbol{\theta}} \in \mathbb{R}^N$ are the true measurements at each node. In our assumptions the noise is bounded by a fixed constant $\bar{\epsilon} > 0$ for each realization. The latter definition can be written in matrix form as

$$\boldsymbol{\eta} = B\bar{\boldsymbol{\theta}} - \boldsymbol{\epsilon},$$

with B incidence matrix of the graph.

Suppose to implement the least squares algorithm that gives as output the optimal estimate $\hat{\boldsymbol{\theta}} \in \mathbb{R}^N$, that minimizes the cost function

$$V(\boldsymbol{\theta}) = \left\| B\boldsymbol{\theta} - \boldsymbol{\eta} \right\|^2,$$

namely $\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} V(\boldsymbol{\theta})$. The global minimum $\boldsymbol{\theta}^* \in \mathbb{R}^N$ can be computed explicitly as

$$\boldsymbol{\theta}^* = (B^T B)^\dagger (B^T \boldsymbol{\eta}) = ZB^T \boldsymbol{\eta},$$

where $B^T B = L$ is the Laplacian of the graph, and its pseudoinverse is the so-called Green matrix Z, see Section 2.1. The distributed algorithm proposed in Barooah and Hespanha [2005], provides the optimal estimate, hence $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}^*$. Moreover such estimator is unbiased, i.e. $\mathbb{E}[\hat{\boldsymbol{\theta}}] = \bar{\boldsymbol{\theta}}$. We now recall some basic results from Barooah and Hespanha [2005, 2007], in which they establish asymptotic upper and lower bounds on the estimation error variance of a node's variable as a function of the Euclidean distance in a drawing of the graph between the node and the anchor. These bounds result in a classification of graphs (civilized and dense), based on how the variance grows asymptotically with distance (linearly, logarithmically, or bounded in grids of dimension $1, 2$ or $d \geq 3$ respectively). These bounds, being true for the optimal estimate, serve as a benchmark against which algorithms devised for specific applications can be evaluated for large networks. In deriving these asymptotic behavior, the analogy between the minimum variance and the effective resistance in the associated electrical network is exploited, see Section 2.3.2. We briefly recall formally the latter in the following Theorem. Consider an electrical network where on each edge there is a resistor of 1 Ohm, and denote by $\mathscr{R}_{v1}$ the effective resistance between node v and the anchor node 1.

Theorem 3.1. *[Barooah and Hespanha, 2005, Th. 10, Th. 11]*
(Asymptotic characterization of the optimal estimate variance) *If $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ is d-dimensional lattice embedded in $\mathbb{R}^d$ with N nodes and M edges, and if the noises ϵ_e along each edge are i.i.d. with mean value zero, and variance σ^2, then*

$$\frac{1}{N}\mathrm{Var}(\hat{\boldsymbol{\theta}}) := \frac{1}{N}\mathbb{E}[\|\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}\|^2] = \frac{1}{N}\mathbb{E}\|ZB^T\boldsymbol{\epsilon}\|^2 = \frac{\sigma^2}{N}\sum_{v\in\mathscr{V}} \mathscr{R}_{v1}.$$

Furthermore,

$$\frac{1}{N}\mathrm{Var}(\hat{\boldsymbol{\theta}}) \sim \begin{cases} N, \text{ if } d = 1, \\ \log(N), \text{ if } d = 2, \\ \text{const, if } d \geq 3. \end{cases}$$

3.3 Propagation of errors

The formula in Theorem 3.1 does not clarify at which extent cooperation is useful in this context, and, more generally, how the various measurements of the network affect each other. A basic question is to understand the effect of data fusion on a single edge measurement. An interesting issue is to understand at which extent a single very bad measurement on an edge, will affect the rest of the network. The idea underlying this study, is that local effect should take place and the influence of other measurements should thus decay with distance in the graph. The purpose of this Chapter is to investigate these issues. Our goal is to estimate the *relative error* on each edge (misplacement of the estimate towards the correct relative difference, on component e), in terms of

$$m_e := (B\hat{\boldsymbol{\theta}} - \boldsymbol{\eta})_e = \hat{\theta}_{s(e)} - \hat{\theta}_{t(e)} - \eta_e = x_{s(e)} - x_{t(e)} + \epsilon_e \tag{3.1}$$

where $x_i := \hat{\theta}_i - \bar{\theta}_i$denotes the error on the state of each node $v \in \mathcal{V}$.

In particular, we are interested in investigating how m_e depends on the measurements attained at edges which are “far”in the graph.

First, we consider a simple random walk over the graph $\mathcal{G}$, defined by the transition matrix $P = D^{-1}A$, where D is the degree matrix and A the adjacency matrix (cfr. Section 2.3 and Aldous and Fill for further details). In this framework, the relative error can be interpreted componentwise in terms of transition probabilities, and in particular can be formulated as a function of the effective resistance of the corresponding electrical network, as stated in Theorem 3.2.

Second, in the case of a ring graph, we can explicitly compute the upper bound on the relative error in the least squares optimal estimator (in Corollary 3.3), in terms of the elements of the Green matrix G, using the tool of the Discrete Fourier Transform, recalled in Section 2.4.

From now on, we consider an undirected graph $\mathcal{G}$ with a fixed arbitrary orientation $s, t : \mathcal{E} \to \mathcal{V}$, as described in Section 2.1. Let us denote by $P_k(T_i < T_j)$ the probability to hit node i before node j, starting from k. Given the edges $(i, j), e \in \mathcal{E}$, we now define the so-called *amplifying factor* $\Gamma_{(i,j),e}$ as

$$\Gamma_{(i,j),e} := 2\mathcal{R}_{ij}(P_{s(e)}(T_i < T_j) - P_{t(e)}(T_i < T_j)), \tag{3.2}$$

where $\mathscr{R}_{ij}$ denotes the effective resistance from node i to node j. Such quantity arises from computations in Theorem 3.2, and it describes how measurement on edge (i,j) is affected by the measurement attained at e. It is straightforward to notice that the amplifying factor is bounded. Our purpose is to investigate how $\Gamma_{(i,j),e}$ decays as the graphical distance between the edges (i,j) and e increases. From the following result, we can recover an explicit expression for $m_{ij} = x_i - x_j + \epsilon_{ij}$, identifying edge e with (i,j).

Theorem 3.2. ***(Propagation of errors in general topologies)*** *Consider the graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, for any edge $(i,j) \in \mathscr{E}$*

$$x_i - x_j = \sum_{e \in \mathscr{E}} \Gamma_{(i,j),e} \varepsilon_e, \tag{3.3}$$

where $\Gamma_{(i,j),e}$ is the amplifying factor defined in Eq. (3.2).

Proof. Consider the simple random walk over the graph $\mathscr{G}$ with transition probabilities $D^{-1}A$ and invariant measure $\boldsymbol{\pi}$, see Section 2.3. The associated Laplacian matrix is $L_M = I - D^{-1}A$ such that $L_{\mathscr{G}} = D - B = DL_M$, where $D = (d_j)_{j \in \mathscr{V}}$ is the degree matrix. The related Green matrix is Z_M, with $Z_{\mathscr{G}} = Z_M D^{-1}$, thus it holds that $(Z_{\mathscr{G}})_{ij} = (Z_M)_{ij} \frac{1}{d_j}$.

We know that $\boldsymbol{x}$ is such that $B^T B \boldsymbol{x} = -B^T \boldsymbol{\varepsilon}$, or equivalently $(I - \mathbf{1}\boldsymbol{\pi}^T)\boldsymbol{x} = -Z_{\mathscr{G}} B^T \boldsymbol{\varepsilon}$, by the definition of the Green matrix. Since $B^T B = L_{\mathscr{G}}$ is the Laplacian of the graph, let $Z_{\mathscr{G}}$ be the Green matrix of $L_{\mathscr{G}}$. Denote with $(Z_{\mathscr{G}})_i$ and $(Z_{\mathscr{G}})_j$ the i-th row and the j-th row of $Z_{\mathscr{G}}$,

$$\begin{aligned} x_i - x_j &= ((Z_{\mathscr{G}})_j - (Z_{\mathscr{G}})_i) B^T \boldsymbol{\varepsilon} = \sum_k ((Z_{\mathscr{G}})_{jk} - (Z_{\mathscr{G}})_{ik}) \left(\sum_e B^T_{ke} \varepsilon_e \right) \\ &= \sum_e ((Z_{\mathscr{G}})_{j,s(e)} - (Z_{\mathscr{G}})_{j,t(e)} - ((Z_{\mathscr{G}})_{i,s(e)} - (Z_{\mathscr{G}})_{i,t(e)})) \varepsilon_e. \end{aligned} \tag{3.4}$$

Let us denote with $E_{jk} = \mathbb{E}_j(\tau_k)$ the expected hitting time τ_k needed to reach k starting from j, see Section 2.3 for further details.

Remind that $(Z_M)_{j,s(e)} - (Z_M)_{i,s(e)} = (E_{j,s(e)} - E_{i,s(e)})\pi_{s(e)}$, where $\pi_{s(e)} = \frac{d_{s(e)}}{|\mathscr{E}|}$ is the stationary distribution for any undirected graph. Then $(Z_{\mathscr{G}})_{j,s(e)} - (Z_{\mathscr{G}})_{i,s(e)} = (E_{j,s(e)} - E_{i,s(e)}) \frac{1}{|\mathscr{E}|}$ and consequently

$$x_i - x_j = \frac{1}{|\mathscr{E}|} \sum_e ((E_{j,s(e)} - E_{j,t(e)}) - E_{i,s(e)} + E_{i,t(e)}) \varepsilon_e. \tag{3.5}$$

Recall that $P_k(T_i < T_j)$ denotes the probability to hit node i before node j, starting from k, and recall Theorem 2.4 from Aldous and Fill. Moreover, from the cyclic tour property of reversible Markov Chains we have that

$$E_{j,s(e)} - E_{j,t(e)} - E_{i,s(e)} + E_{i,t(e)} = E_{s(e),j} - E_{t(e),j} - E_{s(e),i} + E_{t(e),i}$$

we can derive

$$P_{s(e)}(T_i < T_j) - P_{t(e)}(T_i < T_j) = \frac{E_{s(e),j} - E_{s(e),i} - E_{t(e),i} + E_{t(e),j}}{E_{ji} + E_{ij}}.$$

From the latter and (3.5), we obtain

$$x_i - x_j = \frac{1}{|\mathscr{E}|}\sum_e (P_{s(e)}(T_i < T_j) - P_{t(e)}(T_i < T_j))(E_{ji} + E_{ij})\epsilon_e$$

where $E_{ji} + E_{ij} = 2|\mathscr{E}|\mathscr{R}_{ij}$ is the commute time and $\mathscr{R}_{ij}$ is the effective resistance from node i to node j.

As a result, Eq. (3.3) follows immediately. Moreover $\mathscr{R}_{ij}$ is bounded, precisely $\mathscr{R}_{ij} \leq 1$, consequently $x_i - x_j \leq 2\sum_e (P_{s(e)}(T_i < T_j) - P_{t(e)}(T_i < T_j))\epsilon_e$. Observe that $P_{s(e)}(T_i < T_j) - P_{t(e)}(T_i < T_j)$ represents a potential difference, exploiting the analogy between Markov Chains and electric resistive networks, cfr. Section 2.3. □

Corollary 3.3. ***(Explicit relative error for ring graphs)*** *If $\mathscr{G}$ is a ring graph, it holds the following inequality for the misplacement of the estimate towards the correct relative position of node i and node j, where C is a constant ($C = 4$),*

$$\left|m_{ij}\right| := \left|x_i - x_j + \varepsilon_{ij}\right| \leq \bar{\varepsilon}C, \text{ for each } e = (i,j) \in \mathscr{E}. \tag{3.6}$$

Proof. Considering a ring graph, it holds

$$P_k(T_i < T_j) = \frac{k-j}{i-j},$$

hence the (3.6) follows easily observing that

$$\begin{aligned} x_i - x_j &= 2\mathscr{R}_{ij}\varepsilon_{ij} + 2\mathscr{R}_{ij}\sum_{e\neq(i,j)}\left(\frac{s(e)-j}{i-j} - \frac{s(e)+1-j}{i-j}\right)\varepsilon_e \\ &= 2\mathscr{R}_{ij}\varepsilon_{ij} + 2\mathscr{R}_{ij}\sum_{e\neq(i,j)}\left(\frac{1}{j-i}\right)\epsilon_e \leq 2\bar{\epsilon} + 2\bar{\epsilon}\sum_{e\neq(i,j)}\frac{1}{N-1} = 4\bar{\epsilon} \end{aligned}$$

since $P_{s(e)}(T_i < T_j) = 1$ if $e = (i,j)$. □

From the expression derived in Eq. (3.4), it easily follows:

Corollary 3.4. ***(Explicit relative error for Δ-regular graphs)*** *Let $\mathcal{G}$ be a Δ-regular graph, then the amplifying factor defined in Eq.* (3.2) *is*

$$\Gamma_{(i,j),e} = \frac{1}{\Delta}(Z_{j,s(e)} - Z_{j,t(e)} - (Z_{i,s(e)} - Z_{i,t(e)})).$$

Let us take $G = \mathbb{Z}_N^d$ and let $S = \{\pm\boldsymbol{e}_1, \dots, \pm\boldsymbol{e}_d\}$ be the subset of G generating the d-thorus graph, cfr. Section 2.1.2 for the detailed definition of an Abelian Cayley graph. In this case $\Delta = 2d$, and define $S^+ = \{\boldsymbol{e}_1, \dots, \boldsymbol{e}_d\}$, namely the set of all possible directions of connection in the graph, thus $\left|S^+\right| = |S|/2$. Due to the theory on DFT recalled in Section 2.4, we can explicitly write the elements of Z in terms of the eigenvalues and the associated characters in the case of the d-dimensional thorus.

Lemma 3.5. ***(Explicit Green matrix for Cayley graphs)*** *If $\mathcal{G}$ is a d-dimensional Cayley graph, given the simple random walk associated with $\mathcal{G}$, then the elements of the Green matrix are*

$$Z_{\boldsymbol{vw}} = \frac{1}{N^d} \sum_{\boldsymbol{l}\in\mathbb{Z}_N^d - \{\boldsymbol{0}\}} \frac{1}{1 - \frac{1}{|S^+|}\sum_{\boldsymbol{k}\in S^+} cos\left(\frac{2\pi}{N}\boldsymbol{l}\cdot\boldsymbol{k}\right)} e^{\frac{2\pi\imath}{N}\boldsymbol{l}\cdot(\boldsymbol{v}-\boldsymbol{w})}, \boldsymbol{v}, \boldsymbol{w} \in \mathbb{Z}_N^d$$

where $\lambda_{\boldsymbol{l}}$ is the eigenvalue of the character $\chi_{\boldsymbol{l}}$, for any $\boldsymbol{l} \in \mathbb{Z}_N^d$, and precisely (see Section 2.4*)*

$$\lambda_{\boldsymbol{l}} = \frac{1}{1 - \frac{1}{|S^+|}\sum_{\boldsymbol{k}\in S^+} cos\left(\frac{2\pi}{N}\boldsymbol{l}\cdot\boldsymbol{k}\right)}, \boldsymbol{l} \in \mathbb{Z}_N^d,$$

$$\chi_{\boldsymbol{h}}(\boldsymbol{g}) = \sum_{\boldsymbol{g}\in\mathbb{Z}_N^d} \exp\left(\frac{2\pi\imath}{N}\sum_{k=1}^{d} h_k g_k\right), \boldsymbol{h} \in \mathbb{Z}_N^d.$$

First, we consider 1-dimensional Cayley grids in Section 3.3.1, and we provide an estimation method for the relative error m_e defined in Eq. (3.1), along each edge of the graph. Second, we investigate the same estimation problem for a 2-dimensional Cayley grid in Section 3.3.2.

3.3.1 1-D thorus graph

In order to describe one dimensional thorus graphs, let us define $G = \mathbb{Z}_N$ and $S = \{-1, +1\}$, see Fig. 3.1 (left). The Cayley graph $\mathscr{G}(\mathbb{Z}_N, S)$ is 2-regular and the eigenvalues of Z are

$$\lambda_h = \frac{1}{1 - cos\left(\frac{2\pi h}{N}\right)}, h \in \mathbb{Z}_N.$$

It follows that the eigenvalues of $Z_\mathscr{G}$ are given by $\lambda_h/2, h \in \mathbb{Z}_N$, since $Z_\mathscr{G} = \frac{1}{d}Z$, where d is the uniform degree of each node. The associated generator of the simple random walk given by the transition matrix P, is defined as follows

$$\pi = \left(0, \frac{1}{2}, 0, \ldots, 0, \frac{1}{2}\right) \in [0,1]^N.$$

By means of explicit computations, we now prove that the relative error on one-dimensional Cayley graphs is bounded, as already shown in Corollary 3.3. By isotropy properties, namely swapping the nodes labels, it suffices to prove the latter for $i = 0, j = 1$.

Theorem 3.6. ***(Bound for relative error in ring graphs)*** *Given an undirected ring graph $\mathscr{G}$ with arbitrary orientation, and suppose $|\epsilon_e| \le \bar{\epsilon}$, for each edge $e \in \mathscr{E}$. Then,*

$$\left|x_i - x_j\right| \le 2\bar{\epsilon}, \text{ for each } (i,j) \in \mathscr{E}.$$

Proof. From Lemma 3.5, it holds

$$\begin{aligned} x_i - x_j &= \frac{1}{2N} \sum_{h \in \mathbb{Z}_N^*} \sum_{e \in \mathscr{E}} \frac{1}{1 - cos\left(\frac{2\pi h}{N}\right)} \left(e^{\frac{2\pi\imath h}{N}(j - s(e))} - e^{\frac{2\pi\imath h}{N}(j - t(e))} - e^{\frac{2\pi\imath h}{N}(i - s(e))} + e^{\frac{2\pi\imath h}{N}(i - t(e))}\right)\epsilon_e \\ &= \frac{1}{2N} \sum_{h \in \mathbb{Z}_N^*} \frac{1}{1 - cos\left(\frac{2\pi h}{N}\right)} \sum_{e \in \mathscr{E}} \left(e^{\frac{2\pi\imath h}{N} j} - e^{\frac{2\pi\imath h}{N} i}\right)\left(e^{-\frac{2\pi\imath h}{N} s(e)} - e^{-\frac{2\pi\imath h}{N} t(e)}\right)\epsilon_e. \end{aligned}$$

Now keep fixed the edge $(i, j) \in \mathscr{E}$, and every edge $e \in \mathscr{E}$ is parametrized by $t \in \mathbb{Z}_N$ as $e = (t, t+1)$. Our aim is to study the asymptotic behavior as N goes to infinity of the generic addend of the previous sum, defined as

$$C_{h,t} := \frac{1}{1 - cos\left(\frac{2\pi h}{N}\right)} \left(e^{\frac{2\pi\imath h}{N} j} - e^{\frac{2\pi\imath h}{N} i}\right)\left(e^{-\frac{2\pi\imath h}{N} t} - e^{-\frac{2\pi\imath h}{N}(t+1)}\right).$$

Furthermore we will study the asymptotic behavior of

$$x_i - x_j = \frac{1}{2N} \sum_{t \in \mathbb{Z}_N} \sum_{h \in \mathbb{Z}_N^*} C_{h,t} \epsilon_t = \frac{1}{2N} \sum_{t \in \mathbb{Z}_N} L_t \epsilon_t, \tag{3.7}$$

where

$$L_t := \sum_{h \in \mathbb{Z}_N^*} C_{h,t}. \tag{3.8}$$

As already said, by isotropy arguments, it suffices to take $i = 0$, $j = 1$, then

$$\begin{aligned} C_{h,t} &= \frac{1}{1 - cos\left(\frac{2\pi h}{N}\right)} \left(e^{\frac{2\pi\iota h}{N}} - 1\right) e^{-\frac{2\pi\iota h}{N} s(e)} \left(1 - e^{-\frac{2\pi\iota h}{N}}\right) = \frac{-e^{\frac{2\pi\iota h}{N}(-t)}}{1 - cos\left(\frac{2\pi h}{N}\right)} \left|e^{\frac{2\pi\iota h}{N}} - 1\right|^2 \\ &= \frac{-e^{\frac{2\pi\iota h}{N}(-t)}}{1 - cos\left(\frac{2\pi h}{N}\right)} \left(\left(\cos\left(\frac{2\pi h}{N}\right) - 1\right)^2 + \sin^2\left(\frac{2\pi h}{N}\right)\right) = -2e^{\frac{2\pi\iota h}{N}(-t)}. \end{aligned} \tag{3.9}$$

Notice that $C_{h,t} \in \mathbb{R}$ if and only if $\frac{2\pi h}{N}(-t) = k\pi, k \in \mathbb{Z}$ which is equivalent to ask

$$t \in \frac{N}{2h}\mathbb{Z}, h = 1, \ldots, N-1.$$

The latter is not necessarily true, but since $x_0 - x_1 \in \mathbb{R}$, it must hold that the quantity in Eq. (3.8) is a real number, i.e. $L_t \in \mathbb{R}$. It necessarily holds $\sum_{h \in \mathbb{Z}_N^*} \Im(C_{h,t}) = \sum_{h \in \mathbb{Z}_N^*} \sin\left(\frac{2\pi h}{N}(-t)\right) = 0$, since we started from an algebraic sum of elements of Z, which is a real-valued function. Thus, the following equality can be derived

$$|x_0 - x_1| = \left| \frac{1}{2N} \sum_{t=0}^{N-1} \sum_{h=1}^{N-1} C_{h,t} \epsilon_t \right| = \frac{1}{N} \left| \sum_{t=1}^{N-1} \sum_{h=1}^{N-1} e^{\frac{2\pi\iota h}{N} t} \epsilon_t + (N-1)\epsilon_{01} \right|. \tag{3.10}$$

Now we consider the N-th root of unity $z = e^{\frac{2\pi\iota t}{N}}$, such that $z^N = 1$, then

$$\sum_{h=1}^{N-1} z^h = \sum_{h=0}^{N} z^h - 1 = \frac{1 - z^N}{1 - z} - 1 = -1$$

thus the following sum can be computed as follows

$$\sum_{h=1}^{N-1} e^{\frac{2\pi\iota h}{N} t} = \begin{cases} -1, \text{if t} \neq 0, \\ N - 1, \text{if t} = 0. \end{cases}$$

Finally, Eq. (3.10) becomes

$$|x_0 - x_1| = \frac{1}{N}\left|\sum_{l=1}^{N-1}(-\epsilon_l) + (N-1)\epsilon_{10}\right| \leq \frac{1}{N}\left((N-1)\bar{\epsilon} + (N-1)\,|\epsilon_{10}|\right) \leq \frac{2(N-1)}{N}\bar{\epsilon} < 2\bar{\epsilon}.$$

The same can be achieved choosing any $(i,j) \in \mathscr{E}$, then the corresponding error on the relative edge remains bounded. □

From Theorem 3.6, it is clear that the relative error of the optimal estimator remains bounded as the number of nodes becomes asymptotically large. Therefore, a clear consequence is that measurements attained at edge graphically far away from $(0,1) \in \mathscr{E}$, do not affect the considered relative error. In other words, the amplifying factor defined in Eq. (3.2), progressively decays with distance.

3.3.2 2-D thorus graph

In this case $G = \mathbb{Z}_N^2$ ans $S = \{(-1,0),(1,0),(0,1),(0,-1)\}$, see Fig. 3.1 (center). The Cayley graph $\mathscr{G}(\mathbb{Z}_N^2, S)$ is 4-regular. Now the generator π associated to the reversible Markov Chain is the following

$$\pi = \left(0, \overset{2}{\frac{1}{4}}, 0, \ldots, 0, \overset{N-1}{\frac{1}{4}}, 0, \overset{N+1}{\frac{1}{4}}, 0, \ldots, 0, \overset{N^2-1}{\frac{1}{4}}, 0\right) \in \mathbb{Z}_N^2$$

or equivalently $\pi_j = \frac{1}{4}$, if $j = 1, N-1, N+1, N^2-1$, and $\pi_j = 0$ otherwise. Such vector π represents the probability distribution on the graph.

The eigenvalues of Z are

$$\lambda_{\boldsymbol{h}} = \frac{1}{1 - \frac{1}{2}cos\left(\frac{2\pi h_1}{N}\right) - \frac{1}{2}cos\left(\frac{2\pi h_2}{N}\right)}, \boldsymbol{h} = (h_1, h_2) \in \mathbb{Z}_N$$

and the corresponding ones related to $Z_{\mathscr{G}}$ are $4\lambda_{\boldsymbol{h}}, \boldsymbol{h} \in \mathbb{Z}_N^2$. In the following Lemma, we cast the relative error that we want to estimate in a simpler form.

Lemma 3.7. ***(Explicit relative error for** 2**D Cayley graphs)*** *Given a 2-dimensional Cayley grid, for each edge $(i,j) \in \mathscr{E}$, the relative error $m_{ij} = x_i - x_j + \epsilon_{ij}$ can be computed from the following*

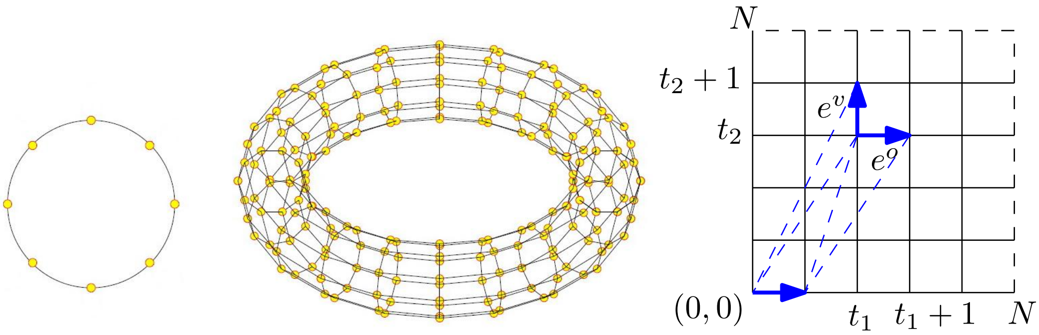


Figure 3.1: The figure shows different d-tori, one dimensional (left) and 2-dimensional (center) respectively. In the right figure, we focus on the 2D Cayley grid, and we fix $(\boldsymbol{i}, \boldsymbol{j}) = ((0,0),(1,0)) \in \mathscr{E}$. In order to compute the quantity in Eq. (3.12), we consider two different kind of edges, horizontal $e^o(t_1,t_2) = ((t_1,t_2),(t_1+1,t_2)) \in \mathbb{Z}_N^2$, and vertical $e^v(t_1,t_2) = ((t_1,t_2),(t_1,t_2+1))$, where the free parameters are $t_1, t_2 \in \mathbb{Z}$.

equality

$$x_{(0,0)} - x_{(1,0)} = \sum_{e \in \mathscr{E}} \Gamma_e \epsilon_e \tag{3.11}$$

where the amplifying factor defined in Eq. (3.2) *is, by abuse of notation,* $\Gamma_e = \Gamma_{((0,0),(1,0)),e}$

$$\Gamma_e := \frac{1}{N^2} \sum_{\boldsymbol{h} \neq \boldsymbol{0}} C_{\boldsymbol{h},e}$$

and we parametrize the edge as $e = (s(e), t(e))$, $s(e) = (t_1, t_2) \in \mathbb{Z}_N^2$, *defining* $C_{\boldsymbol{h},e} := C^o_{\boldsymbol{h},\boldsymbol{t}}$ *if the edge e is horizontal*[1]*, and* $C_{\boldsymbol{h},e} := C^v_{\boldsymbol{h},\boldsymbol{t}}$ *if e is vertical, with*

$$C^o_{\boldsymbol{h},\boldsymbol{t}} = \frac{\cos(\frac{2\pi}{N}\boldsymbol{ht})\left(\cos\left(\frac{2\pi h_1}{N}\right) - 1\right)}{2 - \cos\left(\frac{2\pi h_1}{N}\right) - \cos\left(\frac{2\pi h_2}{N}\right)},$$

$$C^v_{\boldsymbol{h},\boldsymbol{t}} = -\frac{1}{2}\frac{\Re\left(e^{-\frac{2\pi\imath}{N}\boldsymbol{ht}}\left(1 - e^{\frac{2\pi\imath}{N}h_1}\right)\left(1 - e^{-\frac{2\pi\imath}{N}h_2}\right)\right)}{2 - \cos\left(\frac{2\pi h_1}{N}\right) - \cos\left(\frac{2\pi h_2}{N}\right)}.$$

[1] Edge $e = (s(e), t(e)) \in \mathbb{Z}_N^2$ is said to be horizontal if $s(e) - t(e) = (\pm 1, 0)$.

Proof. Starting from Eq. (3.4), we obtain

$$
\begin{aligned}
x_{\boldsymbol{i}} - x_{\boldsymbol{j}} &= \frac{1}{4N^2} \sum_{\boldsymbol{h} \in \mathbb{Z}_N^2 - \{\mathbf{0}\}} \lambda_{\boldsymbol{h}} \sum_{e \in \mathscr{E}} \left(e^{\frac{2\pi\imath \boldsymbol{h}}{N}(\boldsymbol{j}-\boldsymbol{s}(e))} - e^{\frac{2\pi\imath \boldsymbol{h}}{N}(\boldsymbol{j}-\boldsymbol{t}(e))} - e^{\frac{2\pi\imath \boldsymbol{h}}{N}(\boldsymbol{i}-\boldsymbol{s}(e))} + e^{\frac{2\pi\imath \boldsymbol{h}}{N}(\boldsymbol{i}-\boldsymbol{t}(e))} \right) \epsilon_e \\
&= \frac{1}{4N^2} \sum_{\boldsymbol{h} \in \mathbb{Z}_N^2 - \{\mathbf{0}\}} \frac{1}{1 - \frac{1}{2} cos\left(\frac{2\pi h_1}{N}\right) - \frac{1}{2} cos\left(\frac{2\pi h_2}{N}\right)} \sum_{e \in \mathscr{E}} \left(e^{\frac{2\pi\imath \boldsymbol{h}}{N}\boldsymbol{j}} - e^{\frac{2\pi\imath \boldsymbol{h}}{N}\boldsymbol{i}} \right) \left(e^{-\frac{2\pi\imath \boldsymbol{h}}{N}\boldsymbol{s}(e)} - e^{-\frac{2\pi\imath \boldsymbol{h}}{N}\boldsymbol{t}(e)} \right) \epsilon_e.
\end{aligned}
\tag{3.12}
$$

Let us keep fixed the edges $(\boldsymbol{i}, \boldsymbol{j}), e \in \mathscr{E}$. Define the distance $d(\boldsymbol{h}, \boldsymbol{k}) = |h_1 - k_1| + |h_2 - k_2|, \boldsymbol{h}, \boldsymbol{k} \in \mathbb{Z}_N^2$. Remind that $\boldsymbol{i} - \boldsymbol{j}, \boldsymbol{s}(e) - \boldsymbol{t}(e) \in S$, i.e. $d(\boldsymbol{i}, \boldsymbol{j}) = d(\boldsymbol{s}(e), \boldsymbol{t}(e)) = 1$. W.l.o.g. two cases have to be considered: firstly the edges $(\boldsymbol{i}, \boldsymbol{j}), (\boldsymbol{s}(e), \boldsymbol{t}(e))$ have the same direction. Secondly they are perpendicular. Due to the intrinsic isotropy of the system, suppose edge $(\boldsymbol{i}, \boldsymbol{j})$ to be horizontal and shifted to the origin without loss of generality: $\boldsymbol{i} = (0,0), \boldsymbol{j} = (1,0)$. As depicted in Fig. 3.1, the generic considered edge, horizontal e^o and vertical e^v respectively, is parametrized as follows

$$
\begin{aligned}
&e^o(t_1, t_2) = ((t_1, t_2), (t_1 + 1, t_2)), \\
&e^v(t_1, t_2) = ((t_1, t_2), (t_1, t_2 + 1)).
\end{aligned}
$$

As already stated, let us recall

$$
C_{\boldsymbol{h},e} = \frac{1}{2} \frac{\Re\left(\left(e^{\frac{2\pi\imath \boldsymbol{h}}{N}\boldsymbol{j}} - e^{\frac{2\pi\imath \boldsymbol{h}}{N}\boldsymbol{i}} \right) \left(e^{-\frac{2\pi\imath \boldsymbol{h}}{N}\boldsymbol{s}(e)} - e^{-\frac{2\pi\imath \boldsymbol{h}}{N}\boldsymbol{t}(e)} \right) \right)}{2 - cos\left(\frac{2\pi h_1}{N}\right) - cos\left(\frac{2\pi h_2}{N}\right)}
$$

and in particular, if edge e is horizontal, we have

$$
\begin{aligned}
C^o_{\boldsymbol{h},\boldsymbol{t}} &= \frac{1}{2} \frac{\Re\left(e^{\frac{2\pi\imath}{N} h_1} - 1 \right) \left(e^{-\frac{2\pi\imath}{N}(h_1 t_1 + h_2 t_2)} - e^{-\frac{2\pi\imath}{N}(h_1(t_1+1) + h_2 t_2)} \right)}{2 - \cos\left(\frac{2\pi h_1}{N}\right) - \cos\left(\frac{2\pi h_2}{N}\right)} \\
&= -\frac{1}{2} \frac{\Re\left(e^{-\frac{2\pi\imath}{N}(h_1 t_1 + h_2 t_2)} \left(1 - e^{\frac{2\pi\imath}{N} h_1}\right) \left(1 - e^{-\frac{2\pi\imath}{N} h_1}\right) \right)}{2 - \cos\left(\frac{2\pi h_1}{N}\right) - \cos\left(\frac{2\pi h_2}{N}\right)} = -\frac{1}{2} \frac{\Re\left(e^{-\frac{2\pi\imath}{N} \boldsymbol{h}\boldsymbol{t}} \left| e^{\frac{2\pi\imath}{N} h_1} - 1 \right|^2 \right)}{2 - \cos\left(\frac{2\pi h_1}{N}\right) - \cos\left(\frac{2\pi h_2}{N}\right)} \\
&= \frac{\cos(\frac{2\pi}{N} \boldsymbol{h}\boldsymbol{t}) \left(\cos\left(\frac{2\pi h_1}{N}\right) - 1 \right)}{2 - \cos\left(\frac{2\pi h_1}{N}\right) - \cos\left(\frac{2\pi h_2}{N}\right)}.
\end{aligned}
$$

otherwise, if e is vertical, $C^v_{\boldsymbol{h},\boldsymbol{t}}$ can be analogously obtained. Note that, if $h_1 = h_2$, then $C^v_{\boldsymbol{h},\boldsymbol{t}} = \frac{1}{2} e^{-\frac{2\pi i}{N} \boldsymbol{h}\boldsymbol{s}(e)}$. □

Conjecture 1. ***(Asymptotic error propagation in 2D Cayley grids)*** *Suppose we are given a* $2D$ *Cayley grid* $\mathscr{G}$*, the estimated relative position* $\boldsymbol{x} \in \mathbb{R}^{\mathcal{V}}$*, and the noise vector* $\boldsymbol{\epsilon} \in \mathbb{R}^{\mathcal{E}}$ *with* $\|\epsilon\|_\infty \leq \bar{\epsilon}$ *for any realization. Then*

$$|x_{\boldsymbol{i}} - x_{\boldsymbol{j}}| \leq \mathrm{c},$$

where c *is a positive constant.*

In the following we give some hint to justify the latter conjecture. Equation (3.11) can be split as follows

$$x_{(0,0)} - x_{(1,0)} = \frac{1}{N^2} \sum_{(h_1,h_2)\neq(0,0)} \sum_{t_1=0}^{N-1} \sum_{t_2=0}^{N-1} C^o_{\boldsymbol{h},\boldsymbol{t}} \epsilon_{e^o} + \frac{1}{N^2} \sum_{(h_1,h_2)\neq(0,0)} \sum_{t_1=0}^{N-1} \sum_{t_2=0}^{N-1} C^v_{\boldsymbol{h},\boldsymbol{t}} \epsilon_{e^v}$$

and we proceed fixing $\boldsymbol{t}$, thus the goal is to estimate the two quantities

$$\frac{1}{N^2} \sum_{\boldsymbol{h}\neq\boldsymbol{0}} C^o_{\boldsymbol{h},\boldsymbol{t}} \epsilon_{\boldsymbol{t}} \sim O_{\boldsymbol{t}}, \quad \frac{1}{N^2} \sum_{\boldsymbol{h}\neq\boldsymbol{0}} C^v_{\boldsymbol{h},\boldsymbol{t}} \epsilon_{\boldsymbol{t}} \sim V_{\boldsymbol{t}}$$

and finally give a estimate of

$$x_{(0,0)} - x_{(1,0)} \sim \sum_{\boldsymbol{t}} (O_{\boldsymbol{t}} + V_{\boldsymbol{t}}) \epsilon_{\boldsymbol{t}}. \tag{3.13}$$

In this formulation the amplifying factor is $\Gamma_e = O_{\boldsymbol{t}} + V_{\boldsymbol{t}}$ since e is parametrized by $\boldsymbol{t} \in \mathbb{Z}^2_N$, $\boldsymbol{t} \neq \boldsymbol{0}$. Note that, for our purposes, it suffices to prove that the amplifying factor decays as $\boldsymbol{t}$ increases. Passing to the limit $N \to \infty$, consider the following change of variable: $x = \frac{h_1}{N}$, $y = \frac{h_2}{N}$. Thus, as the number of nodes N goes to infinity, we start considering the addends corresponding to horizontal edges

$$\sum_{\boldsymbol{t}} O_{\boldsymbol{t}} \sim \sum_{\boldsymbol{t}} \epsilon_{e^o} \int_0^1 \int_0^1 \frac{\cos(2\pi(xt_1 + yt_2))\,(\cos(2\pi x) - 1)}{2 - \cos(2\pi x) - \cos(2\pi y)} dxdy \tag{3.14}$$

Given t_1, t_2 as parameters, study now the behavior of the function $f : [0,1] \times [0,1] \to \mathbb{R}$, defined as follows

$$f_{\boldsymbol{t}}(x,y) = \frac{\cos(2\pi(xt_1 + yt_2))\,(\cos(2\pi x) - 1)}{2 - \cos(2\pi x) - \cos(2\pi y)}, \boldsymbol{t} = (t_1, t_2). \tag{3.15}$$

The following Lemma describes some properties of the latter function, as shown also in Fig.

3.2.

Lemma 3.8. ***(Properties of the approximating function)*** *Consider $f_{\boldsymbol{t}}$ defined in Eq.* (3.15), *then*

(i) $|f_{\boldsymbol{t}}(x,y)| \le 1$, *for every* $(x,y) \in [0,1]^2$*;*

(ii) $f_{\boldsymbol{t}}$ *is periodic with period* 1*;*

(iii) *symmetry with respect to* $(0,0)$*:* $f_{\boldsymbol{t}}(-x,-y) = f_{\boldsymbol{t}}(x,y)$*;*

(iv) *symmetry with respect to* $(\frac{1}{2},\frac{1}{2})$*:* $f_{\boldsymbol{t}}(1-x,1-y) = f_{\boldsymbol{t}}(x,y)$*;*

(v) $f_{\boldsymbol{t}}$ *has an integrable singularity in* $(0,0)$.

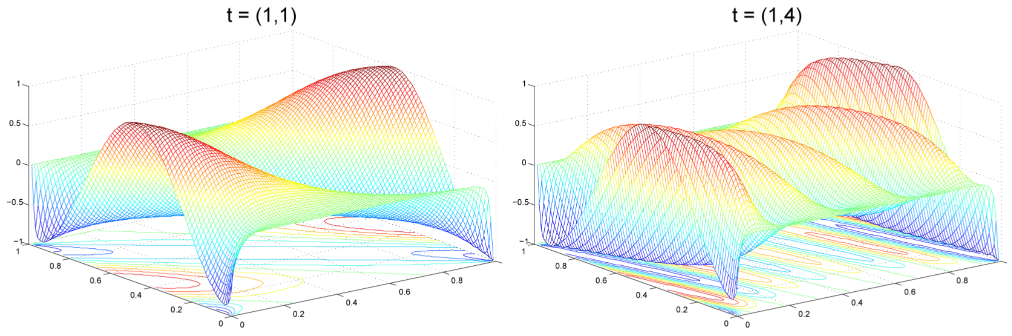


Figure 3.2: 3D plot and contour lines of the two variable function f, with $\boldsymbol{t} = (1,4)$ (left), and $\boldsymbol{t} = (1,4)$ (right). It is clear that t_1 and t_2 are the frequency of oscillations along the x-axis and y-axis respectively.

First, let us neglect the noise by considering the intrinsic upper bound we assumed, i.e. the worst-case noise $|\epsilon_e| \le \bar{\epsilon}$. Formally

$$\left|\frac{1}{N^2}\sum_{\boldsymbol{h}\neq\boldsymbol{0}} C^o_{\boldsymbol{h},\boldsymbol{t}}\epsilon_{e^o}\right| \le \frac{\bar{\epsilon}}{N^2}\sum_{\boldsymbol{h}\neq\boldsymbol{0}} |C^o_{\boldsymbol{h},\boldsymbol{t}}|.$$

From Fig. 3.3, it is clear that the relative error diverges, thus we have to consider the initial expression in Eq. (3.13), in order to get a sharp upper bound.

3.4 Conclusions

In this Chapter, we provide estimates for component on each edge of the relative error of the optimal least squares estimator. Concerning the exemplary class of graphs called Abelian

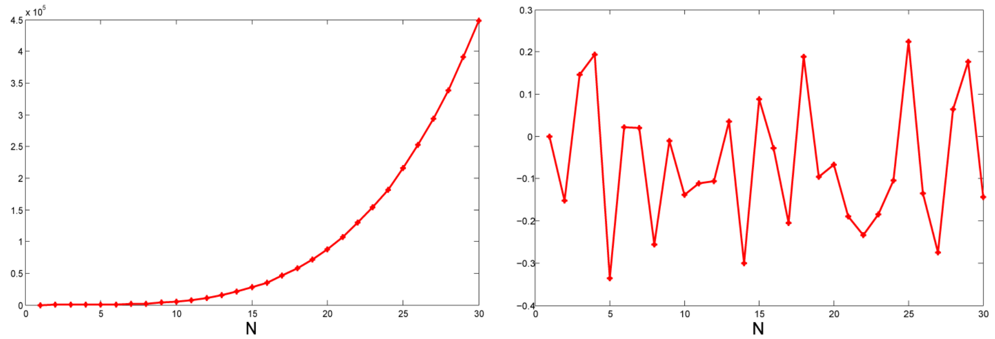


Figure 3.3: The Figure (left) show how the quantity $\sum_{\boldsymbol{h}\neq\mathbf{0}} |C^{o}_{\boldsymbol{h},\boldsymbol{t}}|/N^2$ goes to infinity as N asymptotically increases in a 2D Cayley grid. As opposite, in right plot it is shown the asymptotic behavior of $x_{(0,0)} - x_{(1,0)}$ as defined in Eq. (3.13), setting the maximum noise $\bar{\epsilon} = 1$. Note that, as N is asymptotically large, the considered quantity remains bounded.

Cayley graphs, we explicitly compute such misplacement of the optimum with respect to correct position vector, by means of the eigenvalues of the network. We used the Markov Chains theory as mathematical tools, considering the simple random walk associated to the graph, and then we exploited the DFT to compute the relative error in terms of the eigenvalues of the network. We first considered a one-dimensional Cayley grid, and we obtained analytically the boundedness of the relative error with an asymptotically large size of the network. We conjecture that also in 2-dimensional Cayley grid, such quantity is bounded, providing numerical experiments and analytical properties that suggest such conclusion. Of course, further efforts may be needed to deeply analyze the 2-dimensional case. A possible approach may be interpreting the expression in Eq. (3.14), as the real part of the Fourier Transform of a certain function. The same has to be done for the addends corresponding to the vertical edges in grid. Deriving this FT several times and applying standard properties, we could find an estimate in terms of the parameter $\boldsymbol{t}$, and then an estimate of the relative error may be provided. Finally, among the author's intentions, a detailed study must be carried out in the case of Cayley grids of dimension $d \geq 3$.

4 Synchronous distributed calibration algorithms

This Chapter deals with the problem of the angular calibration for a network of cameras, namely the problem of estimating a common orientation reference frame. In the proposed set–up each camera obtains noisy measurements of its relative orientation with respect to some other cameras. The set of measurements can be described by a graph having the cameras as nodes and having an edge between two cameras if a relative orientation measurement is available. We propose a novel two-step algorithm based on a choice of a basis for the set of graph cycles. The first step consists in computing a set of integer numbers, which provides a first rough estimate of the orientations. The second step exploits this information to build up a suitable quadratic minimization problem. Two actual implementations, corresponding to two different basis of cycles, are described and compared in terms of the worst–case scenario. Finally, through numerical simulations the algorithm is compared with another algorithm proposed in the literature for solving the same problem.

4.1 Introduction

In a network of cameras one of the most crucial problems is calibration. For each camera this consists in understanding what is its position and orientation with respect to a global and common reference coordinate system. The importance of this information is clear in case the camera network is used for instance to track an external mobile object. Indeed, in this case, if the object is exiting from the sensing region of the i-th camera and it is entering in the sensing region of the j-th camera, then, in order to avoid to loose the target, the camera i has to communicate to the camera j where it has to move in order to see the object. It is manifest that both cameras must share the same coordinate system in order this to be possible.

Usually, calibration is set off-line by human operators, or by a centralized unit. In any case it is usually a rather time consuming stage of the system setting. The algorithm we propose in this Chapter aims to complete this task autonomously, and requires no or very limited centralized coordination. This also allows the possibility to re–calibrate periodically and this is useful especially in case when some of the cameras are mobile.

The calibration problem over Euclidean spaces has recently been studied by Barooah and Hespanha in a great detail (see Barooah and Hespanha [2005], Barooah et al. [2006], Barooah and Hespanha [2007]). The problem considered there is the localization using noisy relative measurements, namely determining the coordinates of a set of vectors in a Euclidean space starting from the knowledge of some noisy differences of those vectors.

The cameras calibration problem is similar in the sense that each camera can be characterized by a position and an orientation in space. Some well–known methods in computer vision permit to obtain quite easily and efficiently relative positions and orientations of pairs of cameras whose sensing regions overlap. Then the problem that has to be solved is to determine, from these relative positions and orientations, the position and the orientation of the cameras with respect to a common reference coordinate system. Cameras calibration can be casted into an optimization problem (or a consensus) over the manifold $SE(3)$. Under certain assumptions one can show that this problem can decoupled into the estimation of the position and the estimation of the orientation. The latter can be seen as an optimization (or a consensus) over the manifold $SO(3)$. Moreover, under some further assumptions, the calibration over $SO(3)$ can be reduced to the calibration over the simpler manifold $SO(2)$.

This class of problems has already attracted much attention in the last years. In Sarlette [2009b], Sarlette and Sepulchre [2009b] a consensus algorithm on $SO(2)$ based on the gradient flow of a potential defined using the chordal distance is studied. In Tron and Vidal [2009b] a similar approach based on the geodesic distance is proposed in order to study the more general calibration problem on $SE(3)$. The drawback associated with both these approaches is that, in case we have many cameras, the proposed potentials exhibit a great number of local minima.

In Piovan et al. [2011b] the problem of calibration on $SO(2)$ is considered when measurements of relative orientations are affected by additive noise. The authors propose a procedure to obtain a new set of relative orientations which is ensured to sum up to multiples of 2π over a chosen family of cycles, an idea which closely resembles what we propose in the present Chapter. The new set of relative orientations is then spread along a spanning tree to obtain an estimate of the orientations. Finally, in Singer [2011] an estimation algorithm is proposed for a model of the measurements in which some of them are ideal, while others are completely random. The estimate of the orientations is obtained via the computation of the eigenvector associated with the largest eigenvalue of a suitable Hermitian matrix.

The algorithm proposed in this Chapter is based on a non-convex optimization problem, as in Sarlette [2009b], Tron and Vidal [2009b]. We restrict ourselves to the simple case of calibration over $SO(2)$, and the cost to be minimized is based on the geodesic distance over this manifold. The set of available relative measurements is described by a graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, where the set of nodes $\mathscr{V}$ represents the set of cameras and where the set of edges $\mathscr{E}$ represents the available relative orientation measurements between pairs of cameras. Our main idea is to break the estimation problem into two parts: first we estimate a combinatorial object, which is a set of integers, each associated with an edge in $\mathscr{E}$. Intuitively, these integers take care of the fact that measurement noises along the cycles in the graph do not sum up to 0, in general. Once this is done, the original optimization problem over a manifold can be reduced to a quadratic optimization problem, which can be easily solved using classic algorithms. The idea of using cycles has already been proposed in Russell et al. [2010] in the context of localization over Euclidean spaces in order to improve the quality of the estimates. In fact, we propose two versions of the algorithm. One version is based on sets of cycles associated with spanning trees. Another version instead is based on sets of minimal cycles. Notice finally that the proposed method is consistent in the sense that, if there is no measurement noise, the solution given by the algorithm coincides with the true one.

This research has previously partially appeared in Borra et al. [2012a], and it is here enriched both in theoretical depth and as for simulative comparison with the existing literature.

We refer to Section 2.1 for several useful preliminary graph theoretical definitions and results. This Chapter is organized as follows: in Section 4.2 and Section 4.3 we respectively formulate the optimization problem to be solved and we explain how it relates with the cameras calibration problem. Section 4.4 is devoted to the description of the proposed algorithm and to a worst–case analysis of its performance. In Section 4.5 it is shown how to distribute the algorithm over the network for the two particular choices of the set of cycles mentioned above, while in Section 4.6 we compare their performance. Finally, Section 4.7 proposes numerical simulations and a comparison with the algorithm proposed in Piovan et al. [2011b]. Section 4.8 draws the conclusions.

4.2 Problem Formulation

The estimation problem over $SO(2)$ we want to study can be described as follows. Assume we have a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with an orientation $s : \mathcal{E} \to \mathcal{V}$ and $t : \mathcal{E} \to \mathcal{V}$. With each node $v \in \mathcal{V}$ of the graph we associate an angle $\bar{\theta}_v \in [-\pi, \pi)$ and similarly with each edge $e \in \mathcal{E}$ we associate an angle $\eta_e \in [-\pi, \pi)$. We call the angle $\bar{\theta}_v$ the *orientation* of v. Our aim is to find an estimate of the angles $\bar{\theta}_v$, $v \in \mathcal{V}$, from the knowledge of η_e, $e \in \mathcal{E}$, knowing that these represent a measure of the *relative orientation* among $s(e)$ and $t(e)$, in the sense that

$$\eta_e = (\bar{\theta}_{s(e)} - \bar{\theta}_{t(e)} + \varepsilon_e)_{2\pi} \tag{4.1}$$

where $\varepsilon_e \in [-\pi, \pi)$ are measurement noise terms, which have to be considered small and independent, and where $(x)_{2\pi} := x - 2\pi q_{2\pi}(x)$ with $q_{2\pi}(x) := \lfloor \frac{x+\pi}{2\pi} \rfloor, x \in \mathbb{R}$. Notice that the function $q_{2\pi}(x)$ is a quantizer such that $x \in [-\pi, \pi)$ if and only if $q_{2\pi}(x) = 0$, so that $(x)_{2\pi} = x$. If we stack the angle $\bar{\theta}_v$ and η_e in suitable column vectors we can rewrite (4.1) in the compact form

$$\boldsymbol{\eta} = (B\bar{\boldsymbol{\theta}} + \boldsymbol{\varepsilon})_{2\pi}, \tag{4.2}$$

where $(\cdot)_{2\pi}$ is done componentwise, and B is the incidence matrix of $\mathcal{G}$, cfr. Section 2.1.

The estimator is in principle a function $\hat{\boldsymbol{\theta}} : [-\pi, \pi)^{\mathcal{E}} \to [-\pi, \pi)^{\mathcal{V}}$ mapping the available data $\boldsymbol{\eta}$ into an estimate $\hat{\boldsymbol{\theta}}(\boldsymbol{\eta})$ of $\bar{\boldsymbol{\theta}}$. Its performance can be evaluated using the index

$$W(\hat{\boldsymbol{\theta}}) = \frac{1}{N} \| (\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})_{2\pi} \|^2 = \frac{1}{N} \sum_{v=1}^{N} (\hat{\boldsymbol{\theta}}_v - \bar{\boldsymbol{\theta}}_v)^2_{2\pi}. \tag{4.3}$$

Notice that W is in principle a function of $\bar{\boldsymbol{\theta}}, \boldsymbol{\varepsilon}$, namely $W = W(\bar{\boldsymbol{\theta}}, \boldsymbol{\varepsilon})$. If we have a probabilistic model of $\bar{\boldsymbol{\theta}}$ and $\boldsymbol{\varepsilon}$ the considered index is the expected value of W.

Remark 4.1. *The definition of W is reasonable since the values $\bar{\theta}_v$'s and $\hat{\theta}_v$'s are angles, so they are coincident if they differ of integer multiples of 2π, namely if $(\hat{\theta}_v - \bar{\theta}_v)_{2\pi} = 0$.*

The estimator here proposed is based on the minimization of the following least–square cost

$$V(\boldsymbol{\theta}) = \sum_{e\in\mathscr{E}} (\theta_{s(e)} - \theta_{t(e)} - \eta_e)^2_{2\pi} = \|(B\boldsymbol{\theta} - \boldsymbol{\eta})_{2\pi}\|^2, \tag{4.4}$$

through which one aims to find a set of estimates $\hat{\theta}_1, \ldots, \hat{\theta}_N$ whose differences along the edges fit the measurements η_e.

Remark 4.2. *Notice that $V(\boldsymbol{\theta} + \alpha\mathbb{1}) = V(\boldsymbol{\theta})$, $\forall\boldsymbol{\theta} \in [-\pi,\pi)^{\mathscr{V}}$ and $\forall\alpha \in \mathbb{R}$, where $\mathbb{1}$ is the vector with all entries equal to 1. This feature naturally arises from the fact that the cost is build on* relative *measurements and is unavoidable unless we impose a further constraint. In fact, we will assume in the sequel that node 1 is the anchor and knows its orientation, which, with no loss of generality, can be assumed to be zero, i.e. $\bar{\theta}_1 = 0$. By adding this constraint, namely imposing that $\hat{\theta}_1 = 0$, we can avoid this source of ambiguity.*

Remark 4.3. *Minimizing the cost $V(\boldsymbol{\theta})$ seems to be reasonable, since, in case of noiseless measurements, we have $V(\boldsymbol{\theta}) = 0$ and $\theta_1 = 0$ if and only if $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}}$. In other words $V(\boldsymbol{\theta})$ has only one global minimum for $\boldsymbol{\theta} \in (-\pi,\pi]^{\mathscr{V}}$ such that $\theta_1 = 0$, which is $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}}$. However, $V(\boldsymbol{\theta})$ has, even in the noiseless case, multiple local minima, as it will be shown in the next example. The same problem occurs if we choose a slightly different cost based on the chordal distance, as it has been shown in Sarlette [2009b]. In order to avoid such local minima, the classic approach is to initialize the gradient based minimization algorithm in a suitable region in such a way that the convergence to the global minimum is ensured. A deep analysis of this region for a large class of manifolds is done in Tron et al. [2011], but it is beyond the scope of this work. Our approach is different, and it is motivated by the following simple example.*

Example 4.4. *Consider the ring graph with 3 agents in figure 4.1, and assume for sake of simplicity that $\bar{\theta}_1 = \bar{\theta}_2 = \bar{\theta}_3 = 0$. Consider the ideal noiseless case, so that $\eta_{12} = \eta_{23} = \eta_{31} = 0$. Assuming that we have fixed $\theta_1 = 0$, we have to find $\theta_2, \theta_3 \in (-\pi,\pi]$ such to minimize the cost*

$$V(\theta_1,\theta_2,\theta_3) = \theta_2^2 + \theta_3^2 + (\theta_2 - \theta_3)^2_{2\pi} = \begin{cases} \theta_2^2 + \theta_3^2 + (\theta_2 - \theta_3)^2 & \theta_2 - \theta_3 \in (-\pi,\pi], \\ \theta_2^2 + \theta_3^2 + (\theta_2 - \theta_3 - 2\pi)^2 & \theta_2 - \theta_3 \geq \pi, \\ \theta_2^2 + \theta_3^2 + (\theta_2 - \theta_3 + 2\pi)^2 & \theta_2 - \theta_3 < -\pi. \end{cases} \tag{4.5}$$

It can be verified that this cost has three local minima $\theta_2 = \theta_3 = 0$, $\theta_2 = \frac{2}{3}\pi, \theta_3 = -\frac{2}{3}\pi$ and $\theta_2 = -\frac{2}{3}\pi, \theta_3 = \frac{2}{3}\pi$ being $\theta_2 = \theta_3 = 0$ the global minimum in which $V = 0$ while the other two *are only local minima where* $V = \frac{4}{3}\pi^2$.

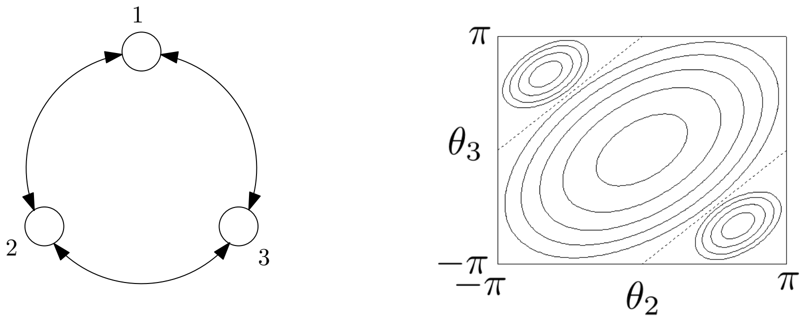


Figure 4.1: A simple ring with 3 agents. On the right panel, the three regions in which $[-\pi,\pi)$ is partitioned, with the contour lines of the quadratic functions in (4.5).

4.3 Cameras calibration

The proposed problem has a certain interest in applications such as calibration Tron and Vidal [2009b] or orientation localization Piovan et al. [2011b] of networks of cameras. In this Section we describe the calibration problem and we relate it to the problem stated above.

Consider a group of cameras perceiving some environment for surveillance purposes. Each camera is modeled as an *ideal pin-hole* device, which consists in a plane, called the image plane, and a point, called the optical center. A point in the environment is sensed by the camera through its radial projection, through the optical center, on the image plane. The sensing region of the camera is the set of points for which this projection exists. To model the perception of a point we endow each camera with a local reference frame, denoted by Σ_v, so that a point in the sensing region of camera v is identified, by v, as a set of coordinates in Σ_v.

Fix now an external reference frame Σ_0. Without loss of generality we can assume that its origin lies on the ground, which is assumed to be a plane, and that it has one axis perpendicular to the ground and pointing “upward”. In such a reference frame, the *pose* of the camera v is identified by an element $(g_v, R_v) \in SE(3) = \mathbb{R}^3 \times SO(3)$. The set $SO(3) = \{R \in \mathbb{R}^{3\times3} : R^T R = I, \det(R) = +1\}$, is the set of rotations in $3D$, while $\mathbb{R}^3$ denotes here the set of translations. The quantity (g_v, R_v) maps the local reference frame Σ_v into the external reference frame Σ_0. More precisely, given a point in the environment whose coordinates in Σ_0 and Σ_v are, respectively, p_0, $p_v \in \mathbb{R}^3$, then

$$p_0 = R_v p_v + g_v. \tag{4.6}$$

For coordination purposes, it is necessary that the cameras obtain as good as possible estimates of their (g_v, R_v). Assume in fact that an agent is exiting from the sensing region of camera v and is entering in that of camera u, and call p_v its coordinates in Σ_v. Then u can easily find the coordinates of the agent in its reference frame Σ_u using twice (4.6) to obtain

$$p_u = R_u^T R_v p_v + R_u^T (g_v - g_u). \tag{4.7}$$

Thus, if each camera has a good estimate of its pose w.r.t. a common external reference frame, they can exchange information on the position of agents in the environment they are monitoring.

An estimate of the g_v's can be obtained through the consensus-like optimization procedures based on relative measurements as proposed in Barooah et al. [2006]. In this Chapter, instead, we concentrate on the estimate of R_v, which we call from now on the *rotational calibration problem*.

In order to estimate the R_v's, the cameras communicate and exchange information with some of the others. The admissible communications are modeled through an undirected communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in which $\mathcal{V} = \{1, \dots, N\}$ is the set of nodes, and $\mathcal{E}$ is the set of edges. An edge $\{v, u\} \in \mathcal{E}$ exists if and only if the cameras u and v are able to exchange information and, moreover, their sensing regions overlap.

The mutual information used to achieved rotational calibration is the *relative orientation* among pairs of cameras. Given the poses (g_v, R_v) and (g_u, R_u) of cameras v and u, we define $R_{vu} := R_u^T R_v$ the relative orientation of v with respect to u. Notice that R_{vu} appears in (4.7) when relating the coordinates of a point in Σ_u given those in Σ_v, and that $R_{uv} = R_{vu}^T$.

Assume now $\{u, v\} \in \mathcal{E}$. Then the quantities R_{uv} and R_{vu}, as well as the relative translations $g_u - g_v$ and $g_v - g_u$, can be obtained by the cameras v and u through the so–called eight points algorithm Ma et al. [2003]. In general, this relative orientation is corrupted by noise. In our setting, the two cameras can thus only compute the following noisy version of R_{vu}

$$\tilde{R}_{vu} = R_u^T N_{vu} R_v, \tag{4.8}$$

where $N_{vu} \in SO(3)$ is a rotational error. Since the two cameras compute these quantities on the basis of the same dataset, then we can assume that $\tilde{R}_{uv} = \tilde{R}_{vu}^T$, which implies $N_{uv} = N_{vu}^T$.

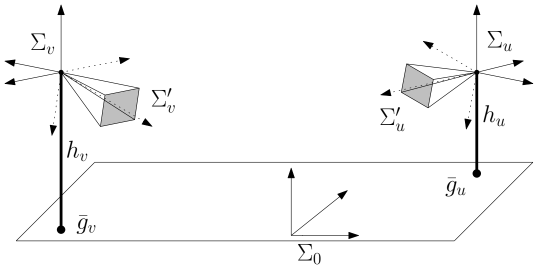


Figure 4.2: Description of a network of cameras satisfying the assumptions. It is shown the external reference frame Σ_0 and, for each camera, the "natural" local reference frame Σ'_v (in dotted lines) and the chosen local reference frame Σ_v (in solid lines).

Rotational calibration in $SO(2)$ is related to the previous problem when the following assumptions hold:

- each camera is deployed in the environment at a known height with respect to the ground, which is assumed to be a plane;
- each camera knows the perpendicular to the ground, and sets its reference frame Σ_v to have one specific axis (common to all the cameras) along this direction and pointing "upward".

Remark 4.5. *The natural reference frame Σ'_v of camera v is the optical reference frame, which has two axis lying on the image plane and the z-axis perpendicular to it and crossing the optical center Ma et al. [2003]. Implicitly, we assume that the camera v knows its tilt and roll angles, so that it is also able to compute the rotational matrix which transforms Σ'_v in Σ_v.*

In the general case, the element $(g_v, R_v) \in SE(3)$ possesses six degrees of freedom in the sense that g_v and R_v can be parameterized using three independent scalars, and three angles, respectively. The previous assumptions correspond to fixing one of such scalars (the height with respect to the ground) and two of such angles (tilt and roll), in the sense that the pose

(g_v, R_v) of the camera v is such that, $\forall v \in \mathcal{V}$,

$$g_v = \begin{bmatrix} \bar{g}_v \\ h_v \end{bmatrix} \qquad R_v = R(\bar{\theta}_v) := \begin{bmatrix} \cos\bar{\theta}_v & -\sin\bar{\theta}_v & 0 \\ \sin\bar{\theta}_v & \cos\bar{\theta}_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.9}$$

Observe that $\bar{g}_v \in \mathbb{R}^2$ is the position, projected on the ground, of camera v, $h_v \in \mathbb{R}$ is the known height of the origin of the reference frame Σ_v, and the angle $\bar{\theta}_v \in [-\pi, \pi)$ represents the rotation about the z axis needed to align Σ_v with Σ_0.

Another consequence of the previous assumptions and the exact knowledge of the tilt and roll angles is that the eight point algorithm naturally provides a matrix $\tilde{R}_{vu}$ having the structure described in (4.9), namely there exists $\eta_{vu} \in [-\pi, \pi)$ such that $\tilde{R}_{vu} = R(\eta_{vu})$. By exploiting now the relations $R(\theta)R(\varphi) = R(\varphi + \theta)$ and $R(\theta)^T = R(\theta)^{-1} = R(-\theta)$, the model in (4.8) for measurement of the relative orientation can be rewritten, as in (4.1), as $\eta_{vu} = (\bar{\theta}_v - \bar{\theta}_u + \varepsilon_{vu})_{2\pi}$, where $\varepsilon_{vu} \in [-\pi, \pi)$ is such that $N_{vu} = R(\varepsilon_{vu})$.

Given two elements $R_1, R_2 \in SO(3)$, their *geodesic* can be be seen as the shortest path joining R_1 and R_2 on the manifold, and the *geodesic distance* $d(R_1, R_2)$ between R_1 and R_2 is simply the length of their geodesic (see do Carmo [1992] for more formal definitions and properties). For elements of $SO(3)$ of the type given in (4.9) we have

$$d(R(\theta_1), R(\theta_2)) = |(\theta_1 - \theta_2)_{2\pi}|$$

Consequently, one can see that

$$\sum_{\{u,v\} \in \mathcal{E}} d(R(\theta_u)^T R(\theta_v),\ R(\eta_{uv}))^2 = V(\boldsymbol{\theta})$$

namely the cost function proposed in Section 4.2 corresponds to summing up the squared geodesic distances between $R(\theta_u)^T R(\theta_v)$ and the measurements $R(\eta_{uv})$ along all the edges of the graph. Notice that this cost already appeared in Tron and Vidal [2009b].

4.4 Description of the proposed algorithm

4.4.1 The regions of convexity of the cost functions

For each $\boldsymbol{K} \in \mathbb{Z}^{\mathcal{E}}$ define the region

$$R_{\boldsymbol{K}}(\boldsymbol{\eta}) := \{\boldsymbol{\theta} \in [-\pi,\pi)^{\mathcal{V}} : A\boldsymbol{\theta} - \boldsymbol{\eta} - 2\pi\boldsymbol{K} \in [-\pi,\pi)^{\mathcal{E}}\}. \tag{4.10}$$

These regions are convex and form a partition of $[-\pi,\pi)^{\mathcal{V}}$. Most of them can actually be empty. It can be seen that, if $\boldsymbol{\theta} \in R_{\boldsymbol{K}}(\boldsymbol{\eta})$, and only for these points, we have

$$V(\boldsymbol{\theta}) = \|B\boldsymbol{\theta} - \boldsymbol{\eta} - 2\pi\boldsymbol{K}\|^2,$$

therefore $V(\boldsymbol{\theta})$ is purely quadratic and convex in $R_{\boldsymbol{K}}(\boldsymbol{\eta})$. For this reason, in each $R_{\boldsymbol{K}}(\boldsymbol{\eta})$ there can be at most one local minimum of $V(\boldsymbol{\theta})$.

Let $\bar{\boldsymbol{K}} \in \mathbb{Z}^{\mathcal{E}}$ be such that $\bar{\boldsymbol{\theta}} \in R_{\bar{\boldsymbol{K}}}(\boldsymbol{\eta})$, namely let $\bar{\boldsymbol{K}}$ identify the region in which the vector of orientations $\bar{\boldsymbol{\theta}}$ lies. Equation (4.10) yields to

$$q_{2\pi}(A\bar{\boldsymbol{\theta}} - \boldsymbol{\eta}) = \bar{\boldsymbol{K}}.$$

On the other hand, from (4.2) we have that

$$\boldsymbol{\eta} = B\bar{\boldsymbol{\theta}} + \boldsymbol{\varepsilon} - 2\pi q_{2\pi}(A\bar{\boldsymbol{\theta}} + \boldsymbol{\varepsilon}).$$

Since we assumed that $\boldsymbol{\varepsilon} \in [-\pi,\pi)^{\mathcal{E}}$, we can argue that

$$\bar{\boldsymbol{K}} = q_{2\pi}(B\bar{\boldsymbol{\theta}} - \boldsymbol{\eta}) = q_{2\pi}\Big(-\boldsymbol{\varepsilon} + 2\pi q_{2\pi}(B\bar{\boldsymbol{\theta}} + \boldsymbol{\varepsilon})\Big) = q_{2\pi}(B\bar{\boldsymbol{\theta}} + \boldsymbol{\varepsilon})$$

where we used the fact that $q_{2\pi}(x + 2\pi k) = q_{2\pi}(x) + k$, if $k \in \mathbb{Z}$. Consequently, the vector of measurements can be expressed as

$$\boldsymbol{\eta} = B\bar{\boldsymbol{\theta}} + \boldsymbol{\varepsilon} - 2\pi\bar{\boldsymbol{K}}. \tag{4.11}$$

4.4.2 Estimation of the convexity region

The idea which inspires the algorithm we are going to propose is first to obtain an estimate $\hat{\boldsymbol{K}}$ of $\bar{\boldsymbol{K}}$ and then to obtain $\hat{\boldsymbol{\theta}}$ by minimizing the reshaped cost

$$V_{\hat{\boldsymbol{K}}}(\boldsymbol{\theta}) := \|B\boldsymbol{\theta} - \boldsymbol{\eta} - 2\pi\hat{\boldsymbol{K}}\|_2^2 \tag{4.12}$$

which corresponds to restricting $V(\boldsymbol{\theta})$ to the region $R_{\hat{\boldsymbol{K}}}(\boldsymbol{\eta})$ and then extending the quadratic form to $\mathbb{R}^{\mathcal{V}}$.

Example 4.6. *Let's consider again the Example 4.4. Assume we are still in the ideal noiseless case, so that* $\eta_{12} = \eta_{23} = \eta_{31} = 0$*. Since we fixed* $\theta_1 = 0$*, then* V *is function only of* θ_2, θ_3*. As shown above* $V(0, \theta_2, \theta_3)$ *is quadratic in the three regions corresponding to the case* $\theta_2 - \theta_3 \in (-\pi, \pi]$*, to the case* $\theta_2 - \theta_3 \geq \pi$ *and to the case* $\theta_2 - \theta_3 < -\pi$*. These three regions correspond to* $R_{\boldsymbol{K}}(0)$ *in the three cases* $\boldsymbol{K} = (K_{12}, K_{23}, K_{31}) = (0,0,0)$*,* $\boldsymbol{K} = (0,-1,0)$*, and* $\boldsymbol{K} = (0,1,0)$*. The other regions are empty. This is depicted in right panel on figure 4.1.*

The idea of how to estimate $\bar{\boldsymbol{K}}$ comes from the observation that the relative differences of the actual orientations $\bar{\theta}_v$ along a cycle must necessarily sum up to a multiple of 2π. More precisely, let h be an oriented cycle and let $\boldsymbol{r}_h$ be its representative vector as explained in Section 2.1. Then, from (4.11), using the fact that $\boldsymbol{r}_h B = 0$, we obtain that

$$\boldsymbol{r}_h\boldsymbol{\eta} + 2\pi\boldsymbol{r}_h\bar{\boldsymbol{K}} = \boldsymbol{r}_h\boldsymbol{\varepsilon}$$

and so we can argue that

$$q_{2\pi}(\boldsymbol{r}_h\boldsymbol{\eta}) + \boldsymbol{r}_h\bar{\boldsymbol{K}} = q_{2\pi}(\boldsymbol{r}_h\boldsymbol{\eta} + 2\pi\boldsymbol{r}_h\bar{\boldsymbol{K}}) = q_{2\pi}(\boldsymbol{r}_h\boldsymbol{\varepsilon}).$$

On the other hand, if it happens that the algebraic sum of the noise along the cycle h is below π in modulus, namely if $|\boldsymbol{r}_h\boldsymbol{\varepsilon}| < \pi$, then we obtain that $q_{2\pi}(\boldsymbol{r}_h\boldsymbol{\varepsilon}) = 0$. This yields

$$\boldsymbol{r}_h\bar{\boldsymbol{K}} = -q_{2\pi}(\boldsymbol{r}_h\boldsymbol{\eta}). \tag{4.13}$$

Observe that in this case $\boldsymbol{r}_h\bar{\boldsymbol{K}}$ could be exactly computed on the basis of the measurements $\boldsymbol{\eta}$ along the cycle h. This would suggest to define implicitly the estimation $\hat{\boldsymbol{K}}$ by imposing $\boldsymbol{r}_h\hat{\boldsymbol{K}} = -q_{2\pi}(\boldsymbol{r}_h\boldsymbol{\eta})$ for any cycle h. However, the resulting system of equations would not be solvable in general due to the non–coherence of the noises along cycles. What can be done is to choose a basis for Γ, and impose the constraint on the corresponding cycles.

Inspired by the previous observations, assume that a generic basis of Γ is given, and let $R \in \mathbb{Z}^{(\mathscr{E} \setminus \mathscr{E}_{\mathscr{T}}) \times \mathscr{E}}$ the matrix whose rows are the elements of the basis. Denote by $\hat{\boldsymbol{K}}_R$ the estimate of $\bar{\boldsymbol{K}}$ we want to obtain using R. More precisely, we want to find $\hat{\boldsymbol{K}}_R$ satisfying the following system of equations on $\mathbb{Z}^{\mathscr{E}}$ in matrix form

$$R\hat{\boldsymbol{K}}_R = -q_{2\pi}(R\boldsymbol{\eta}). \tag{4.14}$$

By Propositions 2.1 and 2.2 we obtain that the general solution of this equation is

$$\hat{\boldsymbol{K}}_R = -X_R q_{2\pi}(R\boldsymbol{\eta}) + A\boldsymbol{h} \tag{4.15}$$

where X_R is the matrix, introduced in Proposition 2.2 such that $RX_R = I$ and $\boldsymbol{h}$ is any column vector in $\mathbb{Z}^{\mathscr{V}}$.

Once $\hat{\boldsymbol{K}}_R$ has been selected, the final solution $\hat{\boldsymbol{\theta}}_R$ (which depends on the choice of R) is determined by minimizing the cost

$$V_{\hat{\boldsymbol{K}}}(\boldsymbol{\theta}) = \|B\boldsymbol{\theta} - \boldsymbol{\eta} - 2\pi\hat{\boldsymbol{K}}_R\|_2^2.$$

The solution of the previous minimization is the solution of the following linear equation

$$B^T B\hat{\boldsymbol{\theta}}_R = B^T\boldsymbol{\eta} + 2\pi B^T\hat{\boldsymbol{K}}_R. \tag{4.16}$$

As done above, in order to avoid the non uniqueness due to the kernel of A, we fix $\theta_1 = 0$. Then consider the vector $\boldsymbol{\xi} \in \mathbb{R}^{\mathscr{V}}$ defined by $\xi_1 = 1$ and $\xi_v = 0$ for any $v \neq 1$. It is well-known Aldous and Fill that there exists a symmetric matrix $Z \in \mathbb{R}^{\mathscr{V} \times \mathscr{V}}$ such that

$$\begin{cases} ZB^T B = I - \mathbb{1}\boldsymbol{\xi}^T, \\ Z\boldsymbol{\xi} = 0. \end{cases} \tag{4.17}$$

This matrix is called the Green matrix associated to the graph $\mathscr{G}$, see Section 2.1. Using the properties of Z and the expression of $\hat{\boldsymbol{K}}_R$ we obtain

$$\hat{\boldsymbol{\theta}}_R = ZB^T\boldsymbol{\eta} + 2\pi ZB^T\hat{\boldsymbol{K}}_R = ZB^T\boldsymbol{\eta} - 2\pi ZB^T X_R q_{2\pi}(R\boldsymbol{\eta}) + 2\pi ZB^T B\boldsymbol{h}.$$

Recall that we look for an estimate in $[-\pi,\pi)^{\mathcal{V}}$, thus the nodes need to project onto this set the computed solution. This can be done entry–wise, and corresponds to the choice

$$\boldsymbol{h} := -q_{2\pi}\left(ZB^T\boldsymbol{\eta} - 2\pi ZB^T X_R q_{2\pi}(R\boldsymbol{\eta})\right).$$

Using the definition of Green matrix, one can show that $h_1 = 0$, thus obtaining

$$\hat{\boldsymbol{\theta}}_R = ZB^T\boldsymbol{\eta} - 2\pi ZB^T X_R q_{2\pi}(R\boldsymbol{\eta}) + 2\pi\boldsymbol{h} = \left(ZB^T\boldsymbol{\eta} - 2\pi ZB^T X_R q_{2\pi}(R\boldsymbol{\eta})\right)_{2\pi} \in [-\pi,\pi)^{\mathcal{V}}. \tag{4.18}$$

A simple continuity argument then shows that, when the threshold $\bar{\varepsilon}$ tends to 0, the estimate $\hat{\boldsymbol{\theta}}_R$ converges to $\bar{\boldsymbol{\theta}}$. In other terms, if the noise is small, we have a guarantee that our solution is close to the true $\bar{\boldsymbol{\theta}}$. In the following Sections we formally prove this statement.

4.4.3 Performance analysis of the proposed algorithm

As mentioned in Section 4.2, in order to evaluate the performance of the proposed algorithm, we need to estimate the index $W = \frac{1}{N}\|(\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})_{2\pi}\|^2$ defined in (4.3). Observe that, since $\boldsymbol{\eta} = B\bar{\boldsymbol{\theta}} + \boldsymbol{\varepsilon} - 2\pi\bar{\boldsymbol{K}}$ and $X_R q_{2\pi}(R\boldsymbol{\eta}) = -\hat{\boldsymbol{K}}_R + B\boldsymbol{h}$ then (4.18) can be rewritten

$$\begin{aligned}\hat{\boldsymbol{\theta}}_R &= ZB^T\boldsymbol{\eta} - 2\pi ZB^T X_R q_{2\pi}(R\boldsymbol{\eta}) + 2\pi\boldsymbol{h}\\ &= ZB^T B\bar{\boldsymbol{\theta}} + ZB^T\boldsymbol{\varepsilon} - 2\pi ZB^T\bar{\boldsymbol{K}} + 2\pi ZB^T\hat{\boldsymbol{K}}_R - 2\pi ZB^T B\boldsymbol{h} + 2\pi\boldsymbol{h}\end{aligned}$$

Since $\bar{\theta}_1 = 0$ and $h_1 = 0$ we can argue that

$$\hat{\boldsymbol{\theta}}_R = \bar{\boldsymbol{\theta}} + ZB^T\boldsymbol{\varepsilon} + 2\pi ZB^T(\hat{\boldsymbol{K}}_R - \bar{\boldsymbol{K}}), \tag{4.19}$$

which shows that the computed estimate differs from the vector of actual orientations by a term depending only on the measurement noise. The second term is instead related to the ability to correctly identify the region in which $\bar{\boldsymbol{\theta}}$ lies.

The previous equation yields

$$W = \frac{1}{N}\|(\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})_{2\pi}\|^2 = \frac{1}{N}\|(ZB^T\boldsymbol{\varepsilon} + 2\pi ZB^T(\hat{\boldsymbol{K}}_R - \bar{\boldsymbol{K}}))_{2\pi}\|^2.$$

This equality implies that, in case $R\hat{\boldsymbol{K}}_R = R\bar{\boldsymbol{K}}$, namely if we have that $\hat{\boldsymbol{K}}_R = \bar{\boldsymbol{K}} + B\boldsymbol{h}$ for some $\boldsymbol{h}$, then

$$\begin{aligned} W &= \frac{1}{N}\|(ZB^T\boldsymbol{\varepsilon} + 2\pi ZB^T B\boldsymbol{h})_{2\pi}\|^2 = \frac{1}{N}\|(ZB^T\boldsymbol{\varepsilon} + 2\pi\boldsymbol{h})_{2\pi}\|^2 \\ &= \frac{1}{N}\|(ZB^T\boldsymbol{\varepsilon})_{2\pi}\|^2 \leq \frac{1}{N}\|ZB^T\boldsymbol{\varepsilon}\|^2. \end{aligned}$$

Notice that the term $\frac{1}{N}\|ZB^T\boldsymbol{\varepsilon}\|^2$ has been already studied in the literature (see Barooah and Hespanha [2005, 2007], Lovisari et al. [2012]), since it characterizes the performance of the calibration algorithm over vector spaces. In these papers the electrical analogy is used. The graph $\mathcal{G}$ is considered as an electrical network where there is a resistance of 1 Ohm along all the edges in $\mathcal{G}$. If we denote by $\mathcal{R}_{v1}$ the effective resistance among the node v and the anchor node 1, then it can be shown that

$$\frac{1}{N}\mathbb{E}\|ZB^T\boldsymbol{\varepsilon}\|^2 = \frac{\sigma^2}{N}\sum_{v\in\mathcal{V}}\mathcal{R}_{v1},$$

where we are assuming that the components of the noise vector $\boldsymbol{\varepsilon}$ have zero mean and variance σ^2. This result allows to obtain the estimate of $\frac{1}{N}\mathbb{E}\|ZB^T\boldsymbol{\varepsilon}\|^2$ as a function of the number of nodes for some families of graphs. For example, if the graph is a line, then

$$\frac{1}{N}\sum_{v\in\mathcal{V}}\mathcal{R}_{v1} \approx N.$$

If the graph is a 2D grid, then

$$\frac{1}{N}\sum_{v\in\mathcal{V}}\mathcal{R}_{v1} \approx \log N,$$

while if the graph is a grid in dimension greater or equal to 3, then $\frac{1}{N}\sum_{v\in\mathcal{V}}\mathcal{R}_{v1}$ is bounded from above by a constant independent of N. This qualitative behavior can be of interest for the designer. We will show this feature in the examples proposed in Section 4.7.

Remark 4.7. *Observe that, since*

$$R\hat{\boldsymbol{K}}_R = -q_{2\pi}(R\boldsymbol{\eta}) = -q_{2\pi}(R\boldsymbol{\varepsilon} - 2\pi R\bar{\boldsymbol{K}}) = R\bar{\boldsymbol{K}} - q_{2\pi}(R\boldsymbol{\varepsilon}),$$

then $R\hat{\boldsymbol{K}}_R = R\bar{\boldsymbol{K}}$ occurs if the components of the vector $R\boldsymbol{\varepsilon}$ belong to $[-\pi,\pi)$, and so if $||R\boldsymbol{\varepsilon}||_\infty < \pi$ [1]*. If the components of the noise vector $\boldsymbol{\varepsilon}$ are supported in $[-\bar{\varepsilon},\bar{\varepsilon}]$, since $||R\boldsymbol{\varepsilon}||_\infty \le \bar{\varepsilon}||R||_\infty$, then*

$$||R||_\infty < \frac{\pi}{\bar{\varepsilon}}$$

implies that $R\hat{\boldsymbol{K}}_R = R\bar{\boldsymbol{K}}$. Notice finally that, if the rows of R are obtained from a set of closed paths, then $||R||_\infty$ is simply the maximum length of the paths in this set. For this reason it is convenient to choose R whose rows are obtained from paths of minimum length. This fact inspires the algorithms proposed in the next Section.

4.5 Distributed algorithms for rotational calibration

In this Section we will propose two possible distributed implementations of the rotational calibration algorithm proposed in the previous Section. Observe that the proposed algorithm is based on two steps. The first consists in the estimation of $\bar{\boldsymbol{K}}$ based on formulas (4.14) and (4.15). The second step consists in the estimation of $\bar{\boldsymbol{\theta}}$ based on formula (4.16). Distributed implementations of the second step have already been proposed in the literature (see Barooah and Hespanha [2007], Bolognani et al. [2010]). We propose here two distributed implementations of the first step, which are based on two ways to select the set of cycles as a basis of Γ.

Fundamental cycles Fix a spanning tree $\mathcal{T} = (\mathcal{V}, \mathcal{E}_\mathcal{T})$ of $\mathcal{G}$. Order arbitrarily the edges in $\mathcal{E} \setminus \mathcal{E}_\mathcal{T}$ as $e_1,\dots,e_{M-N+1}$ and consider cycles $h_1,\dots,h_{M-N+1}$ in $\mathcal{G}$ such that for each i we have that $\mathrm{supp}(h_i) \subseteq \mathcal{E}_\mathcal{T} \cup \{e_i\}$ and $\boldsymbol{r}_{h_i}(e_i) = 1$. In words, cycle h_i is constructed with the edges in $\mathcal{E}_\mathcal{T} \cup \{e_i\}$ and oriented in such a way that $\boldsymbol{r}_{h_i}(e_i) = 1$. Such cycles are called *($\mathcal{T}$-)fundamental cycles.*

[1] We recall that, if $x \in \mathbb{R}^n$, then $||x||_\infty = \max_i\{|x_i|\}$ and if $M \in \mathbb{R}^{n\times m}$, then $||M||_\infty = \max_i\{\sum_j |M_{ij}|\}$. According to these definitions we have that $||Mx||_\infty \le ||M||_\infty ||x||_\infty$.

Minimal cycles Fix again a spanning tree $\mathscr{T} = (\mathscr{V}, \mathscr{E}_{\mathscr{T}})$ of $\mathscr{G}$. Another possible construction is the following iterative one:

- Among all cycles whose edges are all in $\mathscr{E}_{\mathscr{T}}$ except one, choose one of minimal length. Call it h_1 and call e_1 the only edge in h_1 which is not in $\mathscr{E}_{\mathscr{T}}$.
- Suppose edges $e_1, \dots, e_i$ and cycles $h_1, \dots, h_i$ have been selected. Among all cycles whose edges are all in $\mathscr{E}_{\mathscr{T}} \cup \{e_1, \dots e_i\}$ except one, choose one of minimal length. Call it h_{i+1}, and call e_{i+1} the only edge in h_{i+1} which is not in $\mathscr{E}_{\mathscr{T}} \cup \{e_1, \dots e_i\}$. Such cycles are called *($\mathscr{T}$-)minimal cycles.*

Both the fundamental and the minimal cycles provide a basis of Γ as shown in the next Lemma.

Lemma 4.8. *Fix a spanning tree $\mathscr{T} = (\mathscr{V}, \mathscr{E}_{\mathscr{T}})$ of $\mathscr{G}$. The subsets of Γ:*

$$\{r_h \,|\, h\ \mathscr{T}\text{-fundamental cycle}\},\ \{r_h \,|\, h\ \mathscr{T}\text{-minimal cycle}\}$$

are both $\mathbb{Z}$-basis of Γ.

Proof. The fact that the rows in $\mathbb{Z}^{\mathscr{E}}$ obtained from a set of fundamental cycles form a basis of Γ follows from the the part (b) of the proof of Proposition 2.1. The fact the rows in $\mathbb{Z}^{\mathscr{E}}$ obtained from the minimal cycles form a basis of Γ is consequence of the following arguments. We order the edges putting first the edges in $\mathscr{T}$ and the edges $e_1, \dots, e_{M-N+1}$. Let R_F be the matrix having rows $\boldsymbol{r}_{h_1^F}, \dots, \boldsymbol{r}_{h_{M-N+1}^F}$, and R_M be the matrix having rows $\boldsymbol{r}_{h_1^M}, \dots, \boldsymbol{r}_{h_{M-N+1}^M}$, where h_i^F and h_i^M are the fundamental and the minimal cycles, respectively. Then, from the proposed construction, R_F and R_M can be partitioned as

$$R_F = [R_F' \;\; I], \qquad R_M = [R_M' \;\; T],$$

where I is the identity matrix of dimension $M-N+1$, $T \in \mathbb{Z}^{(\mathscr{E}\setminus\mathscr{E}_{\mathscr{T}})\times\mathscr{E}}$ is a square lower triangular matrix with 1 on the diagonal and $R_F', R_M' \in \mathbb{Z}^{(\mathscr{E}\setminus\mathscr{E}_{\mathscr{T}})\times\mathscr{E}_{\mathscr{T}}}$. Observe that $T^{-1} \in \mathbb{Z}^{(\mathscr{E}\setminus\mathscr{E}_{\mathscr{T}})\times(\mathscr{E}\setminus\mathscr{E}_{\mathscr{T}})}$ and, since the rows of R_F generate Γ while the rows of R_F belong to Γ, then there exists a matrix $Z \in \mathbb{Z}^{(\mathscr{E}\setminus\mathscr{E}_{\mathscr{T}})\times(\mathscr{E}\setminus\mathscr{E}_{\mathscr{T}})}$ such that $R_M = ZR_F$. As a consequence, $Z = T$ and $R_M' = ZR_F' = TR_F'$. Finally, we can argue that $R_F = T^{-1}R_M$, which shows that the rows of R_M generate the same submodule generated by the rows of R_F which is Γ. □

4.5.1 The Tree-algorithm

In this distributed implementation the agents are supposed to be the nodes and the edges composing the graph $\mathscr{G}$. Fix a spanning graph $\mathscr{T}$ rooted at the anchor node 1. In order to obtain $\hat{\boldsymbol{K}}$, we need first to obtain the value $\boldsymbol{r}_{h_i}\boldsymbol{\eta}$, for each cycle h belonging to the set of the fundamental cycles. This value will be stored by the agent associated to the edge e_i, which is the edge in h_i not belonging to $\mathscr{T}$. For the computation of $\boldsymbol{r}_{h_i}\boldsymbol{\eta}$, first the measurements are propagated along the tree starting from the root. In other words, given a node v and called $f(v)$ its father, we set $\hat{\theta}_{FE,1} = 0$ and

$$\hat{\theta}_{FE,v} = \hat{\theta}_{FE,f(v)} + \eta_{v,f(v)}. \tag{4.20}$$

As a side effect, we also obtain a first estimate $\hat{\boldsymbol{\theta}}_{FE}$ of $\bar{\boldsymbol{\theta}}$. Then each edge e_i not belonging to $\mathscr{T}$ can compute $\boldsymbol{r}_{h_i}\boldsymbol{\eta}$ as

$$\boldsymbol{r}_{h_i}\boldsymbol{\eta} = \hat{\theta}_{FE,s(e_i)} - \hat{\theta}_{FE,t(e_i)} + \eta_{e_i}.$$

Now we build $\hat{\boldsymbol{K}}$ as follows. For each edge e in $\mathscr{T}$, we assign $\hat{\boldsymbol{K}}_e = 0$. Instead, for each edge e_i not belonging $\mathscr{T}$, we assign

$$\hat{\boldsymbol{K}}_{e_i} = -q_{2\pi}(\boldsymbol{r}_{h_i}\boldsymbol{\eta}).$$

Observe that this assignment of $\hat{\boldsymbol{K}}$ ensures that, for all the fundamental cycles h_i, it holds

$$\boldsymbol{r}_{h_i}\hat{\boldsymbol{K}} = \hat{\boldsymbol{K}}_{e_i} = -q_{2\pi}(\boldsymbol{r}_{h_i}\boldsymbol{\eta})$$

that is needed to correctly estimate $\bar{\boldsymbol{K}}$. Once $\hat{\boldsymbol{K}}$ is obtained, the nodes can compute in a distributed way the components $\left(\operatorname{argmin}_{\boldsymbol{\theta}} \|B\boldsymbol{\theta} - \boldsymbol{\eta} - 2\pi\hat{\boldsymbol{K}}\|\right)_{2\pi}$.

4.5.2 Minimal cycles-algorithm

The second algorithm exploits the construction of a set of minimal cycles for the graph. The procedure in this case is not fully distributed, in the sense that we must assume that each *minimal cycle* is associated with an agent which carries on the computations associated with this cycle.

Fix a spanning tree $\mathscr{T} = (\mathscr{V}, \mathscr{E}_{\mathscr{T}})$ of $\mathscr{G}$ and let $h_1, \ldots, h_{M-N+1}$ be the set of minimal cycles. Benote by $e_1, \ldots, e_{M-N+1}$ the associated edges forming the set $\mathscr{E} \setminus \mathscr{T}$ ordered in the way de-

Algorithm 1 Tree-Algorithm

```
(Input variables)
$\bar{\theta}_1$, value of the anchor;
$\eta_e, e = 1, \dots, M$;
$\mathscr{T}$ spanning tree;

(Step A: first estimate $\hat{\boldsymbol{\theta}}_{FE}$)
$\hat{\theta}_{FE,1} = 0$;
for $i = 1, \dots, N$ do
    for $j = 2, \dots, N$ do
        if $j$ is a son of $i$ in $\mathscr{T}$ then $\hat{\theta}_{FE,j} = \hat{\theta}_{FE,i} + \eta_{j,i}$;

(Step B: estimate $\hat{\boldsymbol{K}}$)
for $e \in \mathscr{E}$ do
    $\hat{\boldsymbol{K}}_e = q_{2\pi}(\hat{\theta}_{FE,s(e)} - \hat{\theta}_{FE,t(e)} - \eta_e)$;

(Step C: second estimate $\hat{\boldsymbol{\theta}}$)
compute $\left(\text{argmin}_{\boldsymbol{\theta}} \|B\boldsymbol{\theta} - \boldsymbol{\eta} - 2\pi\hat{\boldsymbol{K}}\|\right)_{2\pi}$ s.t. $\theta_1 = 0$.
```

scribed above. As in the previous case, assign $\hat{\boldsymbol{K}}_e = 0$ for any $e \in \mathscr{T}$. Consider now the remaining edges $e_1, e_2, \dots, e_{M-N+1}$. We know that cycle h_1 has edges in $\mathscr{T} \cup \{e_1\}$. Since $\hat{\boldsymbol{K}}_e = 0, \forall e \neq e_1$ for any e in the support of h_1, we can assign

$$\hat{\boldsymbol{K}}_{e_1} = -q(\boldsymbol{r}_{h_1}\boldsymbol{\eta}).$$

Then, we proceed iteratively, and at the i-th iteration we already computed $\hat{\boldsymbol{K}}_{e_1}, \dots, \hat{\boldsymbol{K}}_{e_{i-1}}$. In order to assign $\hat{\boldsymbol{K}}_{e_i}$, observe that the following relation must be satisfied

$$\boldsymbol{r}_{h_i}\hat{\boldsymbol{K}} = -q_{2\pi}(\boldsymbol{r}_{h_i}\boldsymbol{\eta}).$$

Therefore, we derive

$$\hat{\boldsymbol{K}}_{e_i} = -q_{2\pi}(\boldsymbol{r}_{h_i}\boldsymbol{\eta}) - \sum_{j \neq i} \boldsymbol{r}_{h_i}(e_j)\hat{\boldsymbol{K}}_{e_j}.$$

In order to achieve this computation, the agent associated with the cycle h_i has to know η_{e_j} and $\hat{\boldsymbol{K}}_{e_j}$ for each edge e_j belonging to h_i. As a consequence, this agent needs to receive the values of $\hat{\boldsymbol{K}}_{e_j}$ from the agents associated to adjacent cycles, before being able to compute $\hat{\boldsymbol{K}}_{e_i}$. After the estimation of $\hat{\boldsymbol{K}}$, the remaining steps are the same as those of the previous algorithm.

This second algorithm allows much better performance than the first one, but it requires

Algorithm 2 Minimal cycles-algorithm

```
(Input variables)
$\eta_e, e = 1, \dots, M$;
$\mathcal{T}$ spanning tree;
$\boldsymbol{r}_1, \dots, \boldsymbol{r}_{M-N+1}$ minimal cycles set;

(Step A: computation of $\boldsymbol{b} = -q_{2\pi}(R\boldsymbol{\eta})$)
for $i = 1, \dots, M-N+1$ do $b_{h_i} = -q_{2\pi}(\boldsymbol{r}_{h_i}\boldsymbol{\eta})$;

(Step B: estimate $\hat{\boldsymbol{K}}$)
for $e \in \mathcal{E}_{\mathcal{T}}$ do $\hat{\boldsymbol{K}}_e = 0$;
for $i = 1, \dots, M-N+1$ do $\hat{\boldsymbol{K}}_{\bar{e}} = b_{h_i} - \sum_{j=1}^{i-1} \boldsymbol{r}_{h_i}(e_j)\hat{\boldsymbol{K}}_{e_j}$;

(Step C: second estimate $\hat{\boldsymbol{\theta}}$)
compute $\left(\mathrm{argmin}_{\boldsymbol{\theta}} \|B\boldsymbol{\theta} - \boldsymbol{\eta} - 2\pi\hat{\boldsymbol{K}}\|\right)_{2\pi}$ s.t. $\theta_1 = 0$.
```

a greater amount of communication and collaboration among nodes. In fact, as already mentioned, it is required that an agent is associated with each cycle, and it must know all the measurements along the edges of its cycle.

Let us illustrate how the *Minimal cycles-algorithm* works with the following simple example.

Example 4.9. *Consider the simple graph in Figure 4.3. For such a graph, the minimal cycles are $h_1, \dots, h_5$, and the edges are* $1, \dots, 13$. *Assume that* $b_1 = 1$, $b_2 = 2$, $b_3 = b_4 = b_5 = 0$, *where* $\boldsymbol{b} = -q_{2\pi}(R\boldsymbol{\eta})$. *Edges* $1, \dots, 8$ *form a spanning tree $\mathcal{T}$ of the graph. First of all, set* $\hat{\boldsymbol{K}}_1 = \dots \hat{\boldsymbol{K}}_8 = 0$. *Now, cycles h_1 and h_2 are made of edges of the tree apart from the edges* 9 *and* 10 *respectively. Thus h_1 sets* $\hat{\boldsymbol{K}}_9 = 1$, *while h_2 sets* $\hat{\boldsymbol{K}}_{10} = -2$, *since the direction of* 10 *is incoherent with the orientation of h_2. Once this is done, we know the value of $\hat{\boldsymbol{K}}$ along all the edges of h_3 and h_4, apart from* 12 *and* 11 *respectively. In order to guarantee that the sum over the cycles h_3 and h_4 is equal to b_3 and b_4 respectively, we can derive* $\hat{\boldsymbol{K}}_{12} = 2$ *and* $\hat{\boldsymbol{K}}_{11} = 1$. *Finally, all the edges* $e \neq 13$ *of h_5 have have their corresponding $\hat{K}_e$ already assigned, so it suffices to set* $\hat{\boldsymbol{K}}_{13} = -3$. *Now the sum of $\hat{\boldsymbol{K}}$ over the five minimal cycles corresponds to $\boldsymbol{b}$ entry–wise.*

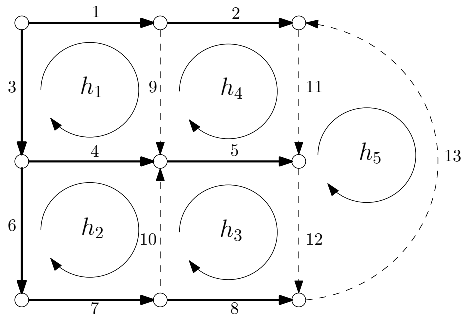


Figure 4.3: A simple graph to show how the second algorithm works.

4.6 Resilience against measurement noise for different graph topologies

In this Section we compare the two algorithms we have proposed for several different graph topologies. We concentrate on grid-like topologies, since they can be used to model real networks of cameras. In order to draw the comparison, consider the graphs shown in figure 4.4. In both cases we have a line–like graph with many nodes deployed along one dimension, and the chosen spanning trees are shown in thick lines. They are rooted on the anchor on the most left-top node. The set of the minimal cycles basis is simply the set of squares which form the graph.

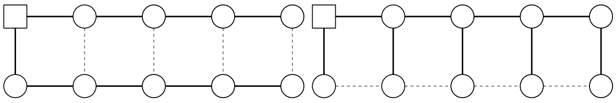


Figure 4.4: Two examples of spanning trees for a line–like graph. The proposed algorithms work in a similar manner for the one on the right, while the *Minimal cycles-algorithm* is far more effective for the one on the left.

For the graph on the left, if we take the tree and we add the last edge on the right we obtain a cycle with maximum length N. On the contrary, the minimal cycles are of length $L_0 = 4$. As an immediate consequence, the *Minimal cycles-algorithm* has much better performance since the upper bound $\bar{\varepsilon} < \frac{\pi}{4}$ is independent on the number of nodes. On the other hand, in order the *Tree-algorithm*, to produce a good estimate $\hat{\boldsymbol{K}}$, the magnitude of the noise should decrease with the dimension of the graph.

If we consider instead the spanning tree on the right, we can see that the fundamental cycles have length 4 as well, since the spanning tree is chosen in a much better way. In this case, the two algorithms have the same performance.

If we consider the ring graph in Figure 4.5, we can easily see that there is only one minimal cycle. Here the two proposed algorithms coincide. In such a case, the *Tree-algorithm* is better, since it is easier to implement and completely distributed, it requires less information on the topology of the network, as well as less communications.

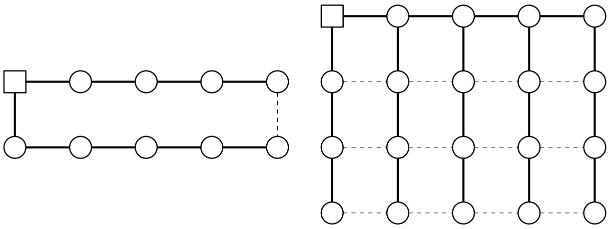


Figure 4.5: On the left a ring graph, for which the two algorithms have the same performance. On the right, a grid graph.

Indeed, consider the $2D$ grid on the right in Figure 4.5. The comb-shaped spanning tree is the one in thick line. Here we see that the maximum length of the fundamental cycles grows as $\sqrt{N}$. Since it can be proved that the maximum length of fundamental cycles in any spanning tree on a $2D$ grid grows *at least* as $\sqrt{N}$, the chosen spanning tree already achieves the maximum resilience to measurement noise. On the contrary, the minimum cycles always have length equal to 4. Thus, in this case the *Minimal cycles-algorithm* has always better performance than the *Tree-algorithm*. Notice that the choice of the spanning tree is fundamental to draw a comparison between the algorithms.

4.7 Numerical results

In this Section we provide numerical comparison between the two approaches we propose in this Chapter, and the frame localization algorithm proposed in Piovan et al. [2011b].

In the first experiment we simulate the *Tree-algorithm* and the *Minimal cycles-algorithm* on square grids of size $N = n^2$, for n ranging from 3 up to 20. An example of square grid is depicted in Figure 4.6 (left panel), where $n = 4$. In the same Figure, we also depict the type of spanning tree we use to build the $\mathscr{T}$-fundamental cycles. The construction is analogous for different values of n.

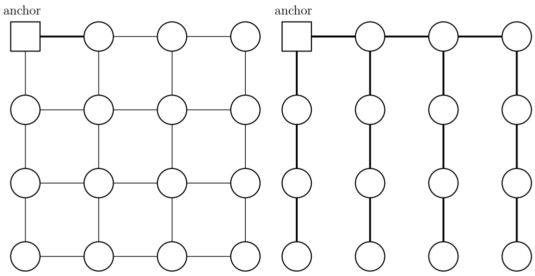


Figure 4.6: On the left a square grid graph for $n = 4$. On the right the correspondent spanning tree used in simulations.

In all simulations we set $\bar{\theta}_1 = 0$, while, for $v \in \{2,\dots,N\}$, $\bar{\theta}_v$ is randomly sampled from a uniform distribution on $[-\pi,\pi)$. The values of the noises ε_e, $e \in \mathcal{E}$, are randomly sampled from a uniform distribution on $[-\bar{\varepsilon},\bar{\varepsilon}]$, where $\bar{\varepsilon} = \frac{\pi}{8}$. For each n, the values we plot are averaged over 200 trials. Different $\bar{\boldsymbol{\theta}}$ and a different set of noises are generated for each trial.

On the left panel of Figure 4.7, we plot the estimate error

$$W(\hat{\theta}) = \frac{1}{N}\|(\bar{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}})_{2\pi}\|^2$$

for both the *Tree-algorithm* and the *Minimal cycles-algorithm*. On the right panel, we plot the error on $\bar{\boldsymbol{K}}$ defined as

$$e_K = \frac{1}{M}\|\bar{\boldsymbol{K}} - \hat{\boldsymbol{K}}\|^2.$$

The choice for $\bar{\varepsilon}$ was dictated by the inequality

$$\frac{\pi}{L_0} = \frac{\pi}{4} > \frac{\pi}{8} = \bar{\varepsilon},$$

which implies that the *Minimal cycles-algorithm* always correctly estimates $\bar{\boldsymbol{K}}$. On the contrary, observe that, in the case of the *Tree-algorithm*, e_K is zero only for low values of n. To conclude, notice that the behavior of $W(\hat{\theta})$ in the case of minimal cycles is approximately logarithmic in the number of nodes N, as predicted in Section 4.4.3. As expected, the *Minimal cycles-algorithm* outperforms the *Tree-algorithm*.

The second experiment concerns the comparison with the procedure proposed in Piovan et al. [2011b], which we call FL for Frame Localization. The goal of this algorithm, as ours, is to obtain an estimate of $\bar{\boldsymbol{\theta}}$ from the measurements $\boldsymbol{\eta}$. The basic idea of the FL algorithm is that

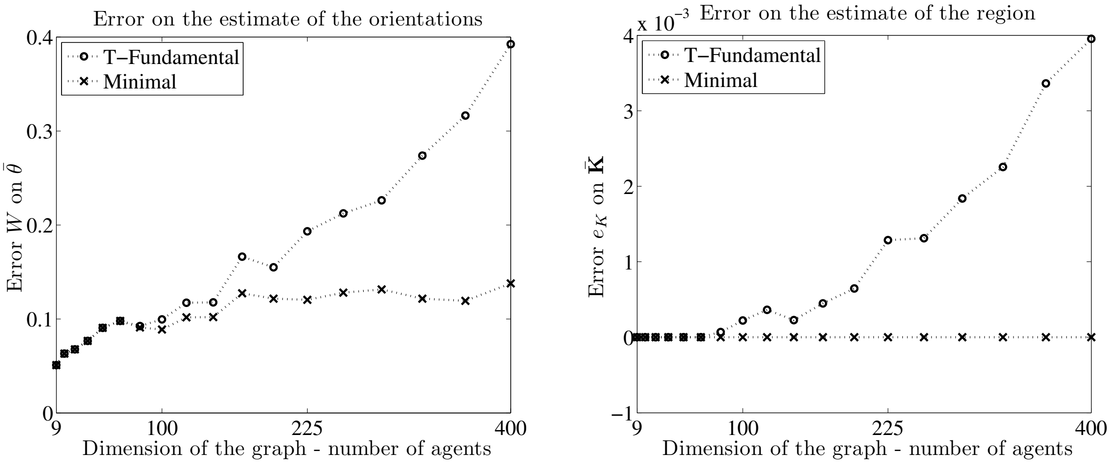


Figure 4.7: Average error on the orientations (modulo 2π) and error on $\bar{\boldsymbol{K}}$ in case of $2-D$ grid with $N = 9,\dots,400$. The circle–marked plot corresponds to the *Minimal cycles-algorithm*, the crossed-marked one to the *Tree-algorithm*.

measurements along cycles of the graph should sum up to multiples of 2π, as it happens in a noise–free scenario. Thus, in the FL algorithm first of all a $\mathbb{Z}$–basis of Γ is selected. Then, it aims to solve the following minimization problem

$$\hat{\boldsymbol{\psi}} = \underset{\boldsymbol{\psi}\in\mathbb{R}^{\mathscr{E}}\ \text{s.t.}\,(R\boldsymbol{\psi})_{2\pi}=0}{\arg\min} \|(\boldsymbol{\psi}-\boldsymbol{\eta})_{2\pi}\|_2, \tag{4.21}$$

where the rows of $R \in \mathbb{Z}^{(\mathscr{E}\setminus\mathscr{E}_{\mathscr{T}})\times\mathscr{E}}$ are the $\mathbb{Z}$–basis of Γ. In order to minimize the expression in (4.21), the authors propose the following iterative procedure

$$\begin{cases} \boldsymbol{\psi}(0) = \boldsymbol{\eta} \\ \boldsymbol{\psi}(t+1) = \boldsymbol{\psi}(t) - \kappa R^T(R\boldsymbol{\psi}(t))_{2\pi} \end{cases}$$

where κ is a strictly positive real number. If κ is small enough, then the algorithm converges to a vector $\boldsymbol{\psi}_\infty$ such that $(R\boldsymbol{\psi}_\infty)_{2\pi} = 0$ (Theorem 12, Piovan et al. [2011b]). Once $\boldsymbol{\psi}_\infty$ is determined, the estimate $\hat{\boldsymbol{\theta}}$ can be found spreading the information from the anchor, similarly to what is done in (4.20).

Remark 4.10. *It is worth remarking that, for both the* Tree-algorithm *and the* Minimal cycles-algorithm, *it is possible to provide a closed form expression for the estimate of* $\bar{\boldsymbol{\theta}}$, *see (4.19). Further research might lead to a better understanding of the influence of the measurement noise on the algorithm performance. Instead, as far as the FL algorithm is concerned, no closed* *formula is known for $\boldsymbol{\psi}_\infty$, which seems to depend on the choice of the parameter κ. Moreover, there is no guarantee that $\boldsymbol{\psi}_\infty$ is the solution to the minimization problem in (4.21).*

In the comparisons, we use the same set of measurements for all the estimation algorithms, and the FL algorithm is run taking the families of $\mathscr{T}$-fundamental cycles and the minimal cycles. The results of the first simulation are shown in Figure 4.8, left panel. In this case, the threshold for the error is again set at $\bar{\varepsilon} = \frac{\pi}{8}$. The plots are the averaged value of $W(\hat{\boldsymbol{\theta}})$ over 200 simulations. The Figure depicts the plots for the *Tree-algorithm*, the *Minimal cycles-algorithm*, and the FL algorithm run using the $\mathscr{T}$-fundamental cycles. The plot for FL algorithm run using the minimal cycles is not shown, since it overlaps the plot of the *Minimal cycles-algorithm*.

It can be seen from the Figure that, for low values of n, $\bar{\boldsymbol{K}}$ is correctly estimated by the *Tree-algorithm* and all the algorithms seem to provide similar performance. For higher values of n, the FL algorithm seems to perform better on the $\mathscr{T}$-fundamental cycles. This suggests that also for the FL algorithm the length of the cycles influences the performance.

On right panel in Figure 4.8, we propose a second comparison in case the measurements are corrupted by a large noise, setting $\bar{\varepsilon} = \frac{\pi}{3}$. Notice thus that, in this case, there is no guarantee that $\bar{\boldsymbol{K}}$ is correctly estimated, even if we exploit the minimal cycles. The plotted values are obtained averaging over 100 simulations on grids of dimension $n = 3, 4, \ldots, 10$. In this case, we show the plots of the results using the FL algorithm on both $\mathscr{T}$-fundamental cycles and minimal cycles. As in the previous simulations, the *Tree-algorithm* is always outperformed by the FL algorithm, but it is also true that the *Minimal cycles-algorithm* gives the best performance. From our numerical experiments, the FL algorithm has somehow an intermediate performance, and they show anyway that the shorter the cycles are, the better the estimates are.

4.8 Conclusions

This Chapter proposes two versions of an algorithm which allows a network of cameras to autonomously calibrate, more precisely to determine an estimate of their orientations w.r.t. a common reference frame. This algorithm is suitable to be distributed rather easily over the cameras which can act as agents in a multi-agent system. The proposed algorithm is based on a two steps procedure. In the first step, the algorithm estimates a vector of integers $\bar{\boldsymbol{K}}$ coding

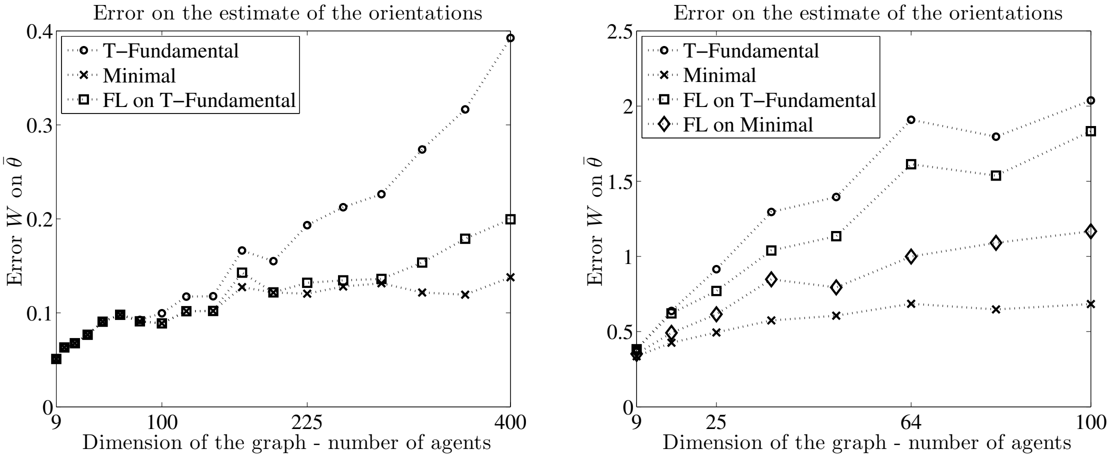


Figure 4.8: Comparison with the Frame Localization algorithm. On left panel, average error on the orientations (modulo 2π) in case of $2-D$ grid with $N = 9,\dots,400$ and small measurement noise. On the right panel, in the case of $N = 9,\dots,100$ with large measurement noise. The plot compares the *Minimal cycles-algorithm* (cross–marks), the *Tree-algorithm* (circle-marks), the FL algorithm using $\mathscr{T}$-fundamental cycles (square-marks), and the FL algorithm using minimal cycles (diamond-marks). On the left panel, the results of FL algorithm on minimal cycles are not shown since they overlap those of *Minimal cycles-algorithm.*

the region of convexity which the global minimum is believed to belong to. In the second step, the algorithm performs the computation of the minimizer belonging to the selected region of convexity. Compared with other existing algorithms, this procedure permits to understand the properties of the proposed solution and to understand when this solution is correct.

More investigations are needed for obtaining in this field of research, in the following directions.

- We believe that the proposed algorithm could be improved by obtaining a better estimate of the vector of integers $\bar{\boldsymbol{K}}$ based on the maximum a posteriori MAP estimation or on the maximum likelihood ML estimation.
- A more refined performance analysis has to be done, in terms of the error to gain the estimate $\bar{\boldsymbol{K}}$, and in terms of the index $W(\theta)$ proposed in (4.3).
- Finally, the more general case of cameras deployed in $\mathbb{R}^3$, and thus rotational calibration in $SO(3)$, needs to be addressed.

5 Asynchronous distributed calibration algorithm

This chapter focuses on the problem of calibrating planar networks of cameras in a distributed and asynchronous fashion. In many practical applications, it is not possible to have synchronous communications, and unpredictability is an intrinsic and unavoidable characteristic of the environment and the network itself. The camera network is modeled by a graph, and along each edge a noisy relative angular measurement is available. The goal is to achieve the absolute orientation of each camera with respect to a fixed external reference frame, in order to be able to perform monitoring and patrolling tasks. The idea is to exploit the cycles in the graph, along which all relative measurements sum to zero, in order to eliminate the noise. We design a distributed algorithm for the cameras to autonomously calibrate and we adopt an asynchronous gossip-like communication protocol. The proposed algorithm is proved to converge, almost surely and in the mean square sense, to the set of angles with zero cycle error. Finally, numerical experiments are presented to compare the performance of the algorithm on different graph topologies.

5.1 Introduction

The main characteristic of the calibration algorithm we propose is its randomness and asynchronous mode of operation. Distributed estimation algorithms with asynchronous randomized mode of operation have been considered by the control community in the last years, see e.g. Tsitsiklis [1984], Fagnani and Zampieri [2008]. In many practical applications collisions among messages in a sensor network may happen and delete data, and for example in the case of processor nets it is not possible to transmit data to more than one node at the same time instant. The same may happen when the agents in the networks are inherently unpredictable, like human beings opinions or animals in geometric formation, also known as flocking phenomena. Intrinsic limitations in such large multiagent networks must be taken into account even if the networks we consider are quite dense. The use of randomized algorithms turns to be appealing, since they allow to achieve better performance than deterministic ones with comparable complexity. In networked system control, randomness may be due to the choice of a randomized network communication protocol. Random linear schemes have been studied for instance in Boyd et al. [2005], Kempe et al. [2003], Fagnani and Zampieri [2008], Boyd et al. [2006], known as gossip algorithms, in which the evolution matrix of the algorithm changes randomly at every time step. Convergence has to be considered in a probabilistic sense and performance is studied in mean square sense.

As in Chapter 4, we deal with a camera network, in which each agent is equipped with a video device and the applications can include motion capture, monitoring, tracking and surveillance issues. The importance of the calibration problem in a coordinate autonomous camera network is clear, from the previous Chapter. As before, the cameras are supposed to lie in the same plane, and the input data that cameras have are the noisy relative orientations of neighboring cameras.

Note that, the distributed calibration algorithm proposed in Chapter 4 is deterministic and synchronous by construction. As a first attempt, one could try to slightly modify the latter to obtain a randomized and asynchronous mode of operation in the evolution of each node. Nevertheless, disregarding the choice of cycles in the graph on which we base the first estimation step of the integer vector $\boldsymbol{K}$, the procedure is strongly iterative, and follows a constraining logic to exploit our information on the network topology. In other words, at each time step, each agent i needs to receive data from the other nodes belonging to the neighboring cycles to which i belongs. Agent i cannot update its estimate of $\boldsymbol{K}$, without the whole data packet previously described. Thus an asynchronous random procedure may slow the convergence time, and make the algorithm not applicable in real life situations. Once the vector of integers $\boldsymbol{K}$ (see Section 4.2) is obtained, the second estimation step can be easily randomized, since it is a gradient flow of a convex function, Bertsekas and Tsitsiklis [1997].

Therefore, a different approach is required, and our focus is to accomplish the calibration task with a completely distributed and asynchronous random algorithm. We assume the communication protocol to be random gossip-like (see Boyd [2006], Fagnani and Zampieri [2008]), in which at each iteration only one link is updated gathering only the states of neighboring cameras. Convergence is considered in a probabilistic sense and performance is studied in terms of mean square convergence.

The algorithm is proved to converge in the mean square sense for general planar graphs, and the cycle-error vector has null variance. If we focus on ring graphs, the proposed algorithm converges almost surely, and in addition the expected value of the limit random variable equals the optimal solution, written in closed form. We also characterize the convergence speed proving that the estimate approaches the limit value exponentially fast. Numerical experiments are run to show further properties of the algorithm on planar graphs, and to investigate its performance on non-planar graphs.

In literature, the so-called localization problem (restricted to the position vector) has been deeply investigated in Barooah and Hespanha [2005], Barooah et al. [2006], Barooah and Hespanha [2007], as already pointed out in the Introduction of this thesis, the optimality of the solution is shown and the scalability with respect to the number of nodes is characterized. In Sarlette et al. [2006], Sarlette and Sepulchre [2009a], Sarlette [2009a] a consensus algorithm over $SO(2)$ is presented, based on the gradient flow of a cost function defined by means of the geodesic distance, while in Tron and Vidal [2009a] a similar cost function is considered, involving the chordal distance, and they face the more general problem of calibration in $SE(3)$. In both cases the cost function shows different local minima. An approach based on a non-convex optimization problem is considered in Borra et al. [2012a, a], explained in Chapter 4 of this thesis. A similar approach has also been exploited in Piovan et al. [2011a] where estimation is carried on by projecting the relative measurements into the sub-manifold of vectors whose sum along the cycle of the graph is a multiple of 2π.

The remainder of this Chapter is organized as follows. We first refer to Section 2.1 in order to recall basic definitions concerning algebraic graph theory. The outline is the following. A description of the setup, already provided in Section 4.2, and the minimization problem we want to deal with is developed in Section 5.2, and we also recall the deterministic algorithm in Piovan et al. [2011a], adding some performance analysis. In Section 5.3, the proposed algorithm is presented and Section 5.4 contains the convergence properties for general planar graphs (Section 5.4.1) and ring graphs (Section 5.4.2). Numerical experiments validating our analysis are provided and investigating the case of non-planar graphs in Section 5.5. Finally the conclusions of the Chapter are drawn in Section 5.6.

5.2 Problem setup

We refer to Section 4.2 for the formal statement of the problem of calibrating a planar camera network. Each camera can measure its noisy relative orientation with respect to its neighbors, and formally we denote the relative noisy orientation along every edge $e \in \mathscr{E}$ as

$$\eta_e = (\bar{\theta}_{s(e)} - \bar{\theta}_{t(e)} + \epsilon_e)_{2\pi},$$

where $\epsilon_e \in \mathbb{R}$ is a bounded noise. More precisely, there exists $\bar{\epsilon} \in \mathbb{R}_{>0}$, such that $|\epsilon_e| \leq \bar{\epsilon}$, for each $e \in \mathscr{E}$ and for each realization. Define $\bar{\psi}_e := (B\boldsymbol{\eta})_e = \bar{\theta}_{s(e)} - \bar{\theta}_{t(e)}$, where B is the incidence matrix. Let $\boldsymbol{\eta} \in [-\pi,\pi)^{\mathscr{E}}, \bar{\boldsymbol{\theta}} \in [-\pi,\pi)^{\mathscr{V}}, \bar{\boldsymbol{\psi}} \in [-\pi,\pi)^{\mathscr{E}}$, and $\boldsymbol{\epsilon} \in [-\bar{\epsilon},\bar{\epsilon}]^{\mathscr{E}}$ denote the vectors of all $\eta_e, \bar{\theta}_i, \bar{\psi}_e$ and ϵ_e respectively. The goal of the Chapter is to find a distributed algorithm that, given the relative measurements $\boldsymbol{\eta}$, estimates the absolute poses of the cameras, up to integer multiples of 2π. If we apply to the vector $\bar{\boldsymbol{\theta}}$ a global translation of integer multiples of 2π, the corresponding relative orientations do not vary. Therefore, without affecting generality, we fix an anchor node, called *root* (say node 1), that knows exactly its orientation with respect to the external reference frame.

In the noiseless case, an efficient way to reconstruct (exactly) $\bar{\theta}$ from η can be achieved considering any spanning tree $\mathscr{T}$ of the graph. For every node $i \in \mathscr{V}$, $\bar{\theta}_i$ can be obtained starting from the root value $\bar{\theta}_1$ and adding the relative orientations along edges in the shortest path connecting the root to node i.

In principle this can also be done in the noisy case; nevertheless, in this way we do not take advantage of all the redundant measurements corresponding to edges in $\mathscr{E} \backslash \mathscr{E}_{\mathscr{T}}$. Notice that in the noisy case, different spanning trees will lead to different estimations. The mathematical reason being that in general the sum of the relative orientation along closed paths will not be equal to 0.

In Piovan et al. [2011a], the following estimation approach is proposed: first η is projected onto the manifold of vectors having the property that sums along closed paths are 0 and then the estimation is obtained through a spanning tree as explained above. We now explain in detail the idea as our proposed algorithm is going to be a randomized asynchronous version of this one.

Step 1 corresponds to solve the following non-convex and non-linear problem.

Problem 5.1. ***(Planar orientation localization problem)*** *The noisy relative orientations are given, i.e.* $\boldsymbol{\eta} = (\bar{\boldsymbol{\psi}} + \boldsymbol{\epsilon})_{2\pi}$*, where* $\bar{\boldsymbol{\psi}} = B\bar{\boldsymbol{\theta}}$ *are the correct relative orientations and* $\boldsymbol{\epsilon}$ *the noise vector. Find an estimate of the relative orientation* $\hat{\boldsymbol{\psi}} \in [-\pi, \pi)^{\mathscr{E}}$ *such that*

$$\begin{cases} \|(\hat{\boldsymbol{\psi}} - \boldsymbol{\eta})_{2\pi}\|_2 = \min_{\boldsymbol{\psi}} \left\|(\boldsymbol{\psi} - \boldsymbol{\eta})_{2\pi}\right\|_2, \\ r_c \hat{\boldsymbol{\psi}} = 0 \quad \mod 2\pi, \text{for every cycle } c. \end{cases} \tag{5.1}$$

In Piovan et al. [2011a], the authors solves Problem 5.1 in distributed fashion, with a synchronous algorithm, here recalled. In the next we use the following definitions. First, define the *cycle-error vector* at time $t \in \mathbb{N}$ as $\hat{\boldsymbol{\epsilon}}(t) := R\hat{\boldsymbol{\psi}}(t) \in \mathbb{R}^{M-N+1}$, therefore $\hat{\boldsymbol{\epsilon}}(0) := R\boldsymbol{\eta}$ is the initial error on cycles. Second, the *projected cycle-error vector* at time $t \in \mathbb{N}$ is $\boldsymbol{\epsilon}(t) := (R\hat{\boldsymbol{\psi}}(t))_{2\pi} \in [-\pi, \pi)^{M-N+1}$. For each iteration $t \in \mathbb{N}$, the following update is performed

$$\begin{cases} \hat{\boldsymbol{\psi}}(0) = \boldsymbol{\eta}, \\ \hat{\boldsymbol{\psi}}(t+1) = \hat{\boldsymbol{\psi}}(t) - kR^T(R\hat{\boldsymbol{\psi}}(t))_{2\pi}. \end{cases} \tag{5.2}$$

Step 2 consists in finding the estimate $\hat{\boldsymbol{\theta}}$ of the absolute poses $\bar{\boldsymbol{\theta}}$ starting from the anchor value $\bar{\theta}_1$ and adding the corresponding estimated relative orientations $\hat{\psi}$ along the the minimum path in the spanning tree $\mathscr{T}$ connecting each node to the anchor. This final step is straightforward and can be performed in a distributed way, therefore our focus is on Step 1.

Concerning Step 1, the authors in Piovan et al. [2011a] provide convergence results, stating that $\hat{\boldsymbol{\psi}}$ solves Problem 5.1. We now provide a corollary of that result (see [Piovan et al., 2011a, Theorem 12]) concerning the convergence rate and the error of the estimate provided by their algorithm, depending on the topology of the underlying graph.

Theorem 5.2. ***(Error characterization for the synchronous algorithm)*** *Given* $\mathscr{G} = (\mathscr{V}, \mathscr{E})$*, and the noisy relative measurements* $\boldsymbol{\eta} \in [-\pi, \pi)^{\mathscr{E}}$*, let* $\hat{\boldsymbol{\psi}} \in [-\pi, \pi)^{\mathscr{E}}$ *be the estimate provided by Algorithm in Piovan et al. [2011a] in Eq.* (5.2) *with stepsize* $0 < k < 2/(1 + \lambda_{max}(C))$*, then*

$$\|\hat{\boldsymbol{\psi}} - \boldsymbol{\eta}\|_2 \leq \frac{\sqrt{\lambda_{max}(C)}}{2-k}\|(R\boldsymbol{\eta})_{2\pi}\|_2 \tag{5.3}$$

where $\lambda_{max}(C)$ *denotes the largest eigenvalue of the essential cycle matrix* C*. Moreover* *(i) Ring graph:* $\lambda_{max}(C) = N$*;*

(ii) Grid graph: $\lambda_{max}(C) \leq 8$*;*

(iii) Complete graph: $\lambda_{max}(C) \leq 6$*;*

(iv) If $\mathscr{G}$ *is such that each edge belongs to* s *minimal cycles, and the minimal cycle length is* l*:* $\lambda_{max}(C) \leq l \cdot s$*. Notice that if* $\mathscr{G}$ *is planar* $s = 2$*.*

Proof. In order to prove Eq. (5.3), note that for every edge e

$$\|\hat{\psi}_e(t) - \eta_e\|_2 = \|\sum_{s=0}^{t-1} \hat{\psi}_e(s+1) - \hat{\psi}_e(s)\|_2 \leq \sum_{s=0}^{t-1} k\|R^T \hat{\boldsymbol{\psi}}(s)\|_2.$$

Hence, passing to the limit as time goes to infinity, and defining $\hat{\boldsymbol{\psi}} := \hat{\boldsymbol{\psi}}(\infty) = \lim_{t\to\infty} \hat{\boldsymbol{\psi}}(t)$, it holds

$$\|\hat{\boldsymbol{\psi}} - \boldsymbol{\eta}\|_2 \leq \sum_{s=0}^{\infty} k\|R^T \epsilon(s)\|_2 \leq \frac{k}{1-\rho}\|R^T\|_2\|\boldsymbol{\epsilon}(0)\|_2 = \frac{\|R^T\|_2\|\boldsymbol{\epsilon}(0)\|_2}{2-k} \leq \frac{\lambda_{max}(C)^{1/2}}{2-k}\|(R\boldsymbol{\eta})_{2\pi}\|_2.$$

We used the fact that $\|\boldsymbol{\epsilon}(t)\|_2^2 \leq \rho^t \|\boldsymbol{\epsilon}(0)\|_2^2, \rho = (1-k)^2$ (see [Piovan et al., 2011a, Theorem 12]) and $\left\|R^T\right\|_2 = \|C\|_2^{1/2} = \sqrt{\lambda_{max}(C)}$. Then, to achieves the subsequent estimates of the largest eigenvalue of C, it suffices to apply Gershgorin Theorem and to observe that $C_{c_i,c_j} = \boldsymbol{r}_{c_i}\boldsymbol{r}_{c_j}^T$. □

5.3 Description of the proposed algorithm

The algorithm we proposed is a randomized asynchronous version of the one proposed in Piovan et al. [2011a] and it is formally described in Algorithm 3.

In the last step the updated value information is locally spread along the graph as follows: if $e(t)$ is the selected edge at time t, for every cycle c such that $r_c(e(t)) \neq 0$, send the updated state $\hat{\psi}_e(t)$ to all $f \in \mathscr{E}$ with $r_c(f) \neq 0$.

The update can be rewritten as

$$\hat{\psi}_e(t+1) - \hat{\psi}_e(t) = \begin{cases} -k\sum_{c\in\mathscr{F}} r_c(e)\epsilon_c(t), & \text{if } e(t) = e, \\ 0, & \text{otherwise,} \end{cases} \tag{5.4}$$

If we define $I_e = (i_{kl}) \in \{0,1\}^{M\times M}$ as the null matrix except for $i_{e,e} = 1$ Eq. (5.4) in vector form

Algorithm 3 Asynchronous-Gossip-Algorithm

```
(Input variables)
$\eta_1, \dots, \eta_M$ noisy relative orientations;
$\boldsymbol{r}_{c_1}, \dots, \boldsymbol{r}_{c_{M-N+1}}$ minimal cycles vectors;
$\tau$ time horizon;
$k$ stepsize;

(Step A: initialization)
$\hat{\boldsymbol{\psi}}(0) = \boldsymbol{\eta}$;

(Step B: estimate $(B\bar{\boldsymbol{\theta}})_{2\pi}$)
for $t = 1, \dots, \tau$ do
    Choose randomly $e \in \mathscr{E}$;
    $\hat{\psi}_e^+ = \hat{\psi}_e - k\sum_{i=1}^{M-N+1} r_{c_i}(e)(\boldsymbol{r}_{c_i} \cdot \hat{\boldsymbol{\psi}})_{2\pi}$;
    Send $\psi_e^+$ to all $f \in \mathscr{E}$ such that $f \in c_i$ and $r_{c_i}(e) \neq 0$.
```

takes the form

$$\hat{\boldsymbol{\psi}}(t+1) = \hat{\boldsymbol{\psi}}(t) - kI_{e(t)}R^T\boldsymbol{\epsilon}(t). \tag{5.5}$$

In the following we take the initial error ϵ as fixed (not random). Randomness is thus completely coded in the sequence of chosen edges $e(t)$ which we assume to be independent and uniformly distributed in the set of all edges. Note that $\boldsymbol{\epsilon}(t)$ and $\hat{\boldsymbol{\psi}}(t)$ are random variables. $\mathbb{E}$ will denote expectation with respect to the choice of the sequence $e(t)$. Also, for a generic random vector, we will use the notation $\|f\|_{L^2}$ to denote the mean square norm: $\|f\|_{L^2} := (\mathbb{E}\|f\|^2)^{1/2}$, where $\|\cdot\|$ is the usual Euclidean norm of a vector.

5.4 Perfomance analysis

In this section we analyze the behavior of the proposed gossip algorithm for the special class of planar graphs.

5.4.1 General graphs

From Eq. (5.5), we can determine the dynamics of the cycle-error vector $\epsilon(t)$:

$$\boldsymbol{\epsilon}(t+1) = (U_{e(t)}\boldsymbol{\epsilon}(t))_{2\pi} \tag{5.6}$$

where, we recall, $e(t)$ is the randomly selected edge at time t, and U_e is a square $(M-N+1)$-dimensional matrix defined as

$$U_e := I - kRI_eR^T. \tag{5.7}$$

Note that U_e is symmetric, positive semidefinite, for every $e \in \mathscr{E}$. Moreover,

$$(U_e)_{cc'} = \begin{cases} 1, & \text{if } c = c', e \not\in \operatorname{supp}(\boldsymbol{c}) \\ 1-k, & \text{if } c = c', e \in \operatorname{supp}(\boldsymbol{c}) \\ k, & \text{if } c \neq c', e \in \operatorname{supp}(\boldsymbol{c}) \cap \operatorname{supp}(\boldsymbol{c}') \\ 0, & \text{otherwise.} \end{cases}$$

It turns out that U_e is stochastic for any non-border edge e, while is sub-stochastic if e is a border edge. Since U_e is a stochastic or sub stochastic matrix, it follows that if $\boldsymbol{x} \in [-\pi,\pi)^{N-M+1}$, then also $U_e\boldsymbol{x} \in [-\pi,\pi)^{N-M+1}$. This has an importance consequence: since $\boldsymbol{\epsilon}(0) \in [-\pi,\pi)^{M-N+1}$, it follows that the dynamics of the $\boldsymbol{\epsilon}(t)$ actually satisfies the simpler dynamics

$$\boldsymbol{\epsilon}(t+1) = U_{e(t)}\boldsymbol{\epsilon}(t). \tag{5.8}$$

We can now state the following convergence result.

Theorem 5.3. ***(Convergence and estimate characterization for general graphs)*** *Given a connected planar graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, the noisy relative measurements $\boldsymbol{\eta} \in [-\pi,\pi)^{\mathscr{E}}$ and the stepsize $0 < k < 1$, then there exists a r.v. $\hat{\boldsymbol{\psi}}(\infty)$ taking values in $\mathbb{R}^{\mathscr{E}}$, a number $\rho \in [0,1)$ such that*

(i) $\hat{\boldsymbol{\psi}}(t)$ converges to $\hat{\boldsymbol{\psi}}(\infty)$ almost surely and exponentially fast in mean square sense, that is

$$\|\hat{\boldsymbol{\psi}}(t) - \hat{\boldsymbol{\psi}}(\infty)\|_{L^2} \leq \frac{k\|RR^T\|^{1/2}}{M(1-\rho)}\rho^t\|\boldsymbol{\epsilon}(0)\|_{L^2}; \tag{5.9}$$

(ii)

$$\|\hat{\boldsymbol{\psi}}(\infty)-\boldsymbol{\eta}\|_{L^2} \leq \frac{k\|RR^T\|^{1/2}}{M(1-\rho)}\|\boldsymbol{\epsilon}(0)\|_{L^2}; \tag{5.10}$$

(iii) the limit cycle-error vector is null, that is

$$\boldsymbol{\epsilon}(\infty) := (R\hat{\boldsymbol{\psi}}(\infty))_{2\pi} = 0 \text{ a.s..}$$

Proof. (i) It follows from (5.8) that, for every $t \geq 0$, $\boldsymbol{\epsilon}(t) = U_{e(t)} \dots U_{e(1)}\boldsymbol{\epsilon}(0)$. Fix now a border edge e^* and notice that we can always build a sequence of edges $e_{j_1}, \dots, e_{j_s}$ such that $e_{j_1} = e^*$, e_{j_l} and $e_{j_{l+1}}$ always belong to the same minimal cycle, and for every minimal cycle $c \in \mathscr{F}$ there exists e_{j_l} with $r_c(e_{j_l}) \neq 0$. This simply follows from the fact that the dual graph of $\mathscr{G}$ is connected and by considering a path in this graph starting from the minimal cycle which e^* belongs to and touching all other minimal cycles. Consider now $Q = U_{e_{j_s}} \cdots U_{e_{j_1}}$. It is sub-stochastic since $U_{e_{j_1}}$ is so and the others are stochastic or sub-stochastic. Moreover, a simple induction argument shows that no row of Q sums to 1. Put $\alpha := \|Q\|_\infty = \max_c (Q\mathbf{1})_c \in (0,1)$. Considering that the probability that the sequence of edges $e_{j_1}, \dots, e_{j_s}$ is chosen from left to right has a positive probability p (equal to $p = M^{-s}$), it follows, by Chernoff bound, that for any $r = 1, \dots t^*$, it holds

$$\mathbb{P}(\|\boldsymbol{\epsilon}(ns+r)\|_\infty \leq \alpha^{np/2}\|\boldsymbol{\epsilon}(r)\|_\infty \geq 1 - e^{-n\beta}$$

where $\beta > 0$. An application of Borel-Cantelli Lemma now yields that $\boldsymbol{\epsilon}(t)$ is summable almost surely. From (5.5) it immediately follows that $\hat{\psi}(t)$ is a Cauchy sequence almost surely, so that it converges, almost surely, to a measurable r.v. $\hat{\psi}(\infty)$.

We now investigate convergence in mean square sense. Notice that

$$\mathbb{E}[\|\boldsymbol{\epsilon}(t+1)\|^2] = \mathbb{E}[\boldsymbol{\epsilon}^T(t)U^2_{e(t)}\boldsymbol{\epsilon}(t)] = \mathbb{E}[\mathbb{E}[\boldsymbol{\epsilon}^T(t)U^2_{e(t)}\boldsymbol{\epsilon}(t) \mid e(0), \dots, e(t-1)]] = \mathbb{E}[\boldsymbol{\epsilon}^T(t)\mathbb{E}(U^2_{e(t)})\boldsymbol{\epsilon}(t)],$$

and $\mathbb{E}(U^2_{e(t)})$ is a symmetric matrix. If we denote by ρ^2 its spectral radius (largest in modulo eigenvalue), we can estimate $\mathbb{E}[\|\boldsymbol{\epsilon}(t+1)\|^2] \leq \rho^2\mathbb{E}\|\boldsymbol{\epsilon}(t)\|^2$. This yields, for all $t \geq 0$,

$$\|\boldsymbol{\epsilon}(t+1)\|_{L^2} \leq \rho^t\|\boldsymbol{\epsilon}(0)\|_{L^2}. \tag{5.11}$$

Let us now verify that $\rho \in [0,1)$. Indeed, since U_e^2 inherits the property of U_e, namely it is stochastic (resp. sub stochastic) if U_e is so, and since U_e is sub-stochastic with positive probability, it turns out that the average $\mathbb{E}(U_{e(t)}^2)$ is sub-stochastic. Moreover, it is easy to see that is also irreducible since the corresponding graph coincide with the dual of $\mathcal{G}$, defined in Section 2.1. This implies the thesis on ρ.

From (5.5) it now immediately follows that $\hat{\psi}(t)$ is a Cauchy sequence in L^2 sense, so that it must converge to $\hat{\psi}(\infty)$ in mean square sense too. Moreover we can estimate

$$\|\hat{\boldsymbol{\psi}}(\infty) - \hat{\boldsymbol{\psi}}(t)\|_{L^2} \leq k \sum_{s=t}^{+\infty} \|R^T I_{e(s)} \boldsymbol{\epsilon}(s)\|_{L^2} \leq \frac{k}{M} \sum_{s=t}^{+\infty} \|RR^T\|^{1/2} \rho^s \|\boldsymbol{\epsilon}(0)\|_{L^2}.$$

This immediately yields (5.9).

(ii) The thesis follows immediately from previous relation by taking $t = 0$.

(iii) Suppose by contradiction $\boldsymbol{\epsilon}(\infty) := (R\hat{\boldsymbol{\psi}}(\infty))_{2\pi} \neq 0$, then it would exist a cycle c, numbers $n \in \mathbb{N}$ and $k \in \mathbb{Z}$ such that

$$\mathbb{P}(R\hat{\boldsymbol{\psi}}(\infty)_c \in]n^{-1}, 2\pi - n^{-1}] + 2k\pi) > 0.$$

Since $R\hat{\boldsymbol{\psi}}(t) \to R\hat{\boldsymbol{\psi}}(\infty)$, it follows that

$$\mathbb{P}(\exists t_0 : \, R\hat{\boldsymbol{\psi}}(t)_c \in]n^{-1}, 2\pi - n^{-1}] + 2k\pi, \, \forall t \geq t_0) > 0$$

which yields

$$\mathbb{P}(\exists t_0 : \, |((R\hat{\boldsymbol{\psi}}(t))_{2\pi})_c| > n^{-1}, \, \forall t \geq t_0) > 0.$$

This contradicts the fact that $\boldsymbol{\epsilon}(t) = (R\hat{\boldsymbol{\psi}}(t))_{2\pi} \to 0$ almost surely. The proof is thus complete.

□

From the proof of Theorem 5.3, a very simple characterization for the rate of convergence ρ can be obtained. Indeed it coincides with the spectral radius of the sub stochastic matrix $\mathbb{E}[U_e^2]$. We now carry on an analysis of this matrix for planar graphs in Theorem 5.4.

Theorem 5.4. ***(Asymptotic misplacement towards initial estimate)*** *Given a connected planar graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the noisy relative measurements $\boldsymbol{\eta} \in [-\pi, \pi)^{\mathcal{E}}$ and the stepsize $k \in (0,1)$, the*

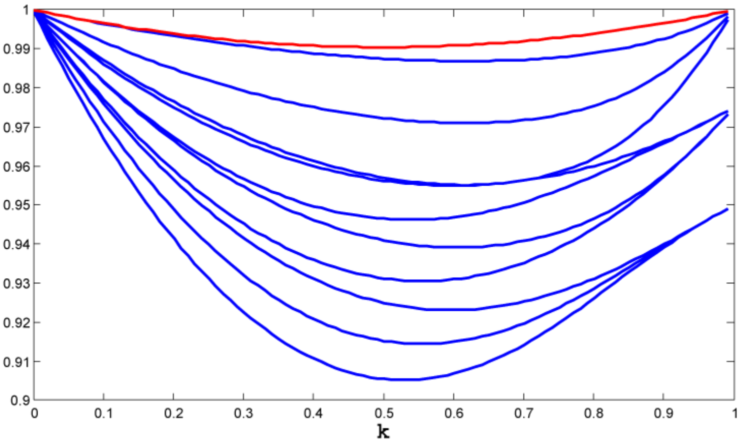


Figure 5.1: Consider a planar grid graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with $|\mathcal{V}| = N = 5^2$ and thus $|\mathcal{E}| = 2N - 2\sqrt{N} = 42$. This figure shows the $M - N + 1 = 18$ (they may have multiplicity greater than 1) eigenvalues (blue lines) of the substochastic matrix $\mathbb{E}[U_e^2]$ defined in Eq. 5.13. The eigenvalues are functions of the algorithm stepsize $k \in (0, 1)$. From this plot, we can compare the spectral radius ρ of the latter matrix, with the upper bound (red line) shown in Theorem 5.4.

spectral radius $\rho := \rho(\mathbb{E}[U_e^2])$ *satisfies the inequality*

$$\rho \le 1 - 2\frac{k(1-k)}{M}\lambda_{min}(C) \le 1 - 2\frac{k(1-k)}{M} < 1. \tag{5.12}$$

Before proving Theorem 5.4, let us present the following instrumental Lemma.

Lemma 5.5. ***(Matrices with equivalent spectral properties)*** *Given an invertible diagonal matrix $J \in \mathbb{R}^{M \times M}$, and the cycle matrix $R \in \mathbb{R}^{(M-N+1)\times M}$ associated with a planar graph $\mathcal{G}$ (cfr. Section 2.1.1), the following matrices have the same non-zero eigenvalues*

$$M_1 := RJR^T,$$
$$M_2 := R^T RJ.$$

Proof. Suppose $\lambda \in \mathbb{R} \setminus \{0\}$, $v \in \mathbb{R}^{M-N+1}$ are associated eigenvalue and eigenvector respectively, i.e. $M_1 v = \lambda v$. Then consider $w = R^T v$, it follows that $M_2 w = \lambda w$. Note that $v \in \ker(R^T) \subseteq \mathbb{R}^{\mathcal{E}}$ if and only if, for each edge e, the number of adjacent cycles oriented coherently and resp. incoherently with e is the same. This does not hold for border edges, thus $\ker(R^T) = \{0\}$. From the injectivity of the linear map R^T, that is $Im(R^T) = \mathbb{R}^{M-N+1}$, we can conclude that the non-zero eigenvalues of M_1 are the same of M_2. □

Proof of Theorem 5.4. We start computing explicitly U_e^2 for a planar graph, in the case (A1) e border edge, and (A2) e internal edge.

(A1) If e is a border edge, then

$$(U_e^2)_{c,c'} = \begin{cases} (1-k)^2, & \text{if } c = c', e \in \operatorname{supp}(c) \\ 1, & \text{if } c = c', e \notin \operatorname{supp}(c) \\ 0, & \text{if } c \neq c. \end{cases}$$

(A2) If e is not a border edge, then

$$(U_e^2)_{c,c'} = \begin{cases} 2(1-k)k, & \text{if } c \neq c', e \in \operatorname{supp}(c) \cap \operatorname{supp}(c') \\ (1-k)^2 + k^2, & \text{if } c = c', e \in \operatorname{supp}(c) \\ 1, & \text{if } c = c', e \notin \operatorname{supp}(c) \\ 0, & \text{otherwise.} \end{cases}$$

Note that U_e in case (A1) is substochastic, while in case (A2) is stochastic. From Eq. (5.7) we compute

$$U_e^2 = I - 2kRI_eR^T + k^2RI_eR^TRI_eR$$

and thus

$$\mathbb{E}[U_e^2] = I - \frac{2k}{M}C + \frac{k^2}{M}RD_{\mathscr{E}}R^T = I - \frac{k}{M}R(2I - kD_{\mathscr{E}})R^T \tag{5.13}$$

where $C = RR^T$ is the essential matrix, $D_{\mathscr{E}}$ is a diagonal matrix with D_{ee} is the number of cycles adjacent to edge e. Since $\mathscr{G}$ is supposed to be planar, $D_{ee} \in \{0,1,2\}$. Let us denote by I_c (resp. B_c), the number of internal (resp. border) edges of c. In the new notations $d_c = I_c, |c| = I_c + B_c$, and $D_{\mathscr{E}}r_c^T = 2I_c + B_c$ is the weighted sum of the edges in cycle c, thus

$$(\mathbb{E}[U_e^2]\mathbf{1})_c = 1 - \frac{2k}{M}B_c + \frac{k^2}{M}B_c$$
$$(\mathbb{E}[U_e^2])_{cc} = 1 - \frac{2k}{M}(I_c + B_c) + \frac{k^2}{M}(2I_c + B_c)$$

using the fact that internal edges e have weight 2 in $D_{\mathscr{E}}$. As already proved in Theorem 5.3, since there exists at least a border cycle c such that $(\mathbb{E}[U_e^2]\mathbf{1})_c = 1-(k+2)B_c < 1$, it can be easily verified that $\mathbb{E}[U_e^2]$ is substochastic. Hence

$$\rho = 1 - k/M\lambda_{min}(R(2I - kD_{\mathcal{E}})R^T),$$

and also the matrix $R(2I - kD_{\mathcal{E}})R^T$ is substochastic. From Lemma 5.5, the eigenvalues of $(2I - kD_{\mathcal{E}})R^T R$ are also the only non-zero eigenvalues of $R(2I - kD_{\mathcal{E}})R^T$, therefore we estimate ρ using the first latter matrix. Recall that $C = RR^T$ and $\lambda_{min}(R^T R) = \lambda_{min}(C)$, and $\lambda_{min}(AB) \leq \lambda_{min}(A)\lambda_{min}(B)$ for any two matrices A, B, then

$$\lambda_{min}((2I - kD_{\mathcal{E}})R^T R) \leq \lambda_{min}(2I - kD_{\mathcal{E}})\lambda_{min}(C) = 2(1-k)\lambda_{min}(C).$$

Thus we can derive the thesis, that is Eq. (5.12). □

From Theorem 5.2 and Theorem 5.3, (ii) provides bounds on the distance between the initial noisy orientations $\boldsymbol{\eta}$ and the final estimates obtained by means of the synchronous and asynchronous gossip-like algorithm respectively. They both depend on the stepsize k and the graph topology by means of the essential matrix C (see Section 2.1.1).

Remark 5.6. ***(Comparison of asymptotic misplacement towards initial estimate)*** *Let us define*

$$c := \sqrt{\lambda_{max}(C)}\|\epsilon(0)\|_{L^2} = \sqrt{\|C\|_2}\|\epsilon(0)\|_{L^2},$$

therefore we have to compare

$$\|\hat{\boldsymbol{\psi}}_1 - \boldsymbol{\eta}\| \leq \frac{c}{2-k}, \quad \|\hat{\boldsymbol{\psi}}_2 - \boldsymbol{\eta}\| \leq \frac{ck}{M(1-\rho)} \tag{5.14}$$

where $\rho = \rho(\mathbb{E}[U_e^2])$, $\hat{\boldsymbol{\psi}}_1$ is the estimate given by the synchronous deterministic Algorithm defined in Eq. (5.2)*, and $\hat{\boldsymbol{\psi}}_2$ is provided by Algorithm in 3. From Theorem 5.4, we can derive that*

$$\|\hat{\boldsymbol{\psi}}_1 - \boldsymbol{\eta}\| \leq \frac{c}{2-k}, \quad \|\hat{\boldsymbol{\psi}}_2 - \boldsymbol{\eta}\| \leq \frac{c}{(2-2k)\lambda_{min}(C)} \leq \frac{c}{2-2k}.$$

In conclusion, the deterministic synchronous algorithm and the asynchronous one, have comparable performances, in terms of final error of the relative orientation estimate.

Of course, a fundamental fact missing in performance analysis of Theorem 5.3, is an estima-

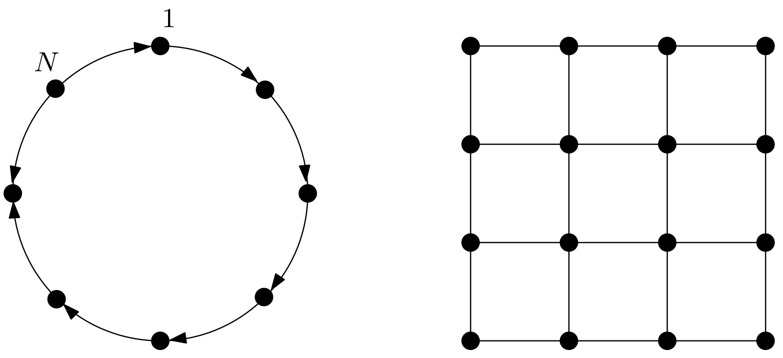


Figure 5.2: This figure shows a ring graph (left), and a grid graph (left) as an example of a planar graph. Algorithm 3 is analyzed on these different communication networks in Section 5.4.1 and 5.4.2 respectively.

tion of the displacement respect to the optimal solution $\|\hat{\boldsymbol{\psi}}(\infty)-\boldsymbol{\psi}^*\|_{L^2}$. In this work we will limit this analysis to the case of a ring graph.

5.4.2 Ring graphs

Consider a ring graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$ with $|\mathcal{V}|=N$. There is just one minimal cycle corresponding to $\boldsymbol{r}=\mathbf{1}$, see Fig. 5.2 (left). Note that Theorem 5.3 applies also in this case, but we can characterize additional properties of the estimate $\hat{\boldsymbol{\psi}}$.

Notice that, if e is the sampled edge, the only updated component is

$$\psi_e(t+1)=\psi_e(t)-k\left(\mathbf{1}\,\boldsymbol{\psi}(t)\right)_{2\pi},$$

that can be casted in vector form as

$$\begin{cases}\hat{\boldsymbol{\psi}}(t+1)=\hat{\boldsymbol{\psi}}(t)-k\boldsymbol{b}_{e(t)}\epsilon(t)\\ \hat{\boldsymbol{\psi}}(0)=\boldsymbol{\eta}\end{cases}\tag{5.15}$$

where $\boldsymbol{b}_{e(t)}\in\{0,1\}^N$ is defined as 1 in the component $e(t)$, and 0 otherwise. First, let us notice that the dynamics on the (scalar) cycle-error becomes deterministic, i.e. Eq. (5.8) becomes

$$\epsilon(t+1)=((1-k)\epsilon(t))_{2\pi}=(1-k)\epsilon(t).\tag{5.16}$$

This has an immediate consequence: $\boldsymbol{\psi}\hat{(}t)$, which is still a sequence of random variables, always converges to $\boldsymbol{\psi}(\hat{\infty})$ (and not just almost surely). Moreover, equation (5.15) immediately yields

$$\hat{\boldsymbol{\psi}}(\infty) = \boldsymbol{\eta} - k\left(\sum_{s=0}^{\infty}(1-k)^{s}\epsilon(0)\boldsymbol{b}_{e(s)}\right). \tag{5.17}$$

It is straightforward to compute an optimal solution of Problem 5.1 (see also Kaczmarz [1993]) for a ring graph, as follows

$$\boldsymbol{\psi}^{*} = \boldsymbol{\eta} - \frac{1}{N}\boldsymbol{r}^{T}(\boldsymbol{r}\boldsymbol{\eta})_{2\pi}. \tag{5.18}$$

If we run the Algorithm described in Eq. (5.2) with $k = 1/N$, we obtain $\boldsymbol{\psi}^{*}$ in one time step (see Piovan et al. [2011a]). From Eq. (5.17) and (5.18) we further obtain

$$\begin{aligned}\|\hat{\boldsymbol{\psi}}(\infty) - \boldsymbol{\psi}^{*}\|_{L^2} &= \|k\sum_{s=0}^{+\infty}(1-k)^{s}\epsilon(0)[b_{e(s)} - N^{-1}\mathbf{1}]\|_{L^2} \leq k\sum_{s=0}^{+\infty}(1-k)^{s}|\epsilon(0)|\|b_{e(s)} - N^{-1}\mathbf{1}\|_{L^2}\\ &\leq k\sum_{s=0}^{+\infty}(1-k)^{s}|\epsilon(0)|(1-N^{-2})^{1/2} \leq |\epsilon(0)|.\end{aligned}$$

The latter difference remains bounded, it does not depend on the number of nodes N, and it implies

$$N^{-1}\|\hat{\boldsymbol{\psi}}(\infty) - \boldsymbol{\psi}^{*}\|_{L^2} = O(N^{-1}). \tag{5.19}$$

This can be rephrased also saying that the difference between the estimate in the two algorithms per edge goes to 0 as N is asymptotically large.

5.5 Numerical examples

In this section we validate our distributed asynchronous Algorithm 3 through a numerical study.

5.5.1 Experiments on planar graphs

First, we consider a ring graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = N = 20$ as depicted in Fig. 5.3 (left). Recall that $\bar{\boldsymbol{\psi}}$ denotes a solution to Problem 5.1, and $\boldsymbol{\psi}^{*}$ is the optimal estimate given by Algorithm in Eq. (5.2). We run Algorithm 3 several times, that is, we sample 50 different sequences of

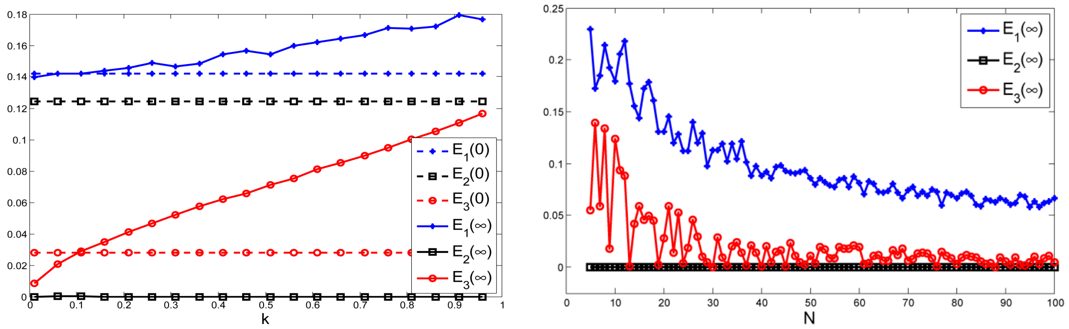


Figure 5.3: In the left figure we consider a ring graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $N = 20$, the time horizon is $\tau = 300$, and the number of samples is $n_s = 50$. Given Eq. (5.20), the figure shows respectively the initial values $E_1(0)$ (blue dash-star line), $E_2(0)$ (black dash-dot line), $E_3(0)$ (red dash-square line), and asymptotic values $E_1(\infty), E_2(\infty), E_3(\infty)$ (solid lines), while we vary the stepsize $k \in (0, 1)$. The projection error E_2 goes to zero for every k, while E_1, E_3 decrease only for k smaller than a certain threshold related to $\lambda_{\max}(C)$ (cfr. Theorem 5.2 for the synchronous algorithm threshold). In the right figure we fix the stepsize $k = 0.3$ and time horizon $\tau = 250$. The figure shows the asymptotic values $E_1(\infty)$ (blue dash-star line), $E_2(\infty)$ (black dash-dot line), $E_3(\infty)$ (red dash-dot line), and $V(\infty)$ (green dashed line) defined in Eq. (5.20), while we vary $N \in [5, 100]$ in Algorithm 3.

edges, and we average over these samples the following quantities

$$
\begin{aligned}
E_1(t) &:= \mathbb{E}[\|(\hat{\boldsymbol{\psi}}(t) - \bar{\boldsymbol{\psi}})_{2\pi}\|_2]/M, \\
E_2(t) &:= \mathbb{E}[\|\boldsymbol{\epsilon}(t)\|_2]/(M - N + 1), \qquad (5.20)\\
E_3(t) &:= \mathbb{E}[\|(\hat{\boldsymbol{\psi}}(t) - \boldsymbol{\psi}^*)_{2\pi}\|_2]/M,
\end{aligned}
$$

As described in Section 5.2, the noise on each each is sampled from a uniform distribution, namely $\epsilon_e \sim \text{Unif}([-\bar{\epsilon}, \bar{\epsilon}])$, for every $e \in \mathcal{E}$, with maximum noise $\bar{\epsilon} = \pi/3$.

We consider the normalized final error $E_1(\infty)$ of the estimate $\hat{\boldsymbol{\psi}}$ with respect to a true solution $\bar{\boldsymbol{\psi}}$ of Problem 5.1, and the normalized final distance $E_3(\infty)$ between our estimate and the optimal estimate $\boldsymbol{\psi}^*$ given by the deterministic algorithm proposed in Piovan et al. [2011a]. Both these quantities approach zero in the 2-norm as time goes to infinity, for any stepsize k under a certain threshold depending on N (see Fig. 5.3), while the asymptotic projected cycle-error $\epsilon(\infty)$ goes to zero for any k. Note that, as k increases, the precision of the estimate gets worse. On the other hand, for tiny stepsizes, the time required to guarantee a certain threshold precision becomes large. Note that the estimate provided by the synchronous deterministic algorithm proposed in Piovan et al. [2011a], with stepsize $k = 1/N$, has the corresponding

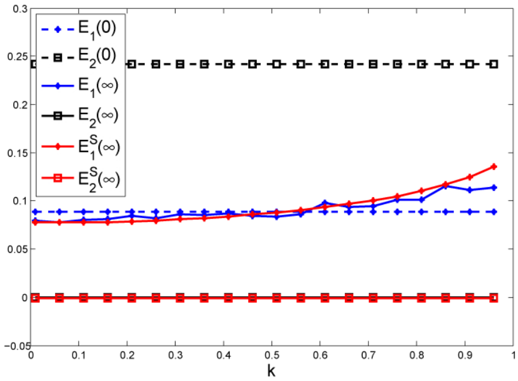


Figure 5.4: Given a grid $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $N = n^2$ nodes, we fix $n = 5$, the time horizon $\tau = 10^3$ and the number of samples $n_s = 50$. The figure shows the asymptotic values $E_1(\infty)$ (blue dash-star line) and $E_2(\infty)$ (black dash-dot line), defined in Eq. (5.20), while we vary $k \in (0,1)$ in Algorithm 3. If we run the synchronous deterministic algorithm in Piovan et al. [2011a], we obtain the corresponding asymptotic values $E^S_1(\infty)$ (red dash-star line) and $E^S_2(\infty)$ (red dash-dot line).

performance index as follows: $E^S_1(\infty) = 2.6216$ (it is not shown in Fig. 5.3 for the sake of space). Therefore the randomized algorithm has a better performance according to E_1. Moreover, considering the previous setup, we vary the number of nodes N (see Fig. 5.3 right), in order to validate Eq. (5.19).

Second, we fix a planar grid graph, and Fig. 5.4 shows the asymptotic values of $E_1(\infty), E_2(\infty)$ of the estimate provided by Algorithm 3 compared to the initial ones $E_1(0), E_2(0)$, and compared to the asymptotic values of $E^S_1(\infty), E^S_2(\infty)$ of the synchronous deterministic version of our algorithm, proposed in Piovan et al. [2011a]. The asynchronous algorithm performance is comparable to its deterministic version, in terms of the indices E_1, E_2.

5.5.2 Experiments on Cayley graphs

Consider a 2-dimensional Cayley graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = N, |\mathcal{E}| = 2N$, a basis for the cycle space is built as follows. First pick $N-1$ minimal cycles of length 4 that set up the grid, and denote them by $c_1, \dots, c_{N-1}$. Second add the two longitudinal minimal cycles c_N, c_{N+1} of length N (see Fig. 5.5). In this case, $d_{c_i} > |c_i|$ for $i = N, N+1$, precisely $d_{c_i} = N$ is the number of adjacent minimal cycles, while $|c_i| = N$ is the length of the cycle. Therefore $\mathbb{E}[U_e]$ is not stochastic anymore, since $(\mathbb{E}[U_e]\mathbf{1})_{c_i} > 1, i = N, N+1$. Nevertheless a simulation study shows that Algorithm 3 converges also in this case, and lowers the performance indices E_1, E_2, see

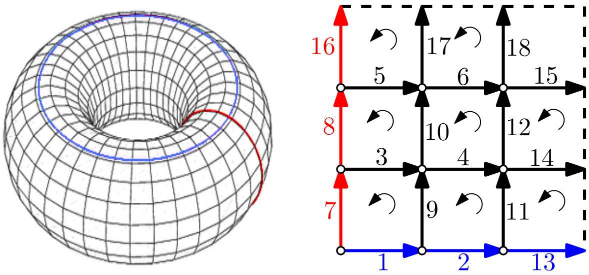


Figure 5.5: This figure shows a 2-dimensional Cayley graph (left). Fix $N = 9$ (right) and consider a minimal cycle basis made of the $N-1 = 8$ cycles of length 4 drawn with counter clockwise orientation, and the two longitudinal cycles $c_9 = \{1,2,13\}$ (blue) and $c_{10} = \{7,8,16\}$ (red), where the numbers denote the edge labels.

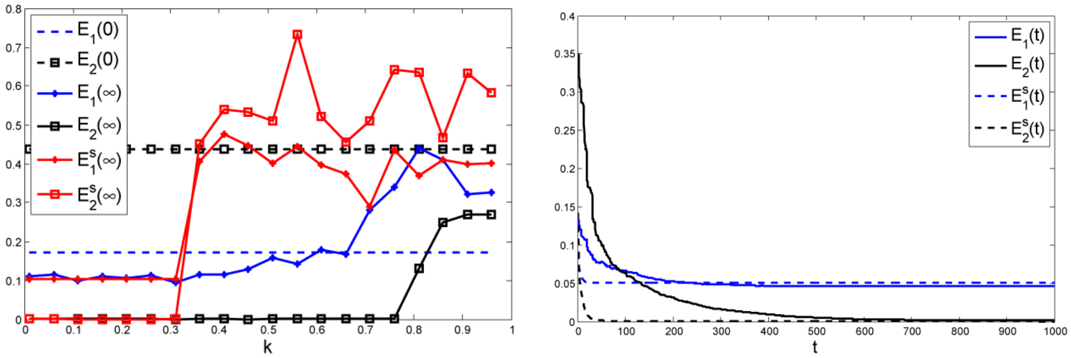


Figure 5.6: Given a Cayley grid $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ with $N = n^2$ nodes, we fix $n = 3$, the time horizon $\tau = 10^5$ and the number of samples $n_s = 10$. The left figure shows the asymptotic values $E_1(\infty)$ (blue dash-star line) and $E_2(\infty)$ (black dash-dot line), defined in Eq. (5.20), while we vary $k \in (0,1)$ in Algorithm 3. The plot shows that if k is greater than a certain threshold, the time required to lower the considered indices becomes greater than 10^5. If we run the synchronous deterministic algorithm in Piovan et al. [2011a], we obtain the corresponding asymptotic values $E_1^S(\infty)$ (red dash-star line) and $E_2^S(\infty)$ (red dash-dot line). The right figure refers to the same graph, with $k = 0.2$, and time horizon $\tau = 10^3$. It shows the evolution in time of $E_1(t)$ (blue line) and $E_2(t)$ (black line), running Algorithm 3, and $E_1^S(t)$ (blue dashed line), $E_2^S(t)$ (black dashed line) running the synchronous deterministic algorithm in Eq. 5.2.

Fig. 5.6. Note that Fig. 5.7 shows a case in which the synchronous version does not converge on a Cayley grid ($k = 0.35$), while the gossip Algorithm defined in 3 does.

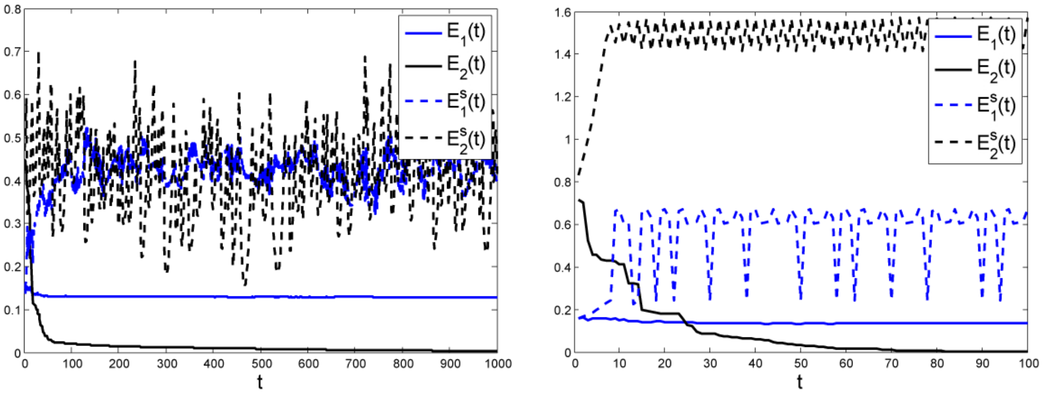


Figure 5.7: Consider the same setup of Fig. 5.6, with stepsize $k = 0.35$. The left figure shows that the synchronous deterministic algorithm in Eq. 5.2 does not converge (a sufficient condition that guarantees convergence is $k < 2/(1+\lambda_{max}(C))$, cfr. Piovan et al. [2011a]), while our Algorithm 3 converges and lowers $E_1(0), E_2(0)$. In the right figure we consider a planar grid with $N = n^2 = 9$, and $k = 0.35$. In both cases $\lambda_{max}(C) = 6$, therefore the latter threshold for the deterministic algorithm is $k < 1/8$, whereas it is sufficient to have $k < 1$ for the gossip algorithm (see Theorem 5.3).

5.6 Conclusions and further work

In this Chapter we consider the calibration problem for a network of cameras, and the main achievement is to provide a distributed and asynchronous algorithm. We cast the latter calibration issue as a constrained minimization problem. The network, modeled by a graph is such that each edge represents an agent. Our goal is to design a simple gossip algorithm with the advantage of being completely distributed and asynchronous. At each time iteration, only one agent is activated randomly, sampled from a uniform distribution, and it updates the corresponding relative orientation. First, we consider and state characterizing properties of the deterministic synchronous version of our algorithm (see Piovan et al. [2011a]). Second, we provide analytical results on the performance properties of the proposed algorithm, for general topologies. We then focus on ring graphs and deeply investigate the optimality of our estimate. Finally, numerical simulations are run to show the effectiveness of the procedure on different graphs.

6 Graph partitioning for camera networks surveillance

In this work we design surveillance trajectories for a network of autonomous cameras to detect intruders in an environment. Intruders, which appear at arbitrary times and locations, are classified as *static* or *dynamic*. While static intruders remain stationary, dynamic intruders are aware of the cameras configuration and move to avoid detection, if possible. As performance criteria we consider the *worst-case detection time* of static and dynamic intruders. We model the environment and the camera network by means of a robotic roadmap. We show that optimal cameras trajectories against static intruders are obtained by solving a continuous graph partitioning problem. We design centralized and distributed algorithms to solve this continuous graph partitioning problem. Our centralized solution relies on tools from convex optimization. For the distributed case, we consider three distinct cameras communication models and propose a corresponding algorithm for each of the models. Regarding dynamic intruders, we identify necessary and sufficient conditions on the cameras locations to detect dynamic intruders in finite time. Additionally, we construct constant-factor optimal trajectories for the case of ring and tree roadmaps.

6.1 Introduction

In this work we focus on the problem of detecting intruders by means of a network of autonomous cameras. In particular, we consider Pan-Tilt-Zoom (PTZ) cameras installed at important locations. We assume the cameras to move their *field of view* (f.o.v.) to cooperatively surveil the whole environment. We develop algorithms for the cameras to self-organize and to detect *intruders* in the environment, that appear at arbitrary locations and times. We consider *static* intruders, which remain stationary, and *dynamic* intruders, which move to avoid detection, if possible. As performance criteria we consider the *worst-case detection time*, that is the longest time needed for the cameras to detect intruders.

Related work. Works related to our camera surveillance problem can be found in the mobile robotics and computer science literatures. In mobile robotics, the patrolling problem consists of scheduling the motion of a team of autonomous agents in order to detect intruders or important events, e.g., see Alberton et al. [2012], Baseggio et al. [2010], Machado et al. [2003], Pasqualetti et al. [2011b]. It should be observed that the patrolling problem and the problem considered in this Chapter significantly differ. Indeed, cameras are fixed at predetermined locations, and their f.o.v.s must lie within the cameras visibility constraints. On the other hand, robots are usually allowed to travel the whole environment, and are usually not subject to visibility constraints. Consequently, algorithms developed for teams of robots are, in general, not applicable in our setup. Similarly, algorithms for graph-clearing and graph-search do not extend to our scenario A. Kehagias and Singh [2009], Kolling and Carpin [2010], Parsons [1978].

In the context of camera networks, the perimeter patrolling problem has recently been studied in Baseggio et al. [2010], Carli et al. [2011], Spindler et al. [2012]. In these works, distributed algorithms are proposed for the cameras to partition a one-dimensional environment, and to synchronize along a trajectory with minimum worst-case detection time of intruders. We improve the results along these directions by, for instance, developing cameras trajectories and partitioning methods for general environment topologies.

In this work we present algorithms for graph partitioning. It is worth noting that our graph partitioning problem differs from classical setups, e.g., see Andreev and Racke [2006], Arkin et al. [2006], Even et al. [2004], G. Even and Schieber [1997]. Indeed, in these works the graph partitioning problem is usually combinatorial, and it consists of partitioning the vertices or the edges as to optimize a certain performance function. Instead, we formulate continuous graph partitioning problems, in which the graph is a physical entity, and the partition is obtained by splitting the edges. As it will be clear in the sequel, our results on graph partitioning are general and applicable to different problems. For instance, if each edge of the graph represents a task to be accomplished by the processors at its endpoints, then our algorithms can be used for dynamic load balancing for multiprocessor networks Cybenko [1989], Lüling et al. [1991].

A preliminary version of this work appeared in Borra et al. [2012b]. Extensions with this work include (i) the design of trajectories against static intruders for cyclic roadmaps, (ii) the design of trajectories against dynamic intruders, and (iii) a characterization of necessary and sufficient conditions for finite detection time of dynamic intruders.

The main contributions of this work are as follows. First, we propose the *continuous graph partitioning problem*, in which a partition of a weighted graph is obtained by splitting the graph edges, and the cost of a partition equals the longest length of its parts (Section 6.2). We show that the continuous graph partitioning problem is convex and non-differentiable, and we characterize its solutions. Then, we derive an equivalent convex and differentiable partitioning problem, which is amenable to distributed implementation.

Second, we define the camera surveillance problem for the detection of static and dynamic intruders (Section 6.3). We model the environment and the camera network by means of a robotic roadmap, and we formalize the worst-case detection time of static and dynamic intruders.

Third, we exhaustively discuss the case of static intruders (Section 6.4). We show that, for tree and ring roadmaps, cameras trajectories with minimum worst-case detection time can be designed by solving a continuous graph partitioning problem. For general cyclic roadmaps, our trajectories based on continuous partitions are proved to be optimal up to a factor 2. However, we conjecture that optimality is achieved also in this case.

Fourth, for the case of dynamic intruders, we derive a necessary and sufficient condition on the cameras locations for the existence of a trajectory with finite detection time (Section 6.5). We focus on ring and tree roadmaps. In particular, for the case of ring roadmaps we design a trajectory with detection time within a factor 3/2 of optimal. Instead, for tree roadmaps, the performance of our trajectory is within a factor 2 of optimal.

Fifth and finally, we consider three different communication models, and we propose distributed algorithms for the cameras for continuous graph partitioning in all these scenarios (Section 6.6). In particular: our first algorithm assumes a synchronous mode of operation of the cameras; our second algorithm assumes an asymmetric broadcast communication model and extends the class of block-coordinate descent algorithms to the constrained case; and our third algorithm only requires gossip communication. We prove convergence of all these algorithms, and we analyze their performance in a simulation study.

6.2 Continuous Partitions of Weighted Graphs

In this section we introduce the problem of continuous graph partitioning. A solution to this problem will be used to design optimal cameras trajectories.

Let $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ be an undirected weighted graph, where $\mathscr{V}$ and $\mathscr{E}$ denote the vertex and edge sets, respectively. Let $\ell_{ij} \in \mathbb{R}_{>0}$ be the weight associated with the edge $\{v_i, v_j\} \in \mathscr{E}$. For a subset of vertices $\mathscr{V}_\mathrm{c} \subseteq \mathscr{V}$, define the i-th set of neighbors as $\mathscr{N}_i = \mathscr{N}_i^\mathrm{in} \cup \mathscr{N}_i^\mathrm{out}$, where

$$\begin{aligned} \mathscr{N}_i^\mathrm{in} &= \{v_j \in \mathscr{V}_\mathrm{c} \,:\, \{v_i, v_j\} \in \mathscr{E}\}, \\ \mathscr{N}_i^\mathrm{out} &= \{v_j \in \mathscr{V} \setminus \mathscr{V}_\mathrm{c} \,:\, \{v_i, v_j\} \in \mathscr{E}\}. \end{aligned} \tag{6.1}$$

A *continuous partition* of the weighted graph $\mathscr{G}$ is a set $\mathscr{P} = \{\mathscr{P}_1, \ldots, \mathscr{P}_n\}$, where (see Fig. 6.1)

$$\mathscr{P}_i = \bigcup_{v_j \in \mathscr{N}_i} [v_i,\, v_{ij}], \tag{6.2}$$

and $v_{ij} \in [v_i, v_j]$ is defined by some $\alpha_{ij} \in [0,1]$ as[1]

$$v_{ij} = \begin{cases} v_i + \alpha_{ij}(v_j - v_i), & \text{if } j \in \mathscr{N}_i^\mathrm{in}, i < j, \\ v_i + (1 - \alpha_{ji})(v_j - v_i), & \text{if } j \in \mathscr{N}_i^\mathrm{in}, i > j, \\ v_j, & \text{if } j \in \mathscr{N}_i^\mathrm{out}. \end{cases} \tag{6.3}$$

The dimension of a continuous partition equals $\mathscr{L}(\mathscr{P}) = \max\{L_1, \ldots, L_n\}$, where L_i is the sum of the lengths of the segments in $\mathscr{P}_i$, $i = 1, \ldots, n$. In other words

$$L_i = \sum_{v_j \in \mathscr{N}_i^\mathrm{in}, i<j} \alpha_{ij}\ell_{ij} + \sum_{v_j \in \mathscr{N}_i^\mathrm{in}, i>j} (1 - \alpha_{ji})\ell_{ij} + \sum_{v_j \in \mathscr{N}_i^\mathrm{out}} \ell_{ij}. \tag{6.4}$$

Let L and $\boldsymbol{\alpha}$ be the vectors of L_i and α_{ij}, respectively. Notice that a continuous partition is entirely specified by a parameters vector $\boldsymbol{\alpha}$.

Let $\mathscr{G}(\mathscr{V}_\mathrm{c}) = (\mathscr{V}_\mathrm{c}, \mathscr{E}_\mathrm{c})$ be the subgraph of $\mathscr{G}$ induced by the vertices $\mathscr{V}_\mathrm{c}$, where $\mathscr{E}_\mathrm{c} = (\mathscr{V}_\mathrm{c} \times \mathscr{V}_\mathrm{c}) \cap \mathscr{E}$.

[1] Given any two points $x_i, x_j \in \mathbb{R}^m$ for some $m \in \mathbb{N}$, we let $[x_i, x_j] = \{t x_i + (1-t) x_j \,:\, t \in [0,1]\}$.

Define the weighted *incidence matrix* $A \in \mathbb{R}^{|\mathcal{V}_\mathrm{c}| \times |\mathcal{E}_\mathrm{c}|}$ as

$$A_{i,e} = \begin{cases} \ell_{ij}, & \text{if } e = \{v_i, v_j\} \in \mathcal{E}_\mathrm{c},\ i < j, \\ -\ell_{ij}, & \text{if } e = \{v_i, v_j\} \in \mathcal{E}_\mathrm{c},\ i < j, \\ 0, & \text{otherwise}, \end{cases} \tag{6.5}$$

and the weighted *incidence vector* $\boldsymbol{b} \in \mathbb{R}^{|\mathcal{V}_\mathrm{c}|}$ as

$$\boldsymbol{b}_i = \sum_{v_j \in \mathcal{N}_i^\mathrm{in}, i > j} \ell_{ij} + \sum_{v_j \in \mathcal{N}_i^\mathrm{out}} \ell_{ij}. \tag{6.6}$$

Notice that $L = A\boldsymbol{\alpha} + \boldsymbol{b}$, and that for every $\boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{E}_\mathrm{c}|}$ it holds

$$\|A\boldsymbol{\alpha} + \boldsymbol{b}\|_1 = \sum_{\{v_i, v_j\} \in \mathcal{E}} \ell_{ij}.$$

Let $\mathbf{0}$ and $\mathbf{1}$ be the vectors of all zeros and ones, respectively. We address the following minimization problem.

Problem 6.1. (Continuous min-max partition) *Given a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a subset of vertices $\mathcal{V}_\mathrm{c} \subseteq \mathcal{V}$, let A and $\boldsymbol{b}$ be as in* (6.5) *and* (6.6)*, respectively. Determine a continuous partition $\boldsymbol{\alpha}_\infty^*$ satisfying*

$$\|A\boldsymbol{\alpha}_\infty^* + \boldsymbol{b}\|_\infty = \min_{\underline{\boldsymbol{\alpha}} \leq \boldsymbol{\alpha} \leq \overline{\boldsymbol{\alpha}}} \|A\boldsymbol{\alpha} + \boldsymbol{b}\|_\infty, \tag{6.7}$$

for some constraints vectors $\underline{\boldsymbol{\alpha}}$, $\overline{\boldsymbol{\alpha}}$ with $\mathbf{0} \leq \underline{\boldsymbol{\alpha}} \leq \overline{\boldsymbol{\alpha}} \leq \mathbf{1}$.

Notice that (6.7) is a convex minimization problem, for which efficient centralized solvers exist Boyd and Vandenberghe [2004]. On the other hand, since (6.7) is not differentiable, distributed solvers may be difficult to implement. We next derive an equivalent differentiable minimization problem, which is instead amenable to distributed implementation.

Problem 6.2. (Continuous min partition) *Given a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a subset of vertices $\mathcal{V}_\mathrm{c} \subseteq \mathcal{V}$, let A and $\boldsymbol{b}$ be as in* (6.5) *and* (6.6)*, respectively. Determine a continuous partition $\boldsymbol{\alpha}_2^*$ satisfying*

$$\|A\boldsymbol{\alpha}_2^* + \boldsymbol{b}\|_2 = \min_{\underline{\boldsymbol{\alpha}} \leq \boldsymbol{\alpha} \leq \overline{\boldsymbol{\alpha}}} \|A\boldsymbol{\alpha} + \boldsymbol{b}\|_2. \tag{6.8}$$

for some constraints vectors $\underline{\boldsymbol{\alpha}}, \overline{\boldsymbol{\alpha}}$ *with* $\mathbf{0} \leq \underline{\boldsymbol{\alpha}} \leq \overline{\boldsymbol{\alpha}} \leq \mathbf{1}$.

Remark 6.3. (Uniqueness of partitions) *Since the minimization problem* (6.8) *is strictly convex, the continuous min partitioning problem admits a unique minimum value, and the set of minimizers is a singleton if and only if the matrix* A *has a trivial null space. It can be shown that* A *has a trivial null space if and only if the induced graph* $\mathcal{G}(\mathcal{V}_{\mathrm{c}})$ *is a tree. In particular, if the graph* $\mathcal{G}(\mathcal{V}_{\mathrm{c}})$ *is connected, then the dimension of the null space of* A *equals* $|\mathcal{E}_{\mathrm{c}}| - |\mathcal{V}_{\mathrm{c}}| + 1$.

Remark 6.4. (Unconstrained partitions) *Consider the (unconstrained) partitioning problems* (6.7) *and* (6.8) *with* $\underline{\boldsymbol{\alpha}}, \overline{\boldsymbol{\alpha}} \in \mathbb{R}^{|\mathcal{E}_{\mathrm{c}}|}$. *It can be verified that* $\boldsymbol{\alpha}^*$ *is a minimizer of Problem* (6.7) *if and only if it is a minimizer of Problem* (6.8). *Moreover, every minimizer can be written as* $\boldsymbol{\alpha}^* = A^{\dagger}(\boldsymbol{v} - \boldsymbol{b}) + \boldsymbol{w}$, *where* $\boldsymbol{v} = \left(\sum_{\{v_i,v_j\}\in\mathcal{E}} \ell_{ij}/n\right)\mathbf{1}$, $A^{\dagger}$ *is the pseudoinverse of* A, $\boldsymbol{w}$ *satisfies* $A\boldsymbol{w} = \mathbf{0}$ *and* $\mathbf{0} \leq \boldsymbol{\alpha}^* \leq \mathbf{1}$.

We next show a relation between min-max partitions and min partitions.

Theorem 6.5. (Min-max and min partitions) *Let* $\boldsymbol{\alpha}_2^*$ *be a min partition solution to Problem* 6.2. *Then,* $\boldsymbol{\alpha}_2^*$ *is also a solution to Problem* 6.1, *that is,*

$$\|A\boldsymbol{\alpha}_2^* + \boldsymbol{b}\|_{\infty} = \min_{\underline{\boldsymbol{\alpha}} \leq \boldsymbol{\alpha} \leq \overline{\boldsymbol{\alpha}}} \|A\boldsymbol{\alpha} + \boldsymbol{b}\|_{\infty}.$$

In order to prove Theorem 6.5, we introduce the following definitions and results. Given a partition $\mathscr{P} = \{\mathscr{P}_1, \ldots, \mathscr{P}_n\}$ defined by $\boldsymbol{\alpha}$, the *maximal graph* associated with $\boldsymbol{\alpha}$ is $\mathcal{G}^{\max} = (\mathcal{V}_{\mathrm{c}}^{\max}, \mathcal{E}_{\mathrm{c}}^{\max})$, where $\mathcal{V}_{\mathrm{c}}^{\max} = \{v_i \in \mathcal{V}_{\mathrm{c}} : L_i = \max\{L_1, \ldots, L_n\}\}$, and $\mathcal{E}_{\mathrm{c}}^{\max} = (\mathcal{V}_{\mathrm{c}}^{\max} \times \mathcal{V}_{\mathrm{c}}^{\max}) \cap \mathcal{E}$.

Lemma 6.6. (Maximal graph) *Let* $\boldsymbol{\alpha}^*$ *be a min partition of the graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, *and let* $\mathcal{G}^{\max} = (\mathcal{V}_c^{max}, \mathcal{E}_c^{max})$ *be the maximal graph associated with* $\boldsymbol{\alpha}^*$. *Then, for each* $v_i \in \mathcal{V}_c^{max}$, $v_j \in \mathcal{V}_{\mathrm{c}} \setminus \mathcal{V}_c^{max}$ *with* $\{v_i, v_j\} \in \mathcal{E}_{\mathrm{c}}$, *it holds* $\alpha_{ij}^* = \underline{\alpha}_{ij}$ *if* $i < j$, *and* $\alpha_{ij}^* = \overline{\alpha}_{ij}$ *if* $i > j$.

Proof. Let $\boldsymbol{\alpha}^*$ be a min partition, and let $L^* = A\boldsymbol{\alpha}^* + \boldsymbol{b}$. By definition, $\mathcal{V}_{\mathrm{c}}^{\max}$ is the set of vertices v_i such that $L_i^* = \|L^*\|_{\infty}$. Let $v_i \in \mathcal{V}_{\mathrm{c}}^{\max}$, and partition its neighbor set as $\mathcal{N}_i^{\mathrm{in}} = \mathcal{N}_i^1 \cup \mathcal{N}_i^2$, where $\mathcal{N}_i^2 = \mathcal{N}_i^{\mathrm{in}} \cap \mathcal{V}_{\mathrm{c}}^{\max}$ and $\mathcal{N}_i^1 = \mathcal{N}_i^{\mathrm{in}} \setminus \mathcal{N}_i^2$. Suppose by contradiction that $\alpha_{ij}^* > \underline{\alpha}_{ij}$ for some $v_j \in \mathcal{N}_i^1$ with $i < j$. Define $\hat{\boldsymbol{\alpha}}$ from $\boldsymbol{\alpha}^*$ by modifying only the entry $\hat{\alpha}_{ij} = \alpha_{ij}^* - \epsilon$, with $\epsilon \in \mathbb{R}_{>0}$. Let $\hat{L} = A\hat{\boldsymbol{\alpha}} + \boldsymbol{b}$, and let ϵ be such that $\hat{\alpha}_{ij} \geq \underline{\alpha}_{ij}$ and $\hat{L}_i > \hat{L}_j$. An equivalent condition for $\hat{L}_i > \hat{L}_j$ is $c_1 - c_2 > \ell_{ij}(\alpha_{ji}^* - \alpha_{ij}^* + 2\epsilon)$, where $c_1 = L_i^* - \alpha_{ij}^* \ell_{ij}$, $c_2 = L_j^* - \alpha_{ji}^* \ell_{ij}$, and $\alpha_{ij} = 1 - \alpha_{ji}$

for $i > j$. It follows that

$$\begin{aligned}\|L^*\|_2^2 - \|\hat{L}\|_2^2 &= (L_i^*)^2 - \hat{L}_i^2 + (L_j^*)^2 - \hat{L}_j^2 \\ &= (c_1 + \alpha_{ij}^* \ell_{ij})^2 - (c_1 + \hat{\alpha}_{ij}\ell_{ij})^2 + (c_2 + \alpha_{ji}^* \ell_{ij})^2 - (c_2 + \hat{\alpha}_{ji}\ell_{ij})^2 \\ &= 2\epsilon\ell_{ij}(c_1 - c_2 + \ell_{ij}(\alpha_{ij}^* - \alpha_{ji}^* - \epsilon)) > 2\epsilon^2\ell_{ij}^2 > 0,\end{aligned}$$

which contradicts our assumption of $\boldsymbol{\alpha}^*$ being a min partition. We conclude that $\alpha_{ij} = \underline{\alpha}_{ij}$. The case of $\alpha_{ij}^* = \overline{\alpha}_{ij}$ is treated analogously, and the theorem follows. □

We are now ready to prove Theorem 6.5.

Proof. Let $\boldsymbol{\alpha}^*$ be a min partition. Recall from Lemma 6.6 that there exists a set of cameras $\mathcal{V}_{\mathrm{c}}^{\max}$ such that (i) $L_i^* = \|L^*\|_\infty$ for all $v_i \in \mathcal{V}_{\mathrm{c}}^{\max}$, and (ii) $\alpha_{ij}^* = \underline{\alpha}_{ij}$ (resp. $\alpha_{ij}^* = \overline{\alpha}_{ij}$) if $i < j$ (resp. $i > j$) for all $\{v_i, v_j\} \in \mathcal{E}_{\mathrm{c}}$ with $v_i \in \mathcal{V}_{\mathrm{c}}^{\max}$ and $v_j \in \mathcal{V}_{\mathrm{c}} \setminus \mathcal{V}_{\mathrm{c}}^{\max}$. Notice also that, because of the cameras constraints and property (ii), it holds

$$\min_{\underline{\boldsymbol{\alpha}} \le \boldsymbol{\alpha} \le \overline{\boldsymbol{\alpha}}} \sum_{v_i \in \mathcal{V}_{\mathrm{c}}^{\max}} (A\boldsymbol{\alpha} + \boldsymbol{b})_i \ge |\mathcal{V}_{\mathrm{c}}^{\max}| \|L^*\|_\infty.$$

Let $\hat{\boldsymbol{\alpha}}$ be such that $\|\hat{L}\|_\infty < \|L^*\|_\infty$. Then $\hat{L}_i < L_i^*$ for all $v_i \in \mathcal{V}_{\mathrm{c}}^{\max}$. It follows that

$$\sum_{v_i \in \mathcal{V}_{\mathrm{c}}^{\max}} (A\hat{\boldsymbol{\alpha}} + \boldsymbol{b})_i < |\mathcal{V}_{\mathrm{c}}^{\max}| \|L^*\|_\infty,$$

which contradicts our hypothesis. □ □

Distributed algorithms to compute continuous graphs partitions are presented in Section 6.6. In the next section we discuss the relationship between continuous graph partitions and the design of trajectories for camera network surveillance.

6.3 Setup for Camera Surveillance

In this section we describe our setup and we introduce some concepts which will be extensively used to state our results.

6.3.1 Problem Setup

We consider the problem of surveilling an environment of interest by means of a camera network. We represent the environment with an undirected weighted roadmap $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, where $\mathscr{V}$ and $\mathscr{E}$ denote the vertex and the edge sets, respectively LaValle [2006]. In particular, each vertex $v_i \in \mathscr{V}$ corresponds to a location in the environment, and $\{v_i, v_j\} \in \mathscr{E}$ if and only if the segment $[v_i, v_j]$ joining vertices v_i and v_j belongs to the environment (vertices v_i and v_j are within line of sight). Finally, the weight of the edge $\{v_i, v_j\} \in \mathscr{E}$ equals the length $\ell_{ij} = \|v_i - v_j\|_2$, and $\ell^{\max} = \max\{\ell_{ij} : \{v_i, v_j\} \in \mathscr{E}\}$.

Cameras are placed at the locations $\mathscr{V}_c \subseteq \mathscr{V}$. Define the set of *neighboring cameras* $\mathscr{N}_i$ as in (6.1). The concept of neighboring cameras will be exploited in Section 6.6 to design distributed algorithms for the cameras.

Let $x_i(t)$ denote the position at time t of the f.o.v. of the i-th camera. We assume that each camera has a limited visibility range along each adjacent edge. In particular,

(A1) at all times t, the i-th f.o.v. is a point along the segment $[v_i, v_j]$ for some $v_j \in \mathscr{N}_i$;

(A2) the speed of the i-th f.o.v. belongs to the set $\{0, 1\}$, that is, the f.o.v. of camera c_i either is stationary at some point or it moves at maximum (unitary) speed;

(A3) for each $v_j \in \mathscr{N}_i$, a point $v_{ij} \in [v_i, v_j]$ is given such that $x_i(t) \in [v_i, v_{ij}]$ at all times t;

(A4) the cameras locations set $\mathscr{V}_c$ satisfies

$$\bigcup_{v_i \in \mathscr{V}_c} \Big\{\{v_i, v_j\} \in \mathscr{E} : v_j \in \mathscr{V}\Big\} = \mathscr{E},$$

so that the roadmap $\mathscr{G}$ is jointly visible by the cameras.

Our setup is illustrated in Fig. 6.1.

6.3.2 Cameras trajectory

A *cameras trajectory* is a set of n continuous functions $X = \{x_1, \dots, x_n\}$, where $x_i : \mathbb{R}_{\geq 0} \to \mathscr{E}$ describes the position of the i-th f.o.v. along the roadmap $\mathscr{G}$. We focus on *periodic* cameras trajectories, for which there exists a finite time $T \in \mathbb{R}_{\geq 0}$ satisfying $X(t + T) = X(t)$ for all $t \in \mathbb{R}_{\geq 0}$.

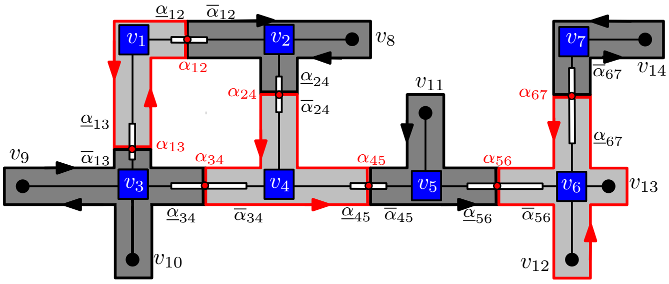


Figure 6.1: This figure shows an environment to be surveilled by a camera network. The environment is represented by a roadmap $\mathscr{G} = (\mathscr{V},\mathscr{E})$ with $\mathscr{V} = \{v_1,\dots,v_{14}\}$. Edges $\mathscr{E}$ are denoted with solid black lines. Cameras are installed at the locations $\mathscr{V}_c = \{v_1,\dots,v_7\}$. White rectangles along the edges represent cameras visibility constraints, and the parameters $\boldsymbol{\alpha}$ define a continuous partition of $\mathscr{G}$. Finally, the DF-Trajectory associated with the partition given by $\boldsymbol{\alpha}$ is identified by the closed paths around the cameras.

Define the *image* of the i-th camera as the set of points visited by the i-th f.o.v. in any period of length T, i.e.,

$$\mathrm{Im}(x_i) = \cup_{t\in[0,T]}\, x_i(t),$$

and the cameras *image set* as $\mathscr{I}^X = \{\mathrm{Im}(x_1),\dots,\mathrm{Im}(x_n)\}$.

Trajectory 4 DF-Trajectory for i-th camera

```
Input: Parameters $\alpha_{ij}$ and set of neighbors $\mathscr{N}_i$;
Set $s_i(t) = (t v_{ij} + (\alpha_{ij}\ell_{ij} - t)v_i)/\alpha_{ij}\ell_{ij}$, for $t \in [0, \alpha_{ij}\ell_{ij}]$;
Set $t_0 = 0$;
for $v_j \in \mathscr{N}_i$ do
    Set $v_{ij}$ as in Eq. (6.3);
    $x_i(t) = s_i(t - t_0)$, for $t \in [t_0, t_0 + \alpha_{ij}\ell_{ij}]$;
    $x_i(t) = s_i(2\alpha_{ij}\ell_{ij} - (t - t_0))$, for $t \in [t_0 + \alpha_{ij}\ell_{ij}, t_0 + 2\alpha_{ij}\ell_{ij}]$;
    $t_0 = t_0 + 2\alpha_{ij}\ell_{ij}$;
```

We now define a particular cameras trajectory associated with a continuous roadmap partition. The optimality properties of this trajectory will be shown in the subsequent sections. Let $\boldsymbol{\alpha}^{\mathrm{df}}$ define the continuous partition $\mathscr{P}^{\mathrm{df}}$ as in (6.2). The *DF-Trajectory* X^{df} with image set $\mathscr{P}^{\mathrm{df}}$ is obtained by letting each camera sweep its subroadmap in a depth-first order Diestel [2000], and it is formally described in Trajectory 4. See Fig. 6.1 for a graphical illustration.

6.3.3 Performance criteria

In this work we design cameras trajectories to detect intruders along the roadmap. We consider both *static*, and *dynamic* intruders. The trajectory of an intruder is a continuous function $p : \mathbb{R}_{\geq 0} \to \mathscr{E}$. Let Π_{d} be the set of all possible intruder trajectories, and let Π_{s} the set of static intruder trajectories ($p(t) = p_0$ for all $t \geq t_0$ and for some $p_0 \in \mathscr{E}$).

An intruder is detected as soon as its position coincides with the f.o.v. of a camera. We define the *worst-case detection time* of a cameras trajectory as the longest time for the detection of an intruder. In particular, for an intruder appearing at time t_0 and moving with trajectory p, and a cameras trajectory X, let

$$t^*(t_0, p, X) = \min\big\{\{t - t_0 \,:\, t > t_0,\, p(t) \in X(t)\} \cup \{\infty\}\big\}.$$

We define the static worst-case detection time as

$$\mathrm{WDT}_{\mathrm{s}}(X) := \sup_{p \in \Pi_{\mathrm{s}},\, t_0 \in [0,T]} t^*(t_0, p, X), \tag{6.9}$$

and the dynamic worst-case detection time as

$$\mathrm{WDT}_{\mathrm{d}}(X) := \sup_{p \in \Pi_{\mathrm{d}},\, t_0 \in [0,T]} t^*(t_0, p, X). \tag{6.10}$$

For the ease of notation we define

$$\mathrm{WDT}_{\mathrm{s}}^* = \inf_{X \in \Omega} \mathrm{WDT}_{\mathrm{s}}(X), \ \ \mathrm{WDT}_{\mathrm{d}}^* = \inf_{X \in \Omega} \mathrm{WDT}_{\mathrm{d}}(X).$$

We conclude this section by observing that $\mathrm{WDT}_{\mathrm{s}}(X) \leq \mathrm{WDT}_{\mathrm{d}}(X)$ for any cameras trajectory X, and that for any periodic cameras trajectory X, $\mathrm{WDT}_{\mathrm{s}}(X) < \infty$ *if and only if* the entire roadmap is persistently surveilled by the cameras, i.e., $\mathscr{E} \subseteq \mathrm{Im}(X)$. Necessary and sufficient conditions for a trajectory to have finite dynamic detection time are discussed in Section 6.5.

6.4 Camera Trajectory for Static Intruders

This section contains our results for the detection of static intruders.

6.4.1 Main results for static intruders

Consider a roadmap $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ with cameras locations $\mathscr{V}_c$. Let $\mathscr{P}^* = \{\mathscr{P}_1^*, \dots, \mathscr{P}_n^*\}$ be a continuous partition of $\mathscr{G}$ of cardinality $|\mathscr{V}_c| = n$ with smallest dimension, that is

$$\max_{i \in \{1,\dots,n\}} L_i^* = \min_{\mathscr{P}} \max_{i \in \{1,\dots,n\}} L_i, \tag{6.11}$$

where $\mathscr{P} = \{\mathscr{P}_1, \dots, \mathscr{P}_n\}$ is a continuous partition of $\mathscr{G}$. Let X^* be the DF-Trajectory associated with the partition $\mathscr{P}^*$. Recall that the roadmap $\mathscr{G}$ with cameras locations $\mathscr{V}_c$ is a *tree* if the induced graph $\mathscr{G}(\mathscr{V}_c)$ contains no cycles, and it is a *ring* if $\mathscr{G}(\mathscr{V}_c)$ consists of a single cycle Diestel [2000].

Theorem 6.7. (Static worst-case detection for DF-Trajectory) *Consider a roadmap $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ with cameras locations $\mathscr{V}_c$ and $|\mathscr{V}_c| = n$. Let X^* be the DF-Trajectory associated with a continuous partition $\mathscr{P}^*$ of $\mathscr{G}$ with smallest dimension. Then,*

(i) $\mathrm{WDT}_s(X^*) = 2\mathscr{L}(\mathscr{P}^*)$ *, and*

(ii) $\mathrm{WDT}_s(X^*) \leq 2\,\mathrm{WDT}_s^*$.

Moreover, if $\mathscr{G}$ is a tree or a ring, then $\mathrm{WDT}_s(X^*) = \mathrm{WDT}_s^*$.

In Theorem 6.7 we show that cameras trajectories designed from a continuous roadmap partition achieve detection performance within a constant factor of optimal. Since optimality is guaranteed for tree and ring roadmaps, we state the following conjecture.

Conjecture 2. (Optimality for cyclic roadmaps) *Motivated by our results in Theorem 6.7, we conjecture that* $\mathrm{WDT}_s(X^*) = \mathrm{WDT}_s^*$ *also for cyclic roadmaps.*

6.4.2 Proof of Theorem 6.7

In this section we derive a proof of Theorem 6.7. We start by introducing the necessary notation and some preliminary results. We define a relative order among the cameras as follows. Let $\{c_1, \dots, c_n\}$ be the set of cameras. For the neighboring cameras c_i and c_j let the time t be such that the i-th f.o.v. and the j-th f.o.v. lie on the edge $\{v_i, v_j\}$. Then we say that $c_i \leq c_j$ if $\|x_i(t) - v_i\|_2 \leq \|x_j(t) - v_i\|_2$. If $x_i(t)$ and $x_j(t)$ lie on different edges at time t, our convention is

$x_i(t) \le x_j(t)$ for $i < j$. A camera trajectory is *order-invariant* if the relative order of the cameras is preserved over time.

Theorem 6.8. (Order-invariant cameras trajectory) *Given a roadmap $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ with cameras locations $\mathscr{V}_{\mathrm{c}}$ and a cameras trajectory X, there exists an order-invariant cameras trajectory $\bar{X}$ with* $\mathrm{WDT}_{\mathrm{s}}(X) = \mathrm{WDT}_{\mathrm{s}}(\bar{X})$.

Proof. Let c_i and c_j be adjacent cameras, and assume the existence of $t \ge 0$ such that $x_i(t) = x_j(t)$. Define $t_{ij}^0 = \min\{t \ge 0 \,:\, x_i(t) = x_j(t)\}$ and, recursively, $t_{ij}^n = \min\{t > t_{ij}^{n-1} \,:\, x_i(t) = x_j(t)\}, n \in \mathbb{N}$. An order-invariant trajectory can be derived from X permuting the cameras labels as follows ($k = 0, 1, \ldots$):

$$\bar{x}_i(t) = x_j(t) \text{ and } \bar{x}_j(t) = x_i(t) \text{ if } t_{ij}^{2k} \le t \le t_{ij}^{2k+1},$$
$$\bar{x}_i(t) = x_i(t) \text{ and } \bar{x}_j(t) = x_j(t) \text{ otherwise.}$$

Each point along the roadmap is visited at the same times in X and $\bar{X}$, thus $\mathrm{WDT}_{\mathrm{s}}(\bar{X}) = \mathrm{WDT}_{\mathrm{s}}(X)$. □

In general, the images of neighboring cameras may overlap. A camera trajectory is called *non-overlapping* if for every pair c_i and c_j, it holds $\mathrm{Int}(\mathrm{Im}(x_i)) \cap \mathrm{Int}(\mathrm{Im}(x_j)) = \emptyset$, where Int denotes the interior of a set.

Theorem 6.9. (Non-overlapping cameras trajectory for tree and ring roadmaps) *Given a tree (resp. ring) roadmap $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ with cameras locations $\mathscr{V}_{\mathrm{c}}$ and a cameras trajectory X, there exists an order-invariant and non-overlapping cameras trajectory $\bar{X}$ with* $\mathrm{WDT}_{\mathrm{s}}(\bar{X}) \le \mathrm{WDT}_{\mathrm{s}}(X)$.

Proof. Without affecting generality, we assume that the trajectory X is order-invariant (cf. Theorem 6.8). We start by considering tree roadmaps, and we define the trajectory $\bar{X}$ from X as follows.

Let $a_1^0 = |\mathscr{N}_1^{\mathrm{in}}| + 1$, $a_i^0 = |\mathscr{N}_i^{\mathrm{in}}|$ for $i = 2, \ldots, n$ and let $\mathscr{P}_i^0 = \emptyset$ for every $i = 1, \ldots, n$. Iteratively

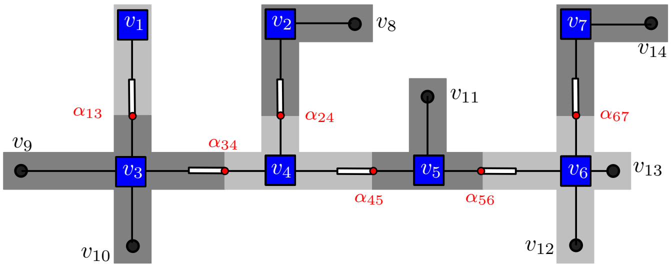


Figure 6.2: This figure shows a tree roadmap, and it illustrates the partitioning procedure in the proof of Theorem 6.9.

perform the following operations:

$$S^k = \{v_i \in \mathcal{V}_\mathrm{c} \,:\, a_i^{k-1} = 1\},$$
$$\mathscr{P}_i^k = \mathrm{Cl}(\mathrm{Im}(x_i) \setminus \left(\cup_{v_j \in \mathcal{N}_i^{\mathrm{in}}} \mathscr{P}_j^{k-1}\right)), \text{ for each } v_i \in S^k,$$
$$a_j^k = a_j^{k-1} - 1, \text{ for each } v_j \in \mathcal{N}_i^{\mathrm{in}} \cup \{v_i\},$$

where $\mathrm{Cl}(\cdot)$ denotes the closure of a set, and $k = 1, 2, \ldots$. After a finite number k_f of iterations, the set $\mathscr{P}^{k_\mathrm{f}} = \{\mathscr{P}_1^{k_\mathrm{f}}, \ldots, \mathscr{P}_n^{k_\mathrm{f}}\}$ is a continuous partition of $\mathscr{G}$. Finally, define the trajectory $\bar{X}$ as the DF-Trajectory on the partition $\mathscr{P}^{k_\mathrm{f}}$. In the interest of space, we do not prove the convergence of the above procedure, and we provide instead an illustration of the final partition in Fig. 6.2.

We now show that $\mathrm{WDT}_\mathrm{s}(\bar{X}) \leq \mathrm{WDT}_\mathrm{s}(X)$. Consider camera c_i, let $|\mathcal{N}_i^{\mathrm{in}}| = n_i$, and let $\partial(\mathscr{P}_i^{k_\mathrm{f}}) = \{v_{i1}, \ldots, v_{i\,n_i}\}$ be the boundary points of $\mathscr{P}_i^{k_\mathrm{f}}$. By construction, there is only one boundary point, say v_{i2}, such that some points in the interior of $[v_i, v_{i2}]$ may be visited by a camera adjacent to c_i with reference to trajectory X (cf. Fig. 6.2). Notice that every boundary point is visited by c_i. Without affecting generality, and possibly after relabeling the boundary points, let t_1 be such that $x_i(t_1) = v_{i1}$, and t_2 such that $x_i(t_2) = v_{i2}$ and no other boundary point is visited in $\mathrm{Int}([t_1, t_2])$. Notice that $t_2 - t_1 \geq \|v_i - v_{i1}\|_2 + \|v_i - v_{i2}\|_2$, and that every boundary point must be visited in the interval $[t_2, t_1 + \mathrm{WDT}_\mathrm{s}(X)]$. It follows $\mathrm{WDT}_\mathrm{s}(X) \geq 2\sum_{j=1}^{n_i} \|v_i - v_{ij}\|_2$, and this must hold for each camera c_i. Finally notice that $\mathrm{WDT}_\mathrm{s}(\bar{X}) = \max_{i \in \{1,\ldots,n\}} 2\sum_{j=1}^{n_i} \|v_i - v_{ij}\|_2$, so that the statement follows. The case of a ring roadmap can be treated analogously. □

We now prove Theorem 6.7.

Proof. Statement (i) follows from the definition of DF-Trajectory, because each camera sweeps its assigned subroadmap at maximum speed along a depth-first tour.

Regarding statement (ii), consider a min partition $\boldsymbol{\alpha}$, and let $\mathcal{G}^{\max} = (\mathcal{V}_{\mathrm{c}}{}^{\max}, \mathcal{E}_{\mathrm{c}}{}^{\max})$ be its associated maximal graph (see Lemma 6.6). Define $\mathrm{Length}(\mathcal{G}^{\max}) = \sum_{\{v_i,v_j\}\in\mathcal{E}_{\mathrm{c}}} \|v_i - v_j\|_2$, and notice that

$$\mathrm{WDT}^*_{\mathrm{s}|\mathcal{G}^{\max}} \geq \frac{\mathrm{Length}(\mathcal{G}^{\max})}{\left|\mathcal{V}_{\mathrm{c}}{}^{\max}\right|},$$

where $\mathrm{WDT}^*_{\mathrm{s}|\mathcal{G}^{\max}}$ denotes the smallest static worst-case detection time for $\mathcal{G}^{\max}$. Indeed, since $L_i = L_j$ for each $v_i, v_j \in \mathcal{V}^{\max}$, each camera needs to sweep (at unitary speed) an image of length $\frac{\mathrm{Length}(\mathcal{G}^{\max})}{|\mathcal{V}_{\mathrm{c}}{}^{\max}|}$ for $\mathcal{G}^{\max}$ to be covered. Moreover, due to Lemma 6.6, cameras outside $\mathcal{G}^{\max}$ cannot visit any point in the interior of $\mathcal{G}^{\max}$. It follows that

$$\mathrm{WDT}^*_{\mathrm{s}} \geq \mathrm{WDT}^*_{\mathrm{s}|\mathcal{G}^{\max}}.$$

Finally, since $\mathrm{WDT}_{\mathrm{s}}(X^*) = 2\frac{\mathrm{Length}(\mathcal{G}^{\max})}{|\mathcal{V}_{\mathrm{c}}{}^{\max}|}$, we conclude that $\mathrm{WDT}_{\mathrm{s}}(X^*) \leq 2\mathrm{WDT}^*_{\mathrm{s}}$.

Consider a tree (resp. ring) roadmap. Due to Theorem 6.9, there exists an order-invariant and non-overlapping trajectory X with $\mathrm{WDT}_{\mathrm{s}}(X) = \mathrm{WDT}^*_{\mathrm{s}}$. To conclude the proof, we have $\mathrm{WDT}_{\mathrm{s}}(X^*) \leq \mathrm{WDT}_{\mathrm{s}}(X)$ since $\mathcal{I}^X$ is a continuous partition.

□

6.5 Cameras Trajectories for Dynamic Intruders

In this section we design cameras trajectories for the detection of dynamic intruders. We start by characterizing a necessary and sufficient condition on the cameras locations for the existence of trajectories with finite dynamic detection time.

Theorem 6.10. (Existence of trajectories with finite dynamic detection time) *Given a roadmap $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with cameras locations $\mathcal{V}_{\mathrm{c}}$, the following statements are equivalent:*

(i) There exists a cameras trajectory X satisfying $\mathrm{WDT}_{\mathrm{d}}(X) < \infty$;

(ii) For every $v_i \in \mathcal{V}_{\mathrm{c}}$ with $|\mathcal{N}_i| \geq 3$, there exists $v_j \in \mathcal{N}_i^{in}$ with $\underline{\alpha}_{ij} = 0$ if $i < j$ and $\overline{\alpha}_{ij} = 1$ if $i > j$.

The following result is useful to prove Theorem 6.10.

Lemma 6.11. (Finite dynamic detection time for single camera) *Given a roadmap* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *with cameras locations* $\mathcal{V}_{\mathrm{c}} = \{v_1\}$*, there exists a cameras trajectory* X *with* $\mathrm{WDT}_{\mathrm{d}}(X) < \infty$ *if and only if* $|\mathcal{N}_1| \leq 2$.

Proof. To show sufficiency, let $|\mathcal{N}_1| \leq 2$, and note that $\mathcal{G}(\mathcal{V}_{\mathrm{c}})$ is a chain. Let x_1 be such that camera c_1 continuously sweeps the chain, and note that $\mathrm{WDT}_{\mathrm{d}}(X) < \infty$.

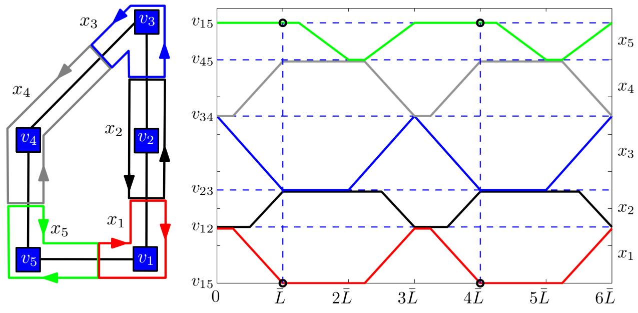
To show necessity of the statement, notice that if $|\mathcal{N}_1| > 2$, an intruder may choose its trajectory p so that $p(t) \neq v_1$ and $p(t+\varepsilon) \in [v_1, v_j]$ whenever $x_1(t) = v_1$ and $x_1(t+\varepsilon) \in [v_1, v_k]$, for $\varepsilon \in \mathbb{R}_{>0}$, $v_j, v_k \in \mathcal{N}_1^{\mathrm{out}}$, and $j \neq k$. □

We are now ready to prove Theorem 6.10.

Proof. In order to show that (ii) is a necessary condition for (i), suppose that (ii) does not hold. Then camera c_i needs to surveil a subroadmap in which $|\mathcal{N}_i| \geq 3$. The statement follows from Lemma 6.11.

We now show that (ii) is also a sufficient condition for (i) by proposing a procedure to clear every subroadmap from intruders appearing at time 0. By periodically repeating this procedure, intruders appearing at different times are also detected. Notice that intruders appearing along the edge $[v_i, v_j]$ can be detected by moving the cameras c_i and c_j towards each other in a way that $x_i(t) = x_j(t)$ for some time t. Starting from v_1, if $|\mathcal{N}_1| \leq 2$, then sweep the adjacent edges by synchronizing the motion of v_1 and its neighboring cameras in any order. If $|\mathcal{N}_1| > 2$, suppose $\{v_1, v_2\} \in \mathcal{V}_{\mathrm{c}}$, then let $\underline{\alpha}_{12} = 0$ from condition (ii). Let camera c_2 sweep the entire edge $\{v_1, v_2\}$ and stop at v_1, and let the other neighboring cameras $c_j \in \mathcal{N}_1^{\mathrm{in}}$ stop at their vertices v_j. Then camera c_1 sequentially sweeps its assigned adjacent segments $[v_1, v_{1j}]$, with $\{v_1, v_j\} \in \mathcal{E}_{\mathrm{c}}$, $j \neq 2$ by synchronizing its motion and those of the neighboring cameras, while keeping the f.o.v. of c_2 at vertex v_1. Notice that any intruder appearing in an edge adjacent to v_1 at time 0 is detected by this procedure. Then let c_1 and its neighboring cameras return to their vertices. Finally iterate the procedure for subsequent cameras in increasing order, and repeat over time. □

In the remainder of this section we design cameras trajectories for the special cases of ring



(a) (b)

Figure 6.3: Consider the ring roadmap $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ with cameras locations $\mathscr{V}_{\mathrm{c}}$ and $n = 5$. Its associated optimal partition $\mathscr{P}$ depicted in Fig. 6.3(a). The corresponding $3\bar{L}$- periodic Ring-Sync-Trajectory described in Trajectory 5 is shown in Fig. 6.3(b).

and tree roadmaps. We refer the reader to Spindler et al. [2012] for a solution to the case of chain roadmaps (cf. Equal-Waiting trajectory).

6.5.1 Ring roadmap

Consider a ring roadmap $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ with cameras locations $\mathscr{V}_{\mathrm{c}}$, and visibility constraints $\underline{\boldsymbol{\alpha}}, \overline{\boldsymbol{\alpha}}$. Let $\mathscr{P}^* = \{\mathscr{P}_1^*, \ldots, \mathscr{P}_n^*\}$ be a min-max partition of $\mathscr{G}$, and let $\bar{L} = \mathscr{L}(\mathscr{P}^*) = \max\{L_1, \ldots, L_n\}$, where L_i is defined in (6.4). Notice that $|\mathscr{N}_i^{\mathrm{in}}| = 2$ for all $v_i \in \mathscr{V}_{\mathrm{c}}$, and that each subroadmap $\mathscr{P}_i^*$ can be written as a segment, parametrized by $s_i : [0, L_i] \to [v_{i-1,i}, v_{i,i+1}]$, such that $s_i(t) = (t v_{i,i+1} + (L_i - t) v_{i-1,i})/L_i$.

Trajectory 5 Ring-Sync-Trajectory for camera c_i

```
Camera $c_i$ surveils $\mathscr{P}_i^* = [v_{i-1,i}, v_i] \cup [v_i, v_{i,i+1}]$;
Set $I_n = [\bar{L}, 2\bar{L}]$, $T_n = \bar{L}$, if $n$ is odd;
    $I_n = \emptyset$, $T_n = 0$, if $n$ even;
if $i$ is even then
    $x_i(t) = v_{i-1,i}$, for $t \in [0, \bar{L} - L_i]$;
    $x_i(t) = s_i(t - (\bar{L} - L_i))$, for $t \in [\bar{L} - L_i, \bar{L}]$;
    $x_i(t) = v_{i,i+1}$, for $t \in I_n \cup [\bar{L} + T_n, 2\bar{L} - L_i + T_n]$;
    $x_i(t) = s_i(2\bar{L} + T_n - t)$, for $t \in [2\bar{L} - L_i + T_n, 2\bar{L} + T_n]$;
else if $i$ is odd then
    $x_i(t) = v_{i,i+1}$, for $t \in [0, \bar{L} - L_i]$;
    $x_i(t) = s_i(\bar{L} - t)$, for $t \in [\bar{L} - L_i, \bar{L}]$;
    $x_i(t) = v_{i-1,i}$, for $t \in I_n \cup [\bar{L} + T_n, 2\bar{L} - L_i + T_n]$;
    $x_i(t) = s_i(t - (2\bar{L} - L_i + T_n))$, for $t \in [2\bar{L} - L_i + T_n, 2\bar{L} + T_n]$;
    if $i = n$ then
        $x_i(t) = v_{i,i+1}$, for $t \in [0, 2\bar{L} - L_i]$;
        $x_i(t) = s_i(2\bar{L} - t)$, for $t \in [2\bar{L} - L_i, 2\bar{L}]$;
        $x_i(t) = v_{i-1,i}$, for $t \in [2\bar{L}, 3\bar{L} - L_i]$;
        $x_i(t) = s_i(t - (3\bar{L} - L_i))$, for $t \in [3\bar{L} - L_i, 3\bar{L}]$;
```

We propose the *Ring-Sync-Trajectory* in Algorithm 5; see Fig. 6.3 for a graphical illustration. We next provide an informal description of the Ring-Sync-Trajectory within its period for the case of an even number of cameras:

(i) the f.o.v. of camera c_i is set at its right (left) boundary point if i is odd (even), so that all the cameras are pairwise synchronized;

(ii) each camera sweeps its subroadmap at maximum speed, and

(iii) each camera stops for $\bar{L} - L_i$ at each boundary point.

If the number of cameras is odd, the algorithm is more involved and it is formally described in Trajectory 5. Notice that, (i) the i-th camera surveils only $\mathscr{P}_i^*$, (ii) the trajectory of each camera is $3\bar{L}$-periodic if n odd, and $2\bar{L}$-periodic if n even, and (iii) the f.o.v.s of each pair of neighboring cameras coincide at some times within the period.

Theorem 6.12. (Dynamic worst-case detection for Ring-Sync-Trajectory) *Given a ring roadmap $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ with cameras locations $\mathscr{V}_\mathrm{c}$, let X^s be the Ring-Sync-Trajectory in Algorithm 5. Then*

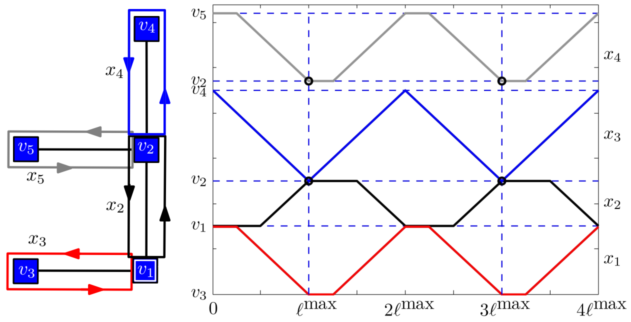
(i) if n is even,

$$\mathrm{WDT}_\mathrm{d}(X^\mathrm{s}) = \mathrm{WDT}_\mathrm{s}(X^\mathrm{s}) = \mathrm{WDT}_\mathrm{d}^*;$$

(ii) if n is odd,

$$\mathrm{WDT}_\mathrm{d}(X^\mathrm{s}) = \mathrm{WDT}_\mathrm{s}(X^\mathrm{s}) \leq \frac{3}{2}\mathrm{WDT}_\mathrm{d}^*.$$

Proof. In order to prove statement (i), consider a ring roadmap with n even, and compute the optimal partition $\mathscr{P}^*$ with dimension $\bar{L}$ (see Section 6.6). Once we synchronize the cameras according to Trajectory 6, the minimum dynamic worst-case detection time is achieved. In order to prove (ii), notice that the Ring-Sync-Trajectory X^s is $3\bar{L}$-periodic when n is odd, and that each camera surveils the corresponding subroadmap $\mathscr{P}_i^*$. The thesis is proved observing that $\mathrm{WDT}_\mathrm{d}^* \geq \mathrm{WDT}_\mathrm{s}^* = 2\bar{L}$. □



(a)

(b)

Figure 6.4: Fig. 6.4(a) show a tree roadmap where v_1 is labeled as root vertex. In Fig. 6.4(b) we report the Tree-Sync-Trajectory described in Trajectory 6. Note that cameras are synchronized, that is, for each pair of adjacent cameras $\{v_i, v_j\} \in \mathcal{E}_c$, there exists $t \in [0, 2\ell^{\max}]$ such that $x_i(t) = x_j(t)$.

6.5.2 Tree roadmap

Consider a tree roadmap $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with cameras locations $\mathcal{V}_c = \mathcal{V} \setminus \{v_1\}$, where node v_1 is labeled as *root*. Assume that each camera can entirely surveil each adjacent edge, that is, $\underline{\boldsymbol{\alpha}} = \mathbf{0}$ and $\overline{\boldsymbol{\alpha}} = \mathbf{1}$. We now design a cameras trajectory for dynamic intruders, and we show that the performance of our trajectory is within a constant factor of optimality. We start by recalling some definitions Diestel [2000]. Then, vertex $v_i \in \mathcal{V}_c$ is a *parent* of vertex $v_j \in \mathcal{V}_c$ (v_i is a *child* of v_j) if $\{v_i, v_j\} \in \mathcal{E}_c$ and v_i lies on the unique shortest path from v_j to v_1. Let v_i^{p} denote the parent of v_i, and let ℓ_i^{p} denote the length of the segment $[v_i, v_i^{\mathrm{p}}]$. Recall that $\ell^{\max}$ is the length of the longest edge. Let us parametrize the segment $[v_i, v_i^{\mathrm{p}}]$ as $s_i : [0, \ell_i^{\mathrm{p}}] \to [v_i, v_i^{\mathrm{p}}]$, $s_i(t) = (t v_i^{\mathrm{p}} + (\ell_i^{\mathrm{p}} - t) v_i)/\ell_i^{\mathrm{p}}$. We divide the vertices into two groups according to the parity of their distance to the root. In particular, we define dist-r(v_i) as the number of edges in the shortest path from v_i to the root v_1.

Trajectory 6 Tree-Sync-Trajectory for camera c_i

```
Camera $c_i$ surveils only the segment $[v_i, {v_i}^{\mathrm{P}}]$;
if dist-r($v_i$) is odd then
    $x_i(t) = {v_i}^{\mathrm{P}}$, for $t \in [0, \ell^{\max} - \ell_i^{\mathrm{p}}]$;
    $x_i(t) = s_i(\ell^{\max} - t)$, for $t \in [\ell^{\max} - \ell_i^{\mathrm{p}}, \ell^{\max}]$;
    $x_i(t) = v_i$, for $t \in [\ell^{\max}, 2\ell^{\max} - \ell_i^{\mathrm{p}}]$;
    $x_i(t) = s_i(t - (2\ell^{\max} - \ell_i^{\mathrm{p}}))$, for $t \in [2\ell^{\max} - \ell_i^{\mathrm{p}}, 2\ell^{\max}]$;
else if dist-r($v_i$) is even then
    $x_i(t) = v_i$, for $t \in \left[0, \ell^{\max} - \ell_i^{\mathrm{p}}\right]$;
    $x_i(t) = s_i(t - (\ell^{\max} - \ell_i^{\mathrm{p}}))$, for $t \in [\ell^{\max} - \ell_i^{\mathrm{p}}, \ell^{\max}]$;
    $x_i(t) = {v_i}^{\mathrm{P}}$, for $t \in [\ell^{\max}, 2\ell^{\max} - \ell_i^{\mathrm{p}}]$;
    $x_i(t) = s_i(2\ell^{\max} - t)$, for $t \in [2\ell^{\max} - \ell_i^{\mathrm{p}}, 2\ell^{\max}]$;
```

We propose the *Tree-Sync-Trajectory* in Trajectory 6 (see Fig. 6.4 for an example). An informal description of the Tree-Sync-Trajectory follows:

(i) the f.o.v. of camera c_i is set at the vertex v_i^{p} (resp. v_i) if dist-r(v_i) is odd (resp. even),

(ii) camera c_i sweeps the segment $[v_i, v_i^{\mathrm{p}}]$, and

(iii) camera c_i stops for $\ell^{\max} - \ell_i^{\mathrm{p}}$ at each boundary point.

Notice that, (i) camera c_i surveils only the segment $[v_i, v_i^{\mathrm{p}}]$, (ii) the trajectory of each camera is $2\ell^{\max}$-periodic, (iii) cameras are *synchronized*, that is $x_i(t_k) = x_i^{\mathrm{p}}(t_k)$ for $t_k = k\ell^{\max}$, with $k \in \mathbb{N}$ even (resp. odd) if dist-r(v_i) is even (resp. odd). Our Tree-Sync-Trajectory in Trajectory 6 extends the concept of Equal-waiting trajectory to the case of tree roadmap Spindler et al. [2012].

Theorem 6.13. (Dynamic worst-case detection for Sync-Trajectory) *Given a tree roadmap $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with cameras locations $\mathcal{V}_{\mathrm{c}} = \mathcal{V} \setminus \{v_1\}$ and $\underline{\boldsymbol{\alpha}} = \mathbf{0}$, $\overline{\boldsymbol{\alpha}} = \mathbf{1}$, let X^{s} be the Tree-Sync-Trajectory in Trajectory 6. Then,*

$$\mathrm{WDT_d}(X^{\mathrm{s}}) = \mathrm{WDT_s}(X^{\mathrm{s}}) \leq 2\mathrm{WDT}_{\mathrm{d}}^{*}.$$

Proof. Consider the Tree-Sync-Trajectory X^{s}. Notice that X^{s} is $2\ell^{\max}$-periodic, that each camera surveils a single edge, and that each edge is surveilled by a different camera. Let an intruder appear at time t_0 along $[v_i, v_i{}^{\mathrm{p}}]$, with $\| v_i - p(t_0) \|_2 \leq \| v_i - x_i(t_0) \|_2$. Then, such intruder is confined in $[v_i, v_i{}^{\mathrm{p}}] \cup \left(\cup_{v_j} [v_i, v_j] \right)$ to avoid detection, where v_j is a child of v_i. Since v_i and all its children synchronize at most every $2\ell^{\max}$, we have $\mathrm{WDT_d}(X^{\mathrm{s}}) = \mathrm{WDT_s}(X^{\mathrm{s}}) = 2\ell^{\max}$. Notice that the case $\| v_i - p(0) \|_2 > \| v_i - x_i(t_0) \|_2$ yields the same conclusion. On the other hand, since each edge can be surveilled by at most two cameras, it holds $\mathrm{WDT}_{\mathrm{d}}^{*} \geq \mathrm{WDT}_{\mathrm{s}}^{*} \geq \ell^{\max}$, which concludes the proof. □

6.6 Distributed Partitioning Algorithms

In what follows we design three distributed algorithms for the continuous min-max partitioning problem. Given an optimal partition, cameras organize along a DF-Trajectory as in Trajectory 4.

Algorithm 7 Synchronous Gradient Partitioning

```
for $v_i \in \mathcal{V}_\mathrm{c}$ do
    Camera $c_i$ receives $S_j^t$ from $c_j$, for all $v_j \in \mathcal{N}_i^{\mathrm{in}}$;
    Set $\alpha_{ij}^{t+1} \leftarrow \alpha_{ij}^t - \varepsilon \ell_{ij}(L_i^t - L_j^t)$;
    if $\alpha_{ij}^{t+1} < \underline{\alpha}_{ij}$ then $\alpha_{ij}^{t+1} = \underline{\alpha}_{ij}$;
    else if $\alpha_{ij}^{t+1} > \overline{\alpha}_{ij}$ then $\alpha_{ij}^{t+1} = \overline{\alpha}_{ij}$;
    Camera $c_i$ transmits $S_i^{t+1}$ to $c_j$, for all $v_j \in \mathcal{N}_i^{\mathrm{in}}$.
```

The algorithms we present rely upon different cameras communication assumptions. We assume each camera to be equipped with a wireless sensor device. In all our algorithms, cameras perform the following operations: (i) receive parameters from (some) neighboring cameras, (ii) update the parameters corresponding to (some) adjacent edges, and (iii) transmit the new values to (some) neighboring cameras. These operations are detailed in the next sections. For convenience, let $S_i^t = \{\alpha_{ij}^t : v_j \in \mathcal{N}_i^{\mathrm{in}}\}$ be the state of camera c_i at iteration $t \in \mathbb{N}$. Finally, we initialize $\alpha_{ij}^0 = \underline{\alpha}_{ij}$ for all $\{v_i, v_j\} \in \mathcal{E}_\mathrm{c}$ with $i < j$.

6.6.1 Synchronous Gradient Partitioning algorithm

The distributed algorithm presented in this section assumes a synchronous mode of operation of the cameras, and it is inspired by the classical gradient projection method Bertsekas and Tsitsiklis [1997]. In particular, every camera performs operations at uniform time instants. The t-th iteration of this algorithm is detailed in Algorithm 7.

Theorem 6.14. ***(Synchronous Gradient Partitioning)*** *For a roadmap $\mathcal{G}$ with cameras locations $\mathcal{V}_\mathrm{c}$, let A and $\boldsymbol{b}$ be as in (6.5) and (6.6), respectively. Let $0 < \varepsilon < (d^{\max}\ell^{\max})^{-1}$, where $d^{\max} = \max\{|\mathcal{N}_i^{\mathrm{in}}| : v_i \in \mathcal{V}_\mathrm{c}\}$. Then, the* Synchronous Gradient Partitioning *algorithm in Algorithm 7 asymptotically converges to $\boldsymbol{\alpha}^*_{\mathrm{SGD}} = \lim_{t\to\infty} \boldsymbol{\alpha}^t$, and*

$$\min_{\underline{\boldsymbol{\alpha}} \le \boldsymbol{\alpha} \le \overline{\boldsymbol{\alpha}}} \|A\boldsymbol{\alpha} + \boldsymbol{b}\|_\infty^2 = \|A\boldsymbol{\alpha}^*_{\mathrm{SGD}} + \boldsymbol{b}\|_\infty^2,$$

where $\underline{\boldsymbol{\alpha}}$ and $\overline{\boldsymbol{\alpha}}$ denote the cameras constraints.

Proof. Note that the update step can be expressed in vector form as

$$\boldsymbol{\alpha}^{t+1} \leftarrow \boldsymbol{\alpha}^t - \varepsilon A^\mathsf{T}(A\boldsymbol{\alpha}^t + \boldsymbol{b}),$$

Algorithm 8 Asymmetric Broadcast Partitioning

```
Camera $c_i$ is randomly selected;
Camera $c_i$ receives $S_j^t$ from camera $c_j$, for all $v_j \in \mathcal{N}_i^{\text{in}}$;
for $v_j \in \mathcal{N}_i^{\text{in}}$ do
    $\alpha_{ij}^{t+1} \leftarrow \alpha_{ij}^t - \varepsilon \ell_{ij}(L_i^t - L_j^t)$;
    if $\alpha_{ij}^{t+1} < \underline{\alpha}_{ij}$ then $\alpha_{ij}^{t+1} = \underline{\alpha}_{ij}$;
    else if $\alpha_{ij}^{t+1} > \overline{\alpha}_{ij}$ then $\alpha_{ij}^{t+1} = \overline{\alpha}_{ij}$;
Camera $c_i$ transmits $S_i^{t+1}$ to camera $c_j$, for all $v_j \in \mathcal{N}_i^{\text{in}}$.
```

and that $A^{\mathsf{T}} A\boldsymbol{\alpha} + A^{\mathsf{T}}\boldsymbol{b}$ is the gradient of the quadratic function $\frac{1}{2}\|A\boldsymbol{\alpha} + \boldsymbol{b}\|_2^2$. Therefore the Synchronous Partitioning algorithm coincides with the gradient projection method Bertsekas and Tsitsiklis [1997]. To conclude the proof note that the gradient of $\frac{1}{2}\|A\boldsymbol{\alpha} + \boldsymbol{b}\|_2^2$ is Lipschitz-continuous with some Lipschitz constant $K \in \mathbb{R}_{>0}$. Thus, for a sufficiently small step size ε, precisely if $0 < \varepsilon < 2/K$, the convergence of $\boldsymbol{\alpha}^t$ to a min partition is guaranteed by [Bertsekas and Tsitsiklis, 1997, Proposition 3.4]. In order to compute the stated upper bound for the stepsize ε, it can be shown that $K \leq 2d^{\max}(\ell^{\max})^2$. Finally, the claimed statement follows from Theorem 6.5. See Bertsekas and Tsitsiklis [1997] for further details. □

6.6.2 Asymmetric Broadcast Partitioning algorithm

The distributed algorithm presented in this section assumes an asymmetric broadcast communication protocol. In particular, at each iteration only one camera updates its state by using local information from its neighboring cameras. In order to guarantee the convergence of the algorithm, we assume the existence of a finite duration $\tau \in \mathbb{R}_{>0}$ such that, for all $t \in \mathbb{R}_{\geq 0}$, every camera in $\mathcal{V}_{\mathrm{c}}$ is selected at least once in the time interval $[t, t+\tau)$ (*partial asynchronism assumption*). The t-th iteration of this algorithm is detailed in Algorithm 8.

Theorem 6.15. ***(Asymmetric Broadcast Partitioning)*** *For a roadmap $\mathcal{G}$ with cameras locations $\mathcal{V}_{\mathrm{c}}$ and cameras edges $\mathcal{E}_{\mathrm{c}}$, let A and $\boldsymbol{b}$ be as in* (6.5) *and* (6.6)*, respectively. Let τ be the partial asynchronism constant, and let $0 < \varepsilon < (K(1+\tau+\tau|\mathcal{E}_{\mathrm{c}}|))^{-1}$, where $K \in \mathbb{R}_{>0}$ is the Lipschitz constant of $\boldsymbol{\alpha} \to A^T(A\boldsymbol{\alpha} + \boldsymbol{b})$. Then, the* Asymmetric Broadcast Partitioning *algorithm in Algorithm 8 asymptotically converges to $\boldsymbol{\alpha}_{\mathrm{AB}}^* = \lim_{t\to\infty} \boldsymbol{\alpha}^t$. Moreover,*

$$\min_{\underline{\boldsymbol{\alpha}} \leq \boldsymbol{\alpha} \leq \overline{\boldsymbol{\alpha}}} \|A\boldsymbol{\alpha} + \boldsymbol{b}\|_\infty^2 = \|A\boldsymbol{\alpha}_{\mathrm{AB}}^* + \boldsymbol{b}\|_\infty^2,$$

where $\underline{\boldsymbol{\alpha}}$ and $\overline{\boldsymbol{\alpha}}$ denote the cameras constraints.

Algorithm 9 Symmetric Gossip Partitioning

```
Neighboring cameras $c_i$ and $c_j$ are randomly selected;
Camera $c_i$ ($c_j$) receives $S_j^t$ ($S_i^t$) from $c_j$ ($c_i$);
$L^* = (L_i^t + L_j^t)/2$;
$\alpha_{ij}^{t+1} = \big(L^* - \sum_{v_k \in \mathcal{N}_i^{\text{in}}, k \neq j} \alpha_{ik}^t \ell_{ik}\big)/\ell_{ij}$;
if $\alpha_{ij}^{t+1} < \underline{\alpha}_{ij}$ then $\alpha_{ij}^{t+1} = \underline{\alpha}_{ij}$;
else if $\alpha_{ij}^{t+1} > \overline{\alpha}_{ij}$ then $\alpha_{ij}^{t+1} = \overline{\alpha}_{ij}$;
Camera $c_i$ transmits $S_i^{t+1}$ to $c_k$, for all $v_k \in \mathcal{N}_i^{\text{in}}$;
Camera $c_j$ transmits $S_j^{t+1}$ to $c_k$, for all $v_k \in \mathcal{N}_j^{\text{in}}$.
```

Proof. As in Theorem 6.14, the algorithm update follows the gradient of $\boldsymbol{\alpha} \to \frac{1}{2}\|A\boldsymbol{\alpha} + \boldsymbol{b}\|_2^2$. Because of the partial asynchronism assumption and the fact that $\boldsymbol{\alpha}^t \in [0,1]^{\mathscr{E}_c}$ is such that $\underline{\boldsymbol{\alpha}} \le \boldsymbol{\alpha}^t \le \overline{\boldsymbol{\alpha}}$ for all $t \in \mathbb{N}$, the statement follows from [Bertsekas and Tsitsiklis, 1997, Section 7, Proposition 5.3] and Theorem 6.5. Note that the bound for the stepsize ε depends on the Lipschitz constant K, the time horizon τ and the number of edges connecting cameras $|\mathscr{E}_c|$. See Bertsekas and Tsitsiklis [1997] for further details. □

6.6.3 Symmetric Gossip partitioning algorithm

The distributed algorithm presented in this section assumes a symmetric gossip-type communication protocol. In particular, at each time iteration only one component of a camera state is updated, and only two adjacent cameras are involved in the computation. The t-th iteration of this algorithm is detailed in Algorithm 9.

Theorem 6.16. ***(Symmetric Gossip Partitioning)*** *For a roadmap $\mathscr{G}$ with cameras locations $\mathscr{V}_c$, let A and $\boldsymbol{b}$ be as in* (6.5) *and* (6.6)*, respectively. Let the partial asynchronism assumption hold. Then, the* Symmetric Gossip Partitioning *algorithm in Algorithm 9 asymptotically converges to $\boldsymbol{\alpha}^*_{\text{SG}} = \lim_{t\to\infty} \boldsymbol{\alpha}^t$. Moreover,*

$$\min_{\underline{\boldsymbol{\alpha}} \le \boldsymbol{\alpha} \le \overline{\boldsymbol{\alpha}}} \|A\boldsymbol{\alpha} + \boldsymbol{b}\|_\infty^2 = \|A\boldsymbol{\alpha}^*_{\text{SG}} + \boldsymbol{b}\|_\infty^2,$$

where $\underline{\boldsymbol{\alpha}}$ and $\overline{\boldsymbol{\alpha}}$ denote the cameras constraints.

Proof. Define $U(\boldsymbol{\alpha}) = \sum_{\{v_i,v_j\}\in\mathscr{E}_c} (L_i - L_j)^2$ as energy storage function. The convergence of the algorithm can be retrieved reasoning along the lines of [Alberton et al., 2012, Theorem IV.1], and by applying Theorem 6.5. - □

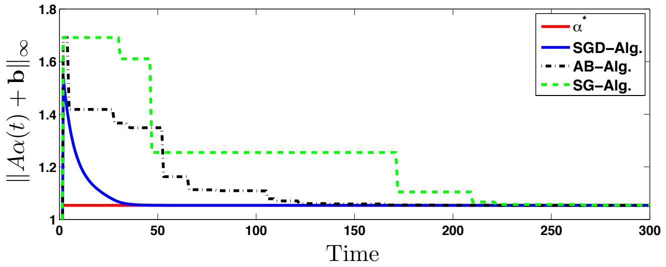


Figure 6.5: This figure shows the convergence of the Synchronous Gradient Partitioning (SGD, blue solid line), the Asymmetric Broadcast Partitioning (AB, black dash-dot line), and the Symmetric Gossip Partitioning algorithms (SG, green dashed line) towards a solution of the continuous min-max partitioning problem. For the simulation we use the configuration in Fig. 6.2, with

To conclude this section, we validate our distributed algorithms through a numerical study. The tree roadmap considered for the simulations is in Fig. 6.2. Notice that $|\mathscr{E}_\mathrm{c}| = 6$, $n = |\mathscr{V}_\mathrm{c}| = 7$, and the number of locations is given by $|\mathscr{V}| = 14$. The stepsizes for Algorithm 8 and Algorithm 9 are chosen sharp, up to a constant $\epsilon = .01$, to their upper bounds stated in Theorem 6.14 and Theorem 6.15, respectively. The results of our simulation study are in Fig. 6.5. Notice that all the proposed algorithms converge to the desired value.

6.7 Conclusion

In this work we design surveillance trajectories for a network of autonomous cameras. We consider both static and dynamic intruders in the environment to be surveilled. As performance criteria we consider the worst-case detection time of intruders. For the case of static intruders, we derive optimal trajectories for ring and tree roadmaps, and constant-factor suboptimal trajectories for general roadmaps. For the case of dynamics intruders, we derive constant-factor suboptimal trajectories for chain, ring, and tree roadmaps. As a complementary result, we introduce the continuous partitioning problem of a weighted graph, and we propose distributed algorithms for its solution.

7 Conclusion and future work

In this Chapter we draw the conclusion of this dissertation and describe several future research directions, which may be followed.

7.1 Summary

This dissertation faces some aspects related to distributed estimation algorithms on graphs.

First, in Chapter 3 we present a general framework in which we aim to estimate the error of relative localization algorithms in sensor networks, focusing on how the error propagates through particular geometric networks. In the considered case, the quantities to be estimated live in some d-dimensional Euclidean space, already investigated in Barooah and Hespanha [2005, 2007]. We consider the intrinsic performance limitations, in terms of the mean error on each component of the optimal estimator of the position vector, applying an additive noise model. The goal is to estimate the relative error on each edge of the graph modeling our multiagent network, in terms of its asymptotic properties when the number of nodes is asymptotically large, and its dependence on the relative error of edges which are sufficiently "far"in the network. Our focus in posed on an exemplary class of networks, called Abelian Cayley networks, for which it is possible to compute an explicit characterization of the relative error by means of the eigenvalues of the Laplacian of the graph. Considering a sequence of Cayley networks with increasing dimension, we analyze the scaling properties of such unavoidable error in the optimal unbiased estimator with minimum variance, depending on the Euclidean dimension of the grid, and making use of the analogy with the effective resistance of the associated electrical resistive network.

Second, we concentrate our attention on camera networks in the more general context of networked control systems. In Chapter 4, we propose two versions of a hybrid algorithm which allows a network of cameras to autonomously calibrate in the plane. The calibration problem is casted in a non-convex non-linear optimization problem, and the proposed algorithm is a two-step procedure. The cost function in not convex, but its domain can be partitioned in convex regions that maintain the convexity of such potential. First, the algorithm estimates a vector of integers $\bar{\boldsymbol{K}}$ coding the region of convexity which the global minimum is believed to belong to. In the second step, the algorithm achieves the minimizer belonging to the selected region of convexity. Analytical characterization of the estimate is provided, with a numerical comparison with other existing algorithms. This procedure permits to understand the properties of the proposed solution and to understand when this solution is correct.

In Chapter 5, we analyze what happens when the interconnection protocol is not deterministic but randomized, asymmetric gossip-like. We propose a novel distributed and completely random and asynchronous procedure, inspired by the work Piovan et al. [2011a]. The algorithm is proved to converge almost surely and in the mean square sense for general planar graphs. If we focus on ring graphs, the proposed algorithm converges for any realization, and the expected value of the limit random variable equals the optimal solution, written in closed form. Numerical experiments are shown to validate our results, and to investigate the case of non-planar graphs.

In Chapter 6, we focus on a different application related to camera networks. The goal is to design surveillance trajectories for a network of autonomous cameras, that minimize the detection time of static and dynamic intruders in the environment to be surveilled. As performance criteria we consider the worst-case detection time of intruders. For general roadmaps, the proposed trajectories are constant-factor suboptimal against static intruders. Whereas, they are optimal for ring and tree roadmaps. In order to detect dynamic intruders, we derive constant-factor suboptimal trajectories for chain, ring, and tree roadmaps. While deriving such camera trajectories, we introduce the continuous partitioning problem of a weighted graph, and we propose distributed algorithms for its solution, that have different communication protocols (synchronous deterministic, asynchronous deterministic, symmetric gossip-like).

7.2 Directions for future research

In this Section, some of the future research directions are described.

First, concerning Chapter 3, our intention is to deepen the analytical study for 2-dimensional Cayley grids, in order to prove Conjecture 1. Moreover a further study, at least numerical, may be followed out for d-dimensional Cayley grids, with $d \geq 3$.

Second, in Chapter 4 and 5 calibration algorithms are provided, for different communication protocols in cooperative large camera networks. Concerning the hybrid algorithm proposed in Chapter 4 for the planar calibration of a camera network, it would be interesting to find a distributed procedure for the nodes to autonomously compute the minimal cycles in a graph, achieving the corresponding cycle vector using only local information. This would permit to completely decentralize the second version of the proposed strategy. Moreover, a more refined performance analysis has to be done, in terms of the error to gain the estimate of the integer vector $\bar{\boldsymbol{K}}$, and in terms of the index $W(\theta)$ proposed in Eq. (4.3). Finally, the generalization of such algorithm to the 3D case would be appealing. On one side, we should properly model the noise, and the regions of convexity, in order to reshape the cost function, naturally showing mutiple local minima. On the other side, a different approach may be considered, generalizing the calibration algorithm proposed in Chapter 5.

Third, in Chapter 5, a performance analysis of our asymmetric gossip algorithm on non-planar graph would be an interesting result. The explicit computation of a global minimum of the considered cost function could help to draw conclusions on the quality of our estimate. Moreover, it is still needed a deeper analytical comparison among the proposed strategy and the existing ones.

Forth, and finally, the patrolling problem presented in Chapter 6 may be extended as follows. Concerning static intruders, one direction is to prove the conjecture that our camera trajectories are optimal also for general roadmaps, avoiding the constant factor suboptimality. On the other hand, camera trajectories against dynamic intruders are much more complicated to be designed, since synchronization among neighboring cameras is required in each time period. Also in this case, the design of optimal synchronized trajectories for general roadmaps is still missing.

Bibliography

G. Hollinger A. Kehagias and S. Singh. A graph search algorithm for indoor pursuit/evasion. *Mathematical and Computer Modelling*, 50(9–10):1305–1317, 2009. URL http://www.sciencedirect.com/science/article/pii/S0895717709002398.

R. Moses et al. A. Savvides, W. Garber. An analysis of error inducing parameters in multihop sensor node localization. *IEEE Transactions on Mobile Computing*, 4(6):567–577, 2005.

H. Aghajan and A. Cavallaro. *Multi-Camera Networks: Principles and Applications*. Academic Press., 2009.

R. Alberton, R. Carli, A. Cenedese, and L. Schenato. Multi-agent perimeter patrolling subject to mobility constraints. June 2012. To appear.

D. Aldous and J. Fill. *Reversible Markov Chains and Random Walks on Graphs*. http://www.stat.berkeley.edu/ aldous/RWG/book.html.

N. Alon and Y. Roichman. Random cayley graphs and expanders. *Random Structures and Algorithms*, 5:271–284, 1994.

K. Andreev and H. Racke. Balanced graph partitioning. *Theory of Computing Systems*, 39: 929–939, 2006. ISSN 1432-4350. URL http://dx.doi.org/10.1007/s00224-006-1350-7.

E. M. Arkin, R. Hassin, and A. Levin. Approximations for minimum and min-max vehicle routing problems. *Journal of Algorithms*, 59(1):1–18, 2006.

L. Babai. Spectra of cayley graphs. *Journal of Combinatorial Theory, Series B*, 27:180–189, 1979.

P. Barooah and J. P. Hespanha. Distributed estimation from relative measurements in sensor networks. In *Proceedings of the 2nd International Conference on Intelligent Sensing and Information Processing*, Dec. 2005.

P. Barooah and J. P. Hespanha. Estimation on graphs from relative measurements: Distributed algorithms and fundamental limits. *IEEE Control System Magazine*, 27(4):57–74, Aug. 2007.

P. Barooah and J. P. Hespanha. Estimation from relative measurements: Electrical analogy and large graphs. 56(6):2181–2193, 2008.

P. Barooah and J. P. Hespanha. Error scaling laws for linear optimal estimation from relative measurements. 55(12):5661–5673, 2009.

P. Barooah, N. M. da Silva, and J. P. Hespanha. Distributed optimal estimation from relative measurements for localization and time synchronization. In *Distributed Computing in Sensor Systems*, volume 4026 of *Lect. Notes in Comput. Science*, pages 266–281. Springer, Berlin, June 2006.

M. Baseggio, A. Cenedese, P. Merlo, M. Pozzi, and L. Schenato. Distributed perimeter patrolling and tracking for camera networks. pages 2093–2098, Atlanta, GA, USA, December 2010.

D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997. ISBN 1886529019.

S. Bolognani, S. Del Favero, L. Schenato, and D. Varagnolo. Consensus-based distributed sensor calibration and least-square parameter identification in wsns. *International Journal of Robust and Nonlinear Control*, 20(2), January 2010. ISSN 1049-8923.

V. Borkar and P. Varaiya. Asymptotic agreement in distributed estimation. 27(3):650–655, 1982.

D. Borra, E. Lovisari, R. Carli, F. Fagnani, and S. Zampieri. Autonomous calibration algorithms for networks of cameras. *Automatica*, a. Submitted.

D. Borra, F. Pasqualetti, and F. Bullo. Continuous graph partitioning for camera network surveillance. *Automatica*, b. Submitted.

D. Borra, E. Lovisari, R. Carli, F. Fagnani, and S. Zampieri. Autonomous calibration algorithms for networks of cameras. In *Proceedings of the Americal Control Conference, ACC'12.*, July 2012a.

D. Borra, F. Pasqualetti, and F. Bullo. Continuous graph partitioning for camera network surveillance. Santa Barbara, CA, USA, September 2012b. To appear.

S. Boyd. Convex optimization of graph Laplacian eigenvalues. In *Int. Congress of Mathematicians*, pages 1311–1319, Madrid, Spain, August 2006.

S. Boyd and L. Vandenberghe. *Convex Optimization.* 2004. ISBN 0521833787.

S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Gossip algorithms: Design, analysis, and application. pages 1653–1664, March 2005.

S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. 52(6):2508–2530, 2006.

J. Caffery and G. Stber. Overview of radiolocation in cdma cellular systems. *IEEE Communications Magazine*, 36(4):38–45, 1998.

R. Carli, A. Cenedese, and L. Schenato. Distributed partitioning strategies for perimeter patrolling. pages 4026–4031, San Francisco, CA, USA, June 2011.

J. Clark and R. Fierro. Mobile robotic sensors for perimeter detection and tracking. *ISA Transactions*, 46(1):3–13, 2007.

G. Cybenko. Dynamic load balancing for distributed memory multiprocessors. 7(2):279–301, 1989.

Y. Peres D.A. Levin and E.L. Wilmer.

M. H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345): 118–121, 1974.

D. Devarajan and R. Radke. Calibrating distributed camera networks using belief propagation. *EURASIP Journal of Applied Signal Processing*, pages 221–221, 2007.

R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics.* 2 edition, 2000.

R. Diestel. *Graph Theory.* Springer. Graduate Texts in Mathematics., August 2005.

M.P. do Carmo. *Riemannian Geometry.* Birkhäuser. Mathematics: Theory & Applications, 1992.

P. G. Doyle and J. L. Snell. *Random Walks and Electrical Networks.* The Mathematical Association of America.

G. Even, N. Garg, J. Könemann, R. Ravi, and A. Sinha. Min–max tree covers of graphs. 32(4): 309–315, 2004.

F. Fagnani and S. Zampieri. Randomized consensus algorithms over large scale networks. *IEEE Journal on Selected Areas in Communications*, 26(4):634–649, 2008.

S. Rao G. Even, J. Naor and B. Schieber. Fast approximate graph partitioning algorithms. In *ACM-SIAM Symposium on Discrete Algorithms*, SODA '97, 1997.

Felix R. Gantmacher. *Matrix Theory*, volume 1. American Mathematical Society, 1990.

A. Ghosh, S. Boyd, and A. Saberi. Minimizing effective resistance of a graph. 50(1):37–66, 2008.

A. Giridhar. *In-Network Computation in Wireless Sensor Network*. PhD thesis, University of Illinois, Urbana-Champaign, 2006.

A. Giridhar and P.R. Kumar. Distributed clock synchronization over wireless networks: Algorithms and analysis. In *In Proceedings of the 45th IEEE Conference on Decision and Control CDC'06*, pages 4915–4920, 2006.

G. Grimmett and D. Stirzaker. *Probability and Random Processes*. 2001. ISBN 0-198-57222-0.

C. M. Grinstead and J. L. Snell. *Introduction to Probability*. 1997. ISBN 0821807498.

G. Gutin. Traveling salesman problems. In J. Gross and J. Yellen, editors, *Handbook of Graph Theory*, chapter 4.6, pages 279–299. CRC Press, Boca Raton, FL, 1 edition, 2003. ISBN 1584880902.

T.W. Hungeford. *Algebra*. Springer. Graduate Texts in Mathematics., December 1980.

A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. 48(6):988–1001, 2003.

R. Govindan K. Chintalapudi, A. Dhariwal and G. Sukhatme. Ad-hoc localization using ranging and sectoring. In *In IEEE Infocomm*, 2004.

S. Kaczmarz. Approximate solution of systems of linear equations. 57(6):1269–1271, 1993. (English translation).

D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *IEEE Symposium on Foundations of Computer Science*, pages 482–491, Washington, DC, October 2003.

D. B. Kingston, R. W. Beard, and R. S. Holt. Decentralized perimeter surveillance using a team of UAVs. 24(6):1394–1404, 2008.

A. Kolling and S. Carpin. Pursuit-evasion on trees by robot teams. 26(1):32–47, 2010.

Y. Kuramoto. Self-entrainment of a population of coupled non-linear oscillators. In H. Araki, editor, *Int. Symposium on Mathematical Problems in Theoretical Physics*, volume 39 of *Lecture Notes in Physics*, pages 420–422. 1975. ISBN 978-3-540-07174-7.

K. S. J. Pister L. Doherty and L. E. Ghaoui. Convex position estimation in wireless sensor network. In *IEEE INFOCOM*, pages 1655–1663, 2001.

S. M. LaValle. *Planning Algorithms*. 2006. ISBN 0521862051. Available at `http://planning.cs.uiuc.edu`.

E. Lovisari. *Synchronization algorithms for multi-agent systems: Analysis, Synthesis and Application*. PhD thesis, University of Padua, 2012.

E. Lovisari, F. Garin, and S. Zampieri. Resistance-based performance analysis of the consensus algorithm over geometric graphs. *SIAM Journal on Control and Optimization*, 2012.

R. Lüling, B. Monien, and F. Ramme. Load balancing in large networks: A comparative study. In *3rd IEEE Symposium on Parallel and Distributed Processing*, pages 686–689, Dallas, TX, USA, December 1991.

N. A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1997. ISBN 1558603484.

Y. Ma, S. Soatto, J. Kosecka, and S. Sastry. *An Invitation to 3D Vision: From Images to Geometric Models*. Springer Verlag, 2003.

A. Machado, G. Ramalho, J. D. Zucker, and A. Drogoul. Multi-agent patrolling: An empirical analysis of alternative architectures. In *Multi-Agent-Based Simulation II*, pages 155–170. 2003.

D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *ACM Conference on Embedded Networked Sensor Systems*, pages 50–61, Baltimore, MD, USA, November 2004.

L. Moreau. Stability of continuous-time distributed consensus algorithms. pages 3998–4003, Nassau, Bahamas, 2004.

Y. Moreno and A. F. Pacheco. Synchronization of Kuramoto oscillators in scale-free networks. *Europhysics Letters*, 68:603, 2004.

H. Balakrishnan N. B. Priyantha, A. K. L. Miu and S. J. Teller. The cricket compass for context-aware mobile applications. *Mobile Computing and Networking*, pages 1–4, 2001.

A. O. H. III N. Patwari and M. Perkins. Relative location estimation in wireless sensor networks. *IEEE Transactions in Signal Processing*, 51(8):2137–2148, 2003.

M. Newman. *Integral Matrices.* Academic Press. Pure and Applied Mathematics. A Series of Mongraphs and Textbooks., 1972.

D. Niculescu and B. Nath. Error characteristics of ad hoc positiong systems (aps). In *MobiHoc'04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, 2004.

J. R. Norris. *Markov Chains (Cambridge Series in Statistical and Probabilistic Mathematics).* Cambridge University Press, 1998. ISBN 0521633966.

G. Notarstefano and F. Bullo. Distributed abstract optimization via constraints consensus: Theory and applications. 56(10):2247–2261, 2011. doi: 10.1109/TAC.2011.2164020.

R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. 95(1):215–233, 2007.

J. O'Rourke. *Art Gallery Theorems and Algorithms.* 1987. ISBN 0195039653.

T. D. Parsons. Pursuit-evasion in a graph. In Y. Alavi and D. Lick, editors, *Theory and Applications of Graphs*, volume 642 of *Lecture Notes in Mathematics*, pages 426–441. 1978.

F. Pasqualetti, R. Carli, A. Bicchi, and F. Bullo. Distributed estimation and detection under local information. pages 263–268, Annecy, France, September 2010.

F. Pasqualetti, R. Carli, and F. Bullo. Distributed estimation via iterative projections with application to power network monitoring. March 2011a. To appear.

F. Pasqualetti, A. Franchi, and F. Bullo. On cooperative patrolling: Optimal trajectories, complexity analysis and approximation algorithms. January 2011b. To appear.

F. Pasqualetti, F. Zanella, J. R. Peters, M. Spindler, R. Carli, and F. Bullo. Camera network coordination for intruder detection. January 2013. Submitted.

G. Piovan, I. Shames, B. Fidan, F. Bullo, and B. D. O. Anderson. On frame and orientation localization for relative sensing networks. February 2011a. To appear (Revised in Sep 2011).

G. Piovan, I. Shames, B. Fidan, F. Bullo, and B. D. O. Anderson. On frame and orientation localization for relative sensing networks. *Automatica*, 2011b.

D. Estrin R. Karp, J. Elson and S. Shenker. Optimal and global time synchronization in sensornets. 2003. Tech. Rep.

W.J. Russell, D.J. Klein, and J.P. Hespanha. Optimal estimation on the graph cycle space. In *American Control Conference. ACC'10*, pages 1918 –1924, 2010.

A. Sarlette. *Geometry and Symmetries in Coordination Control*. PhD thesis, University of Liège, Belgium, January 2009a.

A. Sarlette. *Geometry and Symmetries in Coordination Control*. PhD thesis, University of Liège, 2009b.

A. Sarlette and R. Sepulchre. Consensus optimization on manifolds. 48(1):56–76, 2009a.

A. Sarlette and R. Sepulchre. Consensus optimization on manifolds. *SIAM Journal on Control and Optimization*, 58:56–76, 2009b.

A. Sarlette, R. Sepulchre, and N. E. Leonard. Discrete-time synchronization on the n-torus. Kyoto, Japan, June 2006.

A.V. Savkin. Coordinated collective motion of groups of autonomous mobile robots: Analysis of Vicsek's model. 49(6):981–982, 2004.

L. Scardovi, A. Sarlette, and R. Sepulchre. Synchronization and balancing on the N-torus. 56 (5):335–341, 2007.

A. N. Shiryaev. *Probability*. GTM. 2 edition, 1989. ISBN 0387945490.

A. Singer. Angular synchronization by eigenvectors and semidefinite programming. *Applied and Computational Harmonic Analysis*, 30(1):20–36, 2011.

M. Spindler, F. Pasqualetti, and F. Bullo. Distributed multi-camera synchronization for smart-intruder detection. June 2012. To appear.

B. Sundararaman, U. Buy, and A.D. Kshemkalyani. Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks*, 3(3):281–323, 2005.

S. Susca, S. Martínez, and F. Bullo. Monitoring environmental boundaries with a robotic sensor network. 16(2):288–296, 2008.

A. Terras. *Fourier analysis on finite groups and applications*, volume 43 of *London Mathematical Society Student Texts*. Cambridge University Press, 1999.

R. Tron and R. Vidal. Distributed image-based 3-D localization of camera sensor networks. pages 901–908, Shanghai, China, December 2009a.

R. Tron, B. Afsari, and R. Vidal. Average consensus on riemannian manifolds with bounded curvature. In *Proceedings of the 51th IEEE Conference on Decision and Control CDC'11*, 2011.

Roberto Tron and René Vidal. Distributed image-based 3-d localization of camera sensor networks. In *Proceedings of the 49th IEEE Conference on Decision and Control CDC'09*, pages 901–908, 2009b.

J. N. Tsitsiklis. *Problems in Decentralized Decision Making and Computation.* PhD thesis, Massachusetts Institute of Technology, November 1984. Available at `http://web.mit.edu/jnt/www/Papers/PhD-84-jnt.pdf`.