Università
degli Studi
di Padova

Dipartimento
di Ingegneria
dell'Informazione

Efficient Parametric and Non-Parametric Estimation for Localization and Mapping in Robotic Networks

Ph.D. Candidate
Andrea Carron

Advisor
Luca Schenato

Co-advisor
Ruggero Carli

Ph.D. School in
Information Engineering
2015

To my parents Giuliana and Massimo.

More computing sins are committed in the name of efficiency (without necessarily achieving it) than for any other single reason - including blind stupidity.

W.A. Wulf

Contents

Sommario

Fin dagli anni ottanta i problemi di localizzazione e mapping sono stati argomenti molto studiati nell'ambito della robotica. Un'ulteriore spinta è avvenuta negli ultimi dieci anni grazie all'incremento delle capacità di calcolo dei dispositivi elettronici. Questo ha permesso di porre nuovi ambiziosi obbettivi, come il controllo di sciami robotici, UAVs, veicolo autonomi e reti robotiche. Efficienza, robustezza e scalabilità sono tre caratteristiche fondamentali che non possono mancare negli algoritmi di localizzazione e mapping.

L'*efficienza* è la capacità di un algoritmo di mininizzare l'utilizzo di risorse, in particolare il tempo di utilizzo della CPU e la quantità di memoria utilizzata. Nella applicazioni sopra citate è richiesto l'utilizzo di un mezzo di comunicazione, per *robustezza* quindi intendiamo algoritmi asincroni capaci di funzionare anche in presenza di perdite di pacchetto e ritardi. Per finire con *scalabilità* intendiamo la capacità di un'algoritmo di funzionare senza drammatiche variazioni di prestazioni anche quando il numero di dispositivi coinvolti cresce.

Questa tesi si pone l'obbiettivo di studiare metodi parametri e non parametrici applicati ai problemi di localizzazione e mapping nell'ambito della robotica. In particolare i principali contributi possono essere riassunti nei seguenti quattro argomenti:

(i) **Localizzazione tramite consensus:** Il primo argomento affrontato è dato dal problema di stimare in modo ottimo le posizioni di un gruppo di agenti in una rete. Solamente gli agenti definiti come vicini nel grafo di comunicazione possono scambiarsi misure vettoriali rumorose di distanza. Questo requisito si traduce in una limitata complessità ed il vincolo della sola comunicazione locale, rendendo l'algoritmo indipendente dalla dimensione della rete e dalla sua topologia. Viene quindi proposto un algoritmo di consensus con memoria che ne permette l'implementazione asincrona. Di questo algoritmo è possibile provare la convergenza esponenziale ad una soluzione ottima, sotto le ipotesi di utilizzo di semplici protocolli di comunicazione deterministici o randomizzati e una richiesta minima di trasmissione di pacchetti. Nel caso di comunicazione randomizzata è inoltre presente uno studio della velocità di convergenza in aspettazione e tale risultato viene poi utilizzato per studiare la velocità di convergenza in media quadratica. In particolare viene mostrato che per grafi regolari, come i Cayley, i Ramanjuan ed i completi l'algoritmo proposto, e quello asincrono senza memoria, hanno il medesimo comportamento. Inoltre, l'implementazione asincrona è robusta a ritardi e perdite di pacchetto. Per finire l'analisi analitica è complementata con risultati numerici, comparando l'algoritmo proposto con altri algoritmi presenti in letteratura.

(ii) **Localizzazione distribuita di veicoli aerei:** Successivamente viene studiato il problema della localizzazione distribuita multiagente in presenza di misure eterogenee e comunicazione wireless. L'algoritmo proposto integra misure assolute poco precise, come GPS e bussole, con misure relative più precise, come range e bearing. Le misure assolute sono usate per

ricostruire la posizione e l'orientazione della formazione, mentre quelle relative sono usate per ricostruire le posizioni reciproche degli agenti. Viene proposto un algoritmo distribuito ed asincrono basato su minimi quadrati, che permette di risolvere una versione approssimata del problema di Massima Verosimiglianza non-lineare inizialmente presentato. Tale algoritmo è robusto a perdite di pacchetto e ritardi, inoltre l'uso di un protocollo ACK-less broadcast-based assicura un'efficiente e facile implementazione. Per finire se l'errore sulle misure relative è sufficientemente piccolo, viene mostrato come l'algoritmo raggiunge una soluzione molto vicina a quella del problema originale di massima verosimiglianza. I risultati teorici e le performance dell'algoritmo sono poi verificati attraverso numerose simulazioni Monte-Carlo.

(iii) **Stima e Coverage:** Il terzo argomento studiato d̀ato dal problema di coverage ottimo di una regione attraverso più robot. Si assume non nota a propri la sensory function usata per approssimare la densità di apparizioni degli eventi. Il setup considerato è un'architettura client-server nella quale ogni robot può comunicare con la base-station attraverso una rete di comunicazione soggetta a perdita di pacchetti. Proponiamo un algoritmo di stima basato su regressione Gaussiana che permette di stimare la sensory function con un'accuratezza arbitraria. Per risolvere il problema di coverage è presentata una strategia randomica attraverso la quale i robot mobili e la base-station simultaneamente stimano la distribuzione della density function collezionando misure rumorose e computando le partizioni di Voronoi. Questa strategia è progettata per prima promuovere l'esplorazione e solo successivamente incentivare i robot a spostarsi nel centroide della partizioni di Voronoi stimate. Sotto deboli ipotesi sulla probabilità di errata trasmissione, proviamo che la strategia proposta garantisce la convergenza della density function stimata a quella vera e che le corrispondenti partizioni di Voronoi convergono asintoticamente andando arbitrariamente vicine a quella di Voronoi ottime. Viene anche proposta un'approssimazione numericamente efficiente che trova un compromesso tra la qualità della stima della mappa e le risorse computazionali utilizzate, e.g. memoria e CPU. Per finire, tramite svariate simulazioni, mostriamo l'efficacia dell'approccio proposto.

(iv) **Stima non parametrica di campi spazio-temporali:** Affrontiamo per finire il problema della stima efficiente e ottima di una funzione sconosciuta e tempo variante attraverso la collezione di misure rumorose. Inquadriamo il problema nel framework della stima non parametrica e assumiamo che la funzione sia generata da un processo Gaussiano con covarianza nota. Sotto deboli ipotesi sul kernel del processo Gaussiano, viene proposta una soluzione che collega la classica regressione Gaussiana con il filtro di Kalman grazie all'utilizzo di una griglia su cui vengono prese le misure. Come risultato principale proponiamo un algoritmo efficiente per stimare funzioni tempo e spazio varianti e che combina i vantaggi della regressione Gaussiana, e.g. l'assenza di modelli, con quelle del filtro di Kalman, e.g. l'efficienza.

Abstract

Since the eighties localization and mapping problems have attracted the efforts of robotics researchers. However in the last decade, thanks to the increasing capabilities of the new electronic devices, many new related challenges have been posed, such as swarm robotics, aerial vehicles, autonomous cars and robotics networks. Efficiency, robustness and scalability play a key role in these scenarios.
Efficiency is intended as an ability for an application to minimize the resources usage, in particular CPU time and memory space. In the aforementioned applications an underlying communication network is required so, for *robustness* we mean asynchronous algorithms resilient to delays and packet-losses. Finally *scalability* is the ability of an application to continue functioning without any dramatic performance degradation even if the number of devices involved keep increasing.
In this thesis the interest is focused on parametric and non-parametric estimation algorithms applied to localization and mapping in robotics. The main contribution can be summarized in the following four arguments:

(i) **Consensus-based localization** We address the problem of optimal estimating the position of each agent in a network from relative noisy vectorial distances with its neighbors by means of only local communication and bounded complexity, independent of network size and topology. In particular we propose a consensus-based algorithm with the use of local memory variables which allows asynchronous implementation, has guaranteed exponential convergence to the optimal solution under simple deterministic and randomized communication protocols, and requires minimal packet transmission. In the randomized scenario, we then study the rate of convergence in expectation of the estimation error and we argue that it can be used to obtain upper and lower bound for the rate of converge in mean square. In particular, we show that for regular graphs, such as Cayley, Ramanujan, and complete graphs, the convergence rate in expectation has the same asymptotic degradation of memoryless asynchronous consensus algorithms in terms of network size. In addition, we show that the asynchronous implementation is also robust to delays and communication failures. We finally complement the analytical results with some numerical simulations, comparing the proposed strategy with other algorithms which have been recently proposed in the literature.

(ii) **Aerial Vehicles distributed localization:** We study the problem of distributed multi-agent localization in presence of heterogeneous measurements and wireless communication. The proposed algorithm integrates low precision global sensors, like GPS and compasses, with more precise relative position (i.e., range plus bearing) sensors. Global sensors are used to reconstruct the absolute position and orientation, while relative sensors are used to retrieve the shape of the formation. A fast distributed and asynchronous linear least-squares algorithm is proposed to solve an approximated version of the non-linear Maximum Likelihood problem. The algorithm is provably shown to be robust to communication losses

and random delays. The use of ACK-less broadcast-based communication protocols ensures an efficient and easy implementation in real world scenarios. If the relative measurement errors are sufficiently small, we show that the algorithm attains a solution which is very close to the maximum likelihood solution. The theoretical findings and the algorithm performances are extensively tested by means of Monte-Carlo simulations.

(iii) **Estimation and Coverage:** We address the problem of optimal coverage of a region via multiple robots when the sensory field used to approximate the density of event appearance is not known in advance. We address this problem in the context of a client-server architecture in which the mobile robots can communicate with a base station via a possibly unreliable wireless network subject to packet losses. Based on Gaussian regression which allows to estimate the true sensory field with any arbitrary accuracy, we propose a randomised strategy in which the robots and the base station simultaneously estimate the true sensory distribution by collecting measurements and compute the corresponding optimal Voronoi partitions. This strategy is designed to promote exploration at the beginning and then smoothly transition to station the robots at the centroid of the estimated optimal Voronoi partitions. Under mild assumptions on the transmission failure probability, we prove that the proposed strategy guarantees the convergence of the estimated sensory field to the true field and that the corresponding Voronoi partitions asymptotically becomes arbitrarily close to an optimal Voronoi partition. Additionally, we also provide numerically efficient approximation that trade-off accuracy of the estimated map for reduced memory and CPU complexity. Finally, we provide a set of extensive simulations which confirm the effectiveness of the proposed approach.

(iv) **Non-parametric estimation of spatio-temporal fields:** We address the problem of efficiently and optimally estimating an unknown time-varying function through the collection of noisy measurements. We cast our problem in the framework of non-parametric estimation and we assume that the unknown function is generated by a Gaussian process with a known covariance. Under mild assumptions on the kernel function, we propose a solution which links the standard Gaussian regression to the Kalman filtering thanks to the exploitation of a grid where measurements collection and estimation take place. This work show an efficient in time and space method to estimate time-varying function, which combine the advantages of the Gaussian regression, e.g. model-less, and of the Kalman filter, e.g. efficiency.

1 Introduction

One of the first and biggest challenges that the roboticists tackled more than thirty years ago were the localization and mapping problems. Along the years these two topics created a new robotics field known as SLAM (Simultaneous Localization And Mapping). SLAM is a technique that allows robots to simultaneously create a map of the environment, and localize themselves on that map, in the presence of both measurement and movement noise. These two problems appear together because they are strictly connected, in fact, the knowledge of the position of a robot is of primary necessity to build a map, but at the same time the knowledge of the map is useful to improve the localization estimate.

During the eighties and nineties the main limitations were not coming from the lack of algorithms to solve the SLAM problem but from the hardware and software technological constraints. However in the last decade, thanks to the proliferation of relatively inexpensive devices capable of communicating, computing, sensing, interacting with the environment and storing information is promising an unprecedented number of novel applications throughout the cooperation of these devices toward a common goal. A direct consequence of these advancements is the extension of the SLAM problem into a multi-agent framework.

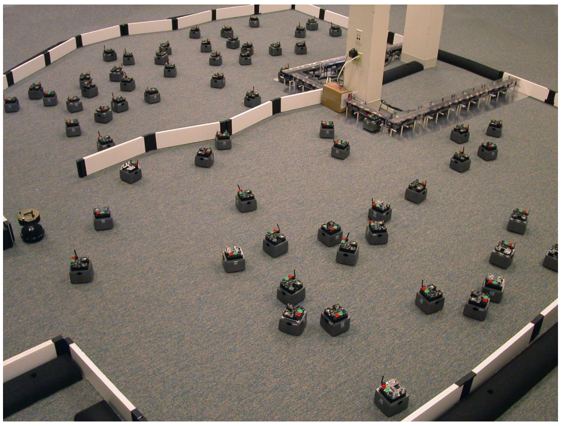


Figure 1.1: Example of a robotics swarm. *Credits: James McLurkin.*

This new scenario also pose new challenges, of which *efficiency, robustness* and *scalability* are the major ones. In computer science algorithm *efficiency* has been studied since the eighteenth century. The mathematician and pioneer of informatics *Ada Lovelace* in 1843 highlighted this aspect:

«In almost every computation a great variety of arrangements for the succession of the processes is possible, and various considerations must influence the selections amongst them for the purposes of a calculating engine. One essential object is to choose that arrangement which shall tend to reduce to a minimum the time necessary for completing the calculation.»

In other words algorithmic efficiency are the properties of an algorithm which relate to the amount of computational resources used by the algorithm. In particular is fundamental to minimize th CPU time and the memory space used. The concept of *robustness* is a cross-referring topic and a definition associated to the systems is the following:

«Robustness is the ability of tolerating perturbations that might affect the system's functional body. In the same line it can be defined as the ability of a system to resist change without adapting its initial stable configuration. »

In these thesis the meaning of robustness is twofold, first is the intrinsic ability of a multi-agent system to accomplish a task even if one or more agents exhibit a failure, second is the capability of the system to overcome issues in communications like packet losses and delays. These two kind of robustness are typical in robotics and are extensively studied in the literature. Finally *scalability* is intended as the ability for an application to continue functioning without any dramatic performance degradation even if the number of devices involved keeps increasing. In particular, an application is scalable if it is not necessary to increase hardware resources nor to adopt more complex software algorithms in each device even if the total number of devices increases.

This thesis is divided in two main parts, one about localization and one about mapping. The first and the second chapters address the problem of designing algorithms that are capable to reconstruct the optimal estimate of the location of a device based on noisy absolute measurements and noisy relative measurements with respect to its neighbors in a connected network. Then it is shown an application to multi-vehicle localization where the estimate has to be performed in real-time and communication is achieved via wireless communication. The goal is to integrate less precise global sensors (GPS and compass) with more precise relative positioning sensors (range and bearing sensors) in order to achieve global high accuracy. Intuitively, precise range and bearing sensors would allow for the reconstruction of a relative formation but provides no information about the global position and orientation of the formation. Differently, compass and GPS installed in multiple vehicles can provide estimation of the centroid and orientation of the whole formation. The fusion of these two types of information would allow an accurate global positioning of all vehicles. Another challenge that we want to address is to provide an algorithm that is totally distributed, asynchronous and robust to communication losses. In fact, a centralized solution is not advisable in a scenario where not all vehicles can communicate with each other and a complex leader-election procedure might be needed. Moreover, synchronous communication is also difficult to enforce since it requires fine time synchronization among the different vehicles and possible packet losses might slow down the algorithm since multiple retransmissions are required to deliver the message.

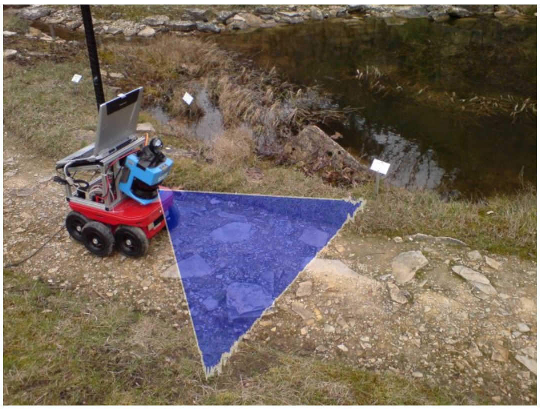


Figure 1.2: Robots involved in a localization-exploration process. *Credits: Universität Osnabrück*

In the second part we will focus on the mapping problem using a non-parametric approach applied to coverage and partitioning problems. In this context, the coverage and partitioning of an area of interest is one important and interesting task. In application like surveillance, pollution monitoring and rescue, just to mention some of them, the ability of a group of robots to sense and automatically cover the surrounding environment in order to maximize the likelihood of detecting an event of interest is appealing. On the other hand, knowledge about the spatial distribution of the event of interest is needed. If this is uniform in space, the optimal partitioning will correspond to a uniform positioning of the robots in the monitored area. More challenging is the case of non-uniform sensory functions. Intuitively, the robots should cover more densely regions where the function is high. Regions where the function is low will be less covered. In this thesis we analyze the framework of coverage the area of interest while estimating the non-uniform measurable field of event appearance from noisy measurements collected by the robots. As an academic example, suppose that we have a number of robots that have to monitor a forest for detecting possible wildfires. It is reasonable to assume that the probability of a wildfire is proportional to the temperature in a certain location, therefore robots should more densely cover areas where the temperature is higher. If the temperature is not known in advance, the robots have to move around to collect temperature samples in order to reconstruct the temperature profile used to partition the environment. However, in order to minimize the time to reach a wildfire, the robots should station near the centroids of these partitions. This simple example clearly exhibits the classical problem associated with the exploration-exploitation dilemma.

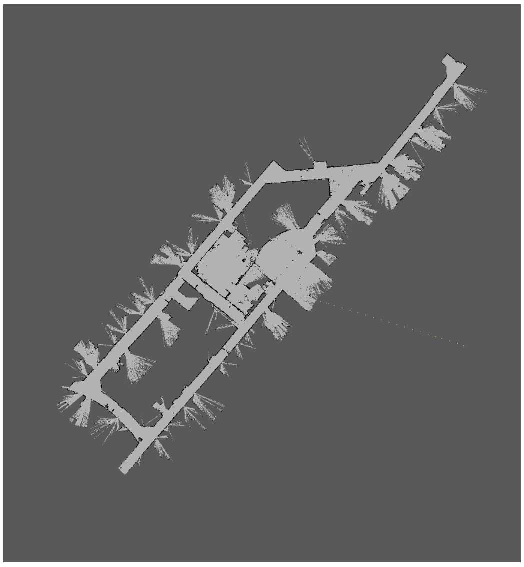


Figure 1.3: Example of a reconstructed map. *Credits: CAR, Components, Agents, Robots.*

If we also assume that the field to estimate changes over time, it is necessary to study an algorithm which is capable to learn the dynamics of the spatio-temporal filed in an efficient way. This is relevant in the wildfire detection problem but also for example to monitor the temperature in an area of the sea or the level of radiation in a region close by a nuclear accident. Again here the keyword is efficient because the statistical inverse problem has a computational complexity which grows with the cube of the number of data, i.e. $O(N^3)$, which is unfeasible for any kind of real-time constraint. On the other hand, the popularity of the state space models is due to the fact that the inference problem can be solved with a linear time complexity, i.e. $O(T)$. The holy Grail for the estimation community is to find an algorithm which combine the best of the two approaches, e.g. the reduced computational complexity of algorithm as the Kalman filter and the advantage of the model-less approaches as the Gaussian Regression, or at least try to do that for a wider class of models.

Literature Review Localization: before stating the novelties introduced in this thesis, we briefly review a list of works related to our framework. Distributed optimization is the perfect framework for solving the localization problem since in the past years many problems in large scale network have been cast as convex optimization problems. In particular, the localization problem can be cast as the following unconstrained optimization problem:

$$\min_{x_1,\ldots,x_N} \sum_{(i,j)\in\mathcal{E}}^{|\mathcal{E}|} f_{ij}(x_i - x_j) \tag{1.1}$$

where $x_i \in \mathbb{R}^\ell$, $\mathcal{E}$ represents all the pair of nodes for which are available relative measurements and f_{ij} are convex functions. Many problems can be written in this framework such as sensor localization [1, 2], sensor calibration [3], clock synchronization [4] and camera localization [5, 6]. For example, in the context of localization from vectorial relative distances in a plane, the cost functions f_{ij} are given by:

$$f_{ij}(x_i - x_j) = \|x_i - x_j - z_{ij}\|^2$$

where $z_{ij} \in \mathbb{R}^\ell$ is the noisy measurement of the relative (vector) distance of node i from node j. As a consequence, the optimization problem in Eqn. (1.1) becomes a distributed least-square problem. Several scalable distributed solutions to this problem are already available in the literature. In [1, 2] the authors propose a distributed Jacobi solution based on a synchronous implementation, which was later extended to account for asynchronous communication and packet losses [1]. The same approach has been independently proposed in [7] in the context of distributed time synchronization in wireless sensor networks. Differently, in [3] a broadcast consensus-based algorithm, which is suitable for asynchronous implementation, is proposed but the local estimates do not converge and exhibit an oscillatory behaviour around the optimal value. A similar approach has been proposed in [8, 9] where the local ergodic average of the gossip asynchronous algorithm is proved to converge to the optimal value as $1/k$, where k is the number of iterations. An alternative approach based on the Kaczmarz method for the solution of general linear systems has been suggested in [10], however a practical asynchronous implementation for distributed localization from relative measurements which satisfies the specific edge and node activation probabilities dictated by the algorithm, is not given, moreover, no robustness analysis in terms of delays is provided.

Literature Review Coverage: another interesting research field is given by the coverage and partitioning problems. In the classical coverage literature, the sensory function is supposed to be known by the robots. In this spirit, [11, 12] present a gradient descent strategy for a class of functions which encode optimal coverage policies. The authors exploit the concept of centroidal Voronoi partitions to optimally divide the monitored area. In [13] the authors consider the coverage problem on a line with non-uniform sensory function which is assumed to be perfectly known and propose a strategy which provides optimal coverage. In the work of [14] only a limited number of noise-free samples of the sensory function are considered. This is practically and computationally more appealing but results in only a sub-optimal solution without any proof of convergence. To address this problem, in [15] the authors presented a bridging step, where cartograms have been exploited to map a non-uniform distribution onto a uniform distribution which can be used to apply the strategy proposed in [12].

Some results to the coupled problem, i.e. when both coverage and estimation are considered, have appeared recently. In [16] the estimation from noisy measurements of a time varying spatial distribution function is considered. The coverage is performed thanks to a gradient ascent algorithm which lets the robots move on the maxima of the sensory function. However, proof of convergence is guaranteed only in the case of noise-free measurements. In [17] the authors analyze the problem of non parametric estimation of Gaussian processes using Kalman filtering and coverage. The algorithm proposed consists of two different phases: during the first, based on information on the posterior variance, the robots are spread throughout the space in order to achieve a good estimate of the sensory function; when the maximum of the posterior is below a certain threshold, the robots switch from exploration to the exploitation phase to achieve coverage. A notable advancement is represented by the work of [18] which propose a distributed consensus-like parametric estimation from noise-free measurements from which the agent are able to reconstruct the sensory function distribution. While estimating, the robots move towards the centroids of the Voronoi partition in order to perform optimal coverage of the area of interest. In [19] the authors extend the work of [13, 14] by analyzing the problem of non-uniform line partitioning from a non-Euclidean perspective. Differently from the previous work, only noisy measurements of the sensory function are exploited. By letting the robots to collect a certain number of noisy samples from the neighbourhood of their positions, the authors proves convergence in probability to the optimal configuration. More recently [20] used the concept of stochastic gradient based on noisy measurements of an unknown sensory function to perform an adaptive deployment of a group of robots. This approach has the interesting feature that the distribution function is not needed to be estimated.

Literature Review Estimation: concerning the spatio-temporal field estimation many techniques has been proposed in the literature. The subspace identification approach [21, 22] has been successfully applied in identification of linear time invariant (LTI) systems, however during the nighties these three approaches have been presented to solve the linear time variant (LTV) problem. The first is the canonical variate analysis (CVA) [23], the second the multi-variable output-error state space (MOESP) [24] and the last the numerical algorithms for subspace state space identification (N4SID) [25]. A unified framework for this three algorithm was proposed in [26]. If we consider the non-parametric or the Gaussian regression approaches many works are based on the study of estimation of Gaussian random field varying in time and space [27]. As we mentioned before, the main issue of the inverse problem is the high computational complexity, so many solutions has been proposed, as the reduction to into SDE form [28], filtering and smoothing [29], recursive Bayesian methods [30], kernel recursive least-square [31] and sparse approximation [32]. Another kind of approach is called Kriging, which essentially is the estimation of a time variant spatial field. One of the best known solution is the Kriged Kalman Filter (KKF) [33], presented also in a distributed fashion in [34]. The algorithm proposed in this thesis is similar to the one proposed in [35] but we extend the class of processes that can be estimated to a wider one.

Statement of contribution: In the first part of the thesis we will focus on a distributed optimization problem with application in localization of a group of robots given noisy relative measurements. The main contribution of this chapter is to propose a novel asynchronous algorithm whose main idea consists in casting the estimation problem as a consensus problem under some suitable changes of coordinates, and then to add some extra memory variables at each node to keep track of the estimated location of its neighbors, i.e. the nodes from which they collected the relative distance measurements. Estimates of these local variables eventually converge to the estimates of the neighbors, thus guaranteeing the convergence of the whole algorithm, at the price of some delay. This strategy has several relevant advantages, namely:

i) is scalable,

ii) has proven exponential rate of convergence under mild assumptions,

iii) is robust to packet losses and delays,

iv) requires the transmission of a single broadcast communication packet per each iteration.

This last feature is particularly relevant for Wireless Sensor Networks (WSNs) applications since agents have a limited energy budget and communication is more expensive than computation from an energy standpoint. We also study the performance of the proposed algorithm in terms of the convergence rate. This task is particularly challenging since the proposed algorithm turns out to be a higher order consensus algorithm, for which analytic tools are available. In fact, the few works available in the literature, which address the rate of convergence of randomized higher order consensus algorithms, are limited to the convergence in expectation [36]. We exactly compute the rate of convergence in expectation of our algorithm for regular graphs, and through extensive numerical simulations we conjecture that it also provides an upper bound for rate of convergence in mean square. Moreover, we show that, asymptotically, for many types of regular graphs such as Caylays, Ramanujan and complete graphs, such rate of convergence in expectation is reduced by a factor N, where N is the number of nodes, which is the same of standard memoryless asynchronous consensus algorithms, thus implying that asymptotically in N the reduction of rate of convergence due to memory is negligible. We also prove the convergence of the proposed algorithm when bounded delays and packet losses are present, thus making it particularly suitable for applications using lossy wireless communication. We finally complement the theoretical results with some numerical simulations which show that the proposed algorithm has a performance in terms of rate of convergence per iteration which is slightly slower than the fastest algorithms available in the literature. However, it greatly outperforms them if the rate of convergence is computed in terms of number of exchanged messages, i.e, the estimation error obtained by sending a fixed number of packets is much lower for our proposed algorithm than the other algorithms available in the literature.

Then, in the second chapter, we propose an asynchronous distributed algorithm for multi-robot localization that integrates GPS, compass, range and bearing measurements and is robust to packet losses and random delays. In particular, we show that, if the range and bearing errors are sufficiently small, it is possible to linearize the localization problem achieving a performance which is very close to the exact maximum likelihood solution. Moreover, such solution can be computed via a broadcast-based communication protocols that does not require ACK packets and is therefore fast and easy to implement.

The third chapter of this thesis describes how to map an unknown density function exploiting an efficient non-parametric approach. This density function is the used to perform a coverage control task. In particular we present a strategy that is suitable for a client-server communication architecture in which the robots can communicate with a base station in order to perform non-parametric estimation of an unknown sensory distribution function from noisy samples while performing optimal coverage of the area of interest. We consider a realistic scenario where communication is performed over an unreliable wireless network subject to packet losses or delays. The contribution is fourfold:

i) differently from [17] the transition from the exploration and the exploitation phases is seamless being based on a randomized control law,

ii) almost-sure convergence to the true sensory function, from a collection of noisy measurements, is guaranteed under mild assumptions on the packet loss probability,

iii) the final configuration of the robots can be arbitrarily close to an optimal partitioning that would be obtained if the sensory function would be known since the beginning,

iv) we study the computational complexity of the algorithm and we additionally propose an alternative approximated grid-based algorithm that can trade-offs accuracy on the estimated map for considerable reduction on CPU and memory requirements.

Finally, in the last chapter, we present an unified framework for Gaussian Regression and Kalman filtering for optimal estimation of a certain class of processes. The main idea behind this results is to use a spatial-grid in order to limit the space where the data can be collected. This transform the infinite dimensional Gaussian regression inverse problem to a finite dimensional one and the computational complexity is fixed and given by the number of points in the grid. The contribution is twofold:

i) we present an efficient and optimal grid based estimation for a wide class of Gaussian processes through a Kalman filter,

ii) an approximated and efficient grid based estimation for all the processes which are not included in the previous class,

Outline: this thesis is divided as follows. The first part is composed by Chapter 2 and 3, and deals with localization problems. More precisely, in Chapter 2 we consider the localization of a group of agents exploiting only noisy relative measurements, while Chapter 3 extend the problem to a more general framework, where both relative and absolute measurements are available. The second part of the thesis, composed by Chapter 4 and 5, deals with the estimation of density functions. In particular in Chapter 4 we study an efficient algorithm to estimate a density function to perform coverage control, while Chapter 5 we propose an unified framework for Gaussian Regression and Kalman filtering for optimal estimation.

Publications: Part of the content of this thesis has already been published in international conferences and journals. In particular the results of Chapter 2 comes from [37] and [38], while the original idea behind the algorithm presented in Chapter 4 has been published in [39]. Finally part of the results obtained in chapter 3 and 5 has been sumbitted and has to be subimtted to international journals.

2 Consensus Based Localization

In this chapter we address the problem of designing algorithms that are capable of reconstructing the optimal estimate of the location of a device based on noisy relative measurements with respect to its neighbors in a connected network. In particular, we want to design distributed algorithms that allow each device to reconstruct its own position only from exchanging information with its neighbors, regardless of the size of the network. First we propose a synchronous distributed consensus based algorithm and we prove to be exponential convergent to an optimal solution of the localization optimization problem presented. The main limitation of this approach it is given by the fact that is applicable only to sensor networks with synchronized and reliable communication. This means that in a real scenario the presented synchronous algorithm may not converge to the optimal solution or may not converge at all. The aim of this chapter is to present an alternative solution capable of solving this limitation. For this reason is presented an asynchronous distributed consensus algorithm which even has guarantee convergence under mild assumption. Then we perform a performance analysis with a dedicated section for regular graphs. Finally we prove the effectiveness of the algorithm also in presence of nonidealities of the communication media as random delays and packet losses. The chapter is organized as follows. In Section 2.2 we formulate the problem. In section 2.1 we introduce some basic notation and we review some useful concepts. In Section 2.3 we introduce the synchronous consensus-based algorithm (denoted as s-CL). In Section 2.4 we propose a more realistic asynchronous implementation of the s-CL algorithm (denoted as a-CL). In Section 2.5 we establish the convergence of the a-CL algorithm and we provide some bounds on the rate of convergence in mean-square. In Section 2.6 we show that the a-CL algorithm is robust to delays and communication failures. In Section 2.7 we provide some numerical results comparing the a-CL algorithm to other strategies recently proposed in the literature.

2.1 Mathematical preliminaries

Before proceeding, we collect some useful definitions and notations. In this paper, $\mathcal{G} = (V, \mathcal{E})$ denotes a directed graph where $V = \{1, ..., N\}$ is the set of vertices and $\mathcal{E}$ is the set of directed edges, i.e., a subset of $V \times V$. More precisely the edge (i, j) is incident on node i and node j and is assumed to be directed away from i and directed toward j. The graph $\mathcal{G}$ is said to be bidirected if $(i, j) \in \mathcal{E}$ implies $(j, i) \in \mathcal{E}$.

Given a directed graph $\mathcal{G} = (V, \mathcal{E})$, a directed path in $\mathcal{G}$ consists of a sequence of vertices $(i_1, i_2, \ldots, i_r)$ such that $(i_j, i_j + 1) \in \mathcal{E}$ for every $j \in \{1, \ldots, r-1\}$. An undirected path in $\mathcal{G}$ consists of a sequence of vertices $(i_1, i_2, \ldots, i_r)$ such that either $(i_j, i_{j+1}) \in \mathcal{E}$ or $(i_{j+1}, i_j) \in \mathcal{E}$ for every $j \in \{1, \ldots, r-1\}$[1]. The directed graph $\mathcal{G}$ is said to be *strongly connected* (resp. *weakly connected*) if for any pair of vertices (i, j) there exists a directed path (resp. undirected path) connecting i to j. Given the directed graph $\mathcal{G}$, the set of neighbors of node i, denoted by $\mathcal{N}_i$, is

[1] Basically, an *undirected path* is a path from a node to another node that does not respect the orientation of the edges.

given by $\mathcal{N}_i = \{j \in V \mid (i,j) \in \mathcal{E}\}$. A directed graph is said to be *regular* if all the nodes have the same number of neighbors.

Given a directed graph $\mathcal{G} = (V, \mathcal{E})$ with $|\mathcal{E}| = M$ let the *incidence matrix* $A \in \mathbb{R}^{M \times N}$ of $\mathcal{G}$ be defined as $A = [a_{ei}]$, where $a_{ei} = 1, -1, 0$, if edge e is incident on node i and directed away from it, is incident on node i and directed toward it, or is not incident on node i, respectively.

Let $\mathbf{1}_N$ be the N-dimensional column vector with all components equal to one. If there is no risk of confusion we will drop the subscript N. Given a matrix B we denote with $B^\dagger$ its pseudo-inverse. Given a vector v with v^T we denote its transpose. A matrix $P \in \mathbb{R}^{N \times N}$ is said to be stochastic if all its elements are nonnegative and $P\mathbf{1} = \mathbf{1}$. Moreover it is said to be doubly stochastic if it is stochastic and, additionally, $\mathbf{1}^T P = \mathbf{1}^T$. A stochastic matrix P is primitive if it has only one eigenvalue equal to 1 and all other eigenvalues are strictly inside the unitary circle. With the symbol $\rho_{ess}(P)$ we denote the essential spectral radius of P (see [40]), namely, the second largest eigenvalue of P in absolute value.

In graph theory, a regular graph is a graph where each vertex has the same number of neighbors; a regular graph with vertices of degree d is called a d-regular graph or regular graph of degree d. A Ramanujan graph is a regular graph whose spectral gap is almost as large as possible. Let $\mathcal{G}$ be a connected d-regular graph with N vertices, and let $\lambda_0 \geq \lambda_1 \geq \ldots \geq \lambda_{n-1}$ be the eigenvalues of the adjacency matrix of $\mathcal{G}$. A d-regular graph $\mathcal{G}$ is a Ramanujan graph if

$$(\max_i \lambda_i) \leq 2\sqrt{d-1}.$$

The symbol $\mathbb{E}$ denotes the expectation operator. Given two functions $f, g : \mathbb{N} \longmapsto \mathbb{R}$, we say that $f \in o(g)$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$.

2.2 Problem Formulation

The problem we deal with is that of estimating N variables $x_1, \ldots, x_N$ from noisy measurements of the form

$$z_{ij} := x_i - x_j + n_{ij}, \qquad i, j \in \{1, \ldots, N\}, \tag{2.1}$$

where n_{ij} is zero-mean measurement noise. Although all results in this work apply to general vector-valued variables, for sake of simplicity, in this paper we assume that $x_i \in \mathbb{R}$, $i \in \{1, \ldots, N\}$. This estimation problem can be naturally associated with a *measurement graph* $\mathcal{G}_m = (V; \mathcal{E}_m)$. The vertex set V of the measurement graph consists of the set of nodes $V = \{1, \ldots, N\}$ where N is the number of nodes, while its edge set $\mathcal{E}_m$ consists of all of the ordered pairs of nodes (i, j) such that a noisy measurement of the form (2.1) between i and j is available to node i. The measurement errors on distinct edges are assumed uncorrelated. The measurement graph $\mathcal{G}_m$ is a directed graph since $(i, j) \in \mathcal{E}_m$ implies the measurement z_{ij} is available to node i, while $(j, i) \in \mathcal{E}_m$ implies the measurement z_{ji} is available to node j, and these two are in general distinct.

Next we formally state the problem we aim at solving. Let $\mathbf{x} \in \mathbb{R}^N$ be the vector obtained by stacking together all the variables $x_1, \ldots, x_N$, i.e., $\mathbf{x} = [x_1, \ldots, x_N]^T$, and let $\mathbf{z} \in \mathbb{R}^M$ and $\mathbf{n} \in \mathbb{R}^M$, where $M = |\mathcal{E}_m|$, be the vectors obtained stacking together all the measurements z_{ij} and the noises n_{ij}, respectively. Additionally, let $R_{ij} > 0$ denote the covariance of the zero mean error n_{ij}, i..e, $R_{ij} = \mathbb{E}[n_{ij}^2]$, and let $R \in \mathbb{R}^{M \times M}$ be the diagonal matrix collecting in its diagonal the covariances of the noises n_{ij}, $(i, j) \in \mathcal{E}$, i.e., $R = \mathbb{E}[\mathbf{n}\mathbf{n}^\mathbf{T}]$. Observe that Eqn. (2.1) can be rewritten in a vector form as

$$\mathbf{z} = A\mathbf{x} + \mathbf{n}$$

Now, define the set

$$\chi := \operatorname*{argmin}_{\mathbf{x} \in \mathbb{R}^N} (\mathbf{z} - A\mathbf{x})^T R^{-1} (\mathbf{z} - A\mathbf{x}).$$

The goal is to construct an optimal estimate $\mathbf{x}_{\text{opt}}$ of $\mathbf{x}$ in a least square sense, namely, to compute

$$\mathbf{x}_{\text{opt}} \in \chi \tag{2.2}$$

Assume the measurement graph $\mathcal{G}_m$ to be *weakly connected*, then it is well known (see [2]) that

$$\chi = \left\{ \left(A^T R^{-1} A\right)^{\dagger} A^T R^{-1} \mathbf{z} + \alpha \mathbf{1} \right\}.$$

Moreover let

$$\mathbf{x}^*_{\text{opt}} = \left(A^T R^{-1} A\right)^{\dagger} A^T R^{-1} \mathbf{z},$$

then $\mathbf{x}^*_{\text{opt}}$ is the minimum norm solution of (2.2), i.e.,

$$\mathbf{x}^*_{\text{opt}} = \min_{\mathbf{x}_{\text{opt}} \in \chi} \parallel \mathbf{x}_{\text{opt}} \parallel$$

The matrix $A^T R^{-1} A$ is called in literature the *Weighted Generalized Grounded Laplacian* [2].

Remark 2.1 Observe that, just with relative measurements, determining the x_i's is only possible up to an additive constant. This ambiguity might be avoided by assuming that a node (say node 1) is used as reference node, i.e., $x_1 = 0$.

2.3 A synchronous distributed consensus based solution

To compute an optimal estimate $\mathbf{x}_{\text{opt}}$ directly, one needs all the measurements and their covariances $(\mathbf{z}, R)$, and the topology of the measurement graph $\mathcal{G}_m$. In this section the goal is to compute the optimal solution in a distributed fashion, employing only local communications. In particular we assume that a node i and another node j can communicate with each other if either $(i, j) \in \mathcal{E}_m$ or $(j, i) \in \mathcal{E}_m$. Accordingly, we introduce the *communication graph* $\mathcal{G}_c(V, \mathcal{E}_c)$, where $(i, j) \in \mathcal{E}_c$ if either $(i, j) \in \mathcal{E}_m$ or $(j, i) \in \mathcal{E}_m$. Observe that, if $(i, j) \in \mathcal{E}_c$ then also $(j, i) \in \mathcal{E}_c$, namely, $\mathcal{G}_c$ is a bidirected graph. From now on, $\mathcal{N}_i$ denotes the set of neighbors of node i in the communication graph $\mathcal{G}_c(V, \mathcal{E}_c)$.

In what follows we introduce a distributed solution which is based on standard linear consensus algorithm. A discussion of the linear consensus algorithm can be found in the review papers [41, 42], a brief overview can be found in Appendix A. Instead we make the presentation of the algorithm self-contained. Firstly, we assume that the communications among the nodes are synchronous, namely, all nodes perform their transmissions and updates at the same instant, and design the algorithm for this scenario. We refer to this algorithm as the *synchronous consensus-based localization* algorithm (denoted hereafter as s-CL algorithm). In section 2.4 we will modify the s-CL algorithm to make it suitable to asynchronous communications. We assume that before running the s-CL algorithm, the nodes exchange with their neighbors their relative measurements as well as the associated covariances. So every node has access to the measurements on the edges that are incident to it, whether the edge is directed to or away from it. Each node uses the measurements obtained initially for all future computations. The s-CL algorithm is formally described as follows.

Algorithm 1 s-CL

Processor states: For $i \in \{1, \ldots, N\}$, node i stores in memory the measurements $\{z_{ij}, (i,j) \in \mathcal{E}_m\}$, and $\{z_{ji}, (j,i) \in \mathcal{E}_m\}$, and the associated covariances $\{R_{ij}, (i,j) \in \mathcal{E}_m\}$ and $\{R_{ji}, (j,i) \in \mathcal{E}_m\}$. Moreover node i stores in memory an estimate $\hat{x}_i$ of x_i.

Initialization: For $i \in \{1, \ldots, N\}$, node i initializes its estimate $\hat{x}_i(0)$ to any arbitrary value.

Transmission iteration: For $k \in \mathbb{N}$, at the start of the $(k+1)$- th iteration of the algorithm, node i transmits its estimate $\hat{x}_i(k)$ to all its neighbors. It also gathers the k-th estimates of its neighbors, $\hat{x}_j(k)$, $j \in \mathcal{N}_i$.

Update iteration: For $k \in \mathbb{N}$, node i, $i \in \{1, \ldots, N\}$, based on the information received from its neighbors, updates its estimate as follows

$$\hat{x}_i(k+1) := p_{ii}\, \hat{x}_i(k) + \sum_{j \in \mathcal{N}_i} p_{ij}\, \hat{x}_j(k) + b_i$$

where

$$b_i = \epsilon \sum_{(i,j) \in \mathcal{E}_m} R_{ij}^{-1} z_{ij} - \epsilon \sum_{(j,i) \in \mathcal{E}_m} R_{ji}^{-1} z_{ji}$$

and where

$$p_{ij} = \begin{cases} \epsilon(R_{ij}^{-1} + R_{ji}^{-1}) & \text{if } (i,j) \in \mathcal{E}_m \text{ and } (j,i) \in \mathcal{E}_m \\ \epsilon R_{ij}^{-1} & \text{if } (i,j) \in \mathcal{E}_m \text{ and } (j,i) \notin \mathcal{E}_m \\ \epsilon R_{ji}^{-1} & \text{if } (j,i) \in \mathcal{E}_m \text{ and } (i,j) \notin \mathcal{E}_m \end{cases}$$

and

$$p_{ii} = 1 - \sum_{j \in \mathcal{N}_i} p_{ij}$$

being ϵ a positive constant *a-priori* assigned to the nodes.

Now, let $P \in R^{N \times N}$ be the matrix defined by the weights p_{ij} above introduced. One can see that such matrix P is equal to

$$P = I - \epsilon A^T R^{-1} A.$$

Moreover let

$$b = \epsilon A^T R^{-1} \mathbf{z},$$

and let $\hat{\mathbf{x}}(k) = [\hat{x}_1(k), \ldots, \hat{x}_N(k)]^T$ .. Then the s-CL algorithm can be written in a compact form as

$$\hat{\mathbf{x}}(k+1) = P\hat{\mathbf{x}}(k) + b$$

To characterize the convergence properties of the s-CL algorithm, we next introduce two definitions and a crucial property of the matrix P. First, let $d_{max} = \max\{|\mathcal{N}_i|, i \in \{1, \ldots, N\}\}$. Second, let $R_{min} = \min\{R_{ij}, (i,j) \in \mathcal{E}_m\}$. Observe that, if $0 < \epsilon < 1/(2d_{\max} R_{min}^{-1})$, then the matrix P is stochastic. If in addition, the measurement graph $\mathcal{G}_m$ is weakly connected, and consequently if communication graph $\mathcal{G}_c$ is strongly connected, then the matrix P is primitive. Under these assumptions, we have the following Proposition:

Proposition 2.2 Consider the s-CL algorithm running over a *weakly connected* measurement graph $\mathcal{G}_m$. Let ϵ be such that $0 < \epsilon < 1/(2d_{\max} R_{min}^{-1})$. Moreover let $\hat{x}_i$, $i \in \{1, \ldots, N\}$, be initialized to any real number. Then the following two facts hold true

(i) the evolution $k \to \hat{\mathbf{x}}(k)$ asymptotically converges to an optimal estimate $\mathbf{x}_{\text{opt}} \in \chi$, i.e., there exists $\alpha \in \mathbb{R}$, such that

$$\lim_{k \to \infty} \hat{\mathbf{x}}(k) = \mathbf{x}^*_{\text{opt}} + \alpha \mathbf{1};$$

where α linearly depends on $\hat{\mathbf{x}}(0)$.

(ii) the convergence is exponential, namely, there exists $C > 0, \rho_{ess} < 1$ such that

$$\|\hat{\mathbf{x}}(k) - \big(\mathbf{x}^*_{\text{opt}} + \alpha\mathbf{1}\big)\| \leq C\rho^k_{ess}(P)\,\|\hat{\mathbf{x}}(0) - \big(\mathbf{x}^*_{\text{opt}} + \alpha\mathbf{1}\big)\|.$$

Proof We start by proving item (i). Let us define the change of variable $\xi = \hat{\mathbf{x}} - \mathbf{x}^*_{\text{opt}}$. Since $\mathbf{x}^*_{\text{opt}} = P\mathbf{x}^*_{\text{opt}} + b$, it is possible to write

$$\begin{aligned}\hat{\mathbf{x}}(k+1) - x^*_{\text{opt}} &= P\hat{\mathbf{x}}(k) + b - \mathbf{x}^*_{\text{opt}}\\ &= P\hat{\mathbf{x}}(k) + b - (P\mathbf{x}^*_{\text{opt}} + b)\\ &= P(\hat{\mathbf{x}}(k) - \mathbf{x}^*_{\text{opt}})\end{aligned}$$

and, in turn, $\xi(k+1) = P\xi(k)$. This equation describes the iteration of the classical consensus algorithm. Since P is a primitive doubly stochastic matrix, we have that

$$\xi(k) \to \frac{\mathbf{1}\mathbf{1}^T}{N}\xi(0)$$

where $\xi(0) = \hat{\mathbf{x}}(0) - \mathbf{x}^*_{\text{opt}}$. This implies that

$$\hat{\mathbf{x}}(k) \to \mathbf{x}^*_{\text{opt}} + \frac{\mathbf{1}\mathbf{1}^T}{N}\hat{\mathbf{x}}(0) - \frac{\mathbf{1}\mathbf{1}^T}{N}\mathbf{x}^*_{\text{opt}}$$

The fact that $\frac{\mathbf{1}\mathbf{1}^T}{N}\mathbf{x}^*_{\text{opt}} = 0$ concludes the proof of item (i).

Concerning item (ii) it is well known ([40]) that the convergence rate of a consensus algorithm ruled by a primitive matrix P, is exponential whose delay coefficient is given by the essential spectral radius $\rho_{ess}(P)$. ♢

Remark 2.3 The s-CL algorithm is similar to the algorithm proposed in [8]. However in [8], the measurement graph is assumed to be undirected, namely, both measurements z_{ij} and z_{ji} are available to node i and j under the additional assumption that $z_{ij} = -z_{ji}$.

Remark 2.4 The authors in [43] solved the problem formulated in (2.2) proposing a synchronous algorithm that implements the Jacobi iterative method. The performance of this algorithm, in terms of rate of convergence to the optimal solution, is similar, for many families of measurement graphs, to the performance of the synchronous consensus-based algorithm introduced in this section.

2.4 An asynchronous implementation of distributed consensus based solution

The distributed algorithm illustrated in the previous section, has an important limitation: it is applicable only to sensor networks with synchronized and reliable communication. Indeed, the s-CL algorithm requires that there exists a predetermined common communication schedule for all nodes and, at each communication round, each node must simultaneously and reliably communicate its information. The aim of this section is to reduce the communication requirements of the s-CL algorithm, in particular in terms of synchronization. To do so, we next introduce the *asynchronous Consensus-based Localization* algorithm (denoted as a-CL hereafter). This algorithm is based on an asymmetric broadcast communication protocol. Differently from the s-CL, at each iteration of the a-CL there is only one node transmitting information to all its neighbors. Since the actual value of neighboring estimates are not available at each iteration, we assume that each node stores in its local memory a copy of the neighbors' variables recorded from the last communication received. For $j \in \mathcal{N}_i$, we denote by $\hat{x}_j^{(i)}(k)$ the estimate of x_j kept in i's local memory at the end of the k-th iteration. If node j performed its last transmission to node i during h-th iteration, $h \leq k$, then $\hat{x}_j^{(i)}(k) = \hat{x}_j(h)$.

The a-CL algorithm is formally described in Algorithm 2.

Algorithm 2 a-CL

Processor states: For $i \in \{1, \ldots, N\}$, node i stores in memory the measurements z_{ij}, z_{ji} and the covariances R_{ij}, R_{ji} for all $j \in \mathcal{N}_i$. Moreover node i stores in memory also the estimate $\hat{x}_i$ of x_i and, for $j \in \mathcal{N}_i$ an estimate $\hat{x}_j^{(i)}$ of $\hat{x}_j$.

Initialization: Every node i initializes its estimate $\hat{x}_i$ and the variables $\hat{x}_j^{(i)}$, $j \in \mathcal{N}_i$, to arbitrary values.

Transmission iteration: For $k \in \mathbb{N}$, at the start of the $(k+\ 1)$-th iteration of the algorithm, there is only one node, say i, which transmits information to its neighbors; precisely, node i sends the value of its estimate $\hat{x}_i(k)$ to node j, $j \in \mathcal{N}_i$.

Update iteration: For $j \in \mathcal{N}_i$, node j performs the following actions in order

(i) it sets $\hat{x}_i^{(j)}(k+1) = \hat{x}_i(k)$, while for $s \in \mathcal{N}_j \setminus \{i\}$, $\hat{x}_s^{(j)}$ is left unchanged, i.e., $\hat{x}_s^{(j)}(k+1) = \hat{x}_s^{(j)}(k)$;

(ii) it updates $\hat{x}_j$ as

$$\hat{x}_j(k+1) := p_{jj}\hat{x}_j(k) + \sum_{h \in \mathcal{N}_j} p_{jh}\hat{x}_h^{(j)}(k+1) + b_j. \tag{2.3}$$

Clearly for $s \notin \mathcal{N}_i$, $\hat{x}_s$ is left unchanged during the $(k+1)$-th iteration of the algorithm, i.e, $\hat{x}_s(k+1) = \hat{x}_s(k)$.

Remark 2.5 Observe that the Algorithm 2 has been described assuming that the communication channels are reliable, i.e, no packet losses occur, and that the communication delays are negligible, i.e., when node i perform a transmission, the estimate $\hat{x}_i$ is instantaneously used by its neighbors. We will come back on these non-idealities in Section 2.6.

Next, we rewrite the updating step of the a-CL in a more compact way. Observe preliminarily that, under the assumption of reliable communications and by denoting with $\bar{k}$ the first iteration after which all nodes have transmitted at least once, then the estimate of node x_i stored in the neighbors of node i is always the same, i.e. for all $k \geq \bar{k}$ and $\ell, j \in \mathcal{N}_i$ we have $\hat{x}_i^{(\ell)}(k) = \hat{x}_i^{(j)}(k)$. Moreover, if we denote with $t_i'(k)$ the iteration during which node i has performed its last transmission up to iteration k of the a-CL (that is, $\hat{x}_i(t_i'(k))$ is the value of $\hat{x}_i$ at its last communication round), then for $j \in \mathcal{N}_i$, $\hat{x}_i^{(j)}(t'') = \hat{x}_i(t_i'(k))$ for all t'' such that $t_i'(k) < t'' \leq k$.

Now let us define $x_i'(k) = \hat{x}_i(k)$ and $x_i''(k) = \hat{x}_i(t_i'(k))$ and, accordingly, let $\mathbf{x}'(k) = [x_1'(k), \ldots, x_N'(k)]^T$ and $\mathbf{x}''(k) = [x_1''(k), \ldots, x_N''(k)]^T$. Moreover let $\sigma(k)$ denotes the node performing the transmission action at the beginning of the $k+1$-th iterationlet and let $Q_{\sigma(k)} \in \mathbb{R}^{2N\times 2N}$ be defined as

$$Q_{\sigma(k)} = \begin{bmatrix} Q_{11}^{(\sigma(k))} & Q_{12}^{(\sigma(k))} \\ Q_{21}^{(\sigma(k))} & Q_{22}^{(\sigma(k))} \end{bmatrix} \tag{2.4}$$

where

$$\begin{aligned} Q_{11}^{(\sigma(k))} &= \sum_{h\notin\mathcal{N}_{\sigma(k)}} e_h e_h^T + \sum_{j\in\mathcal{N}_{\sigma(k)}} \left(p_{jj}e_je_j^T + p_{j\sigma(k)}e_je_{\sigma(k)}^T\right) \\ Q_{12}^{(\sigma(k))} &= \sum_{j\in\mathcal{N}_{\sigma(k)}} e_j \left(\sum_{h\in\mathcal{N}_j/\sigma(k)} p_{jh}e_h^T\right) \\ Q_{21}^{(\sigma(k))} &= e_{\sigma(k)}e_{\sigma(k)}^T \\ Q_{22}^{(\sigma(k))} &= I - e_{\sigma(k)}e_{\sigma(k)}^T \end{aligned}$$

being e_ℓ, $\ell \in \{1,\dots,N\}$, the N-dimensional vector having all the components equal to zero except the ℓ-th component which is equal to 1. Observe that, for $\sigma(k) \in \{1,\dots,N\}$, $Q_{\sigma(k)}$ is a $2N$-dimensional stochastic matrix. Finally let

$$B_{\sigma(k)} = \begin{bmatrix} \sum_{j\in\mathcal{N}_{\sigma(k)}} e_j^T b \\ 0_N \end{bmatrix}$$

Assume, without loss of generality, that node $\sigma(k)$ is the node performing the transmission during the $(k+1)$-th iteration of the a-CL. Hence the updating step of a-CL can be written in vector form as

$$\begin{bmatrix} \mathbf{x}'(k+1) \\ \mathbf{x}''(k+1) \end{bmatrix} = Q_{\sigma(k)} \begin{bmatrix} \mathbf{x}'(k) \\ \mathbf{x}''(k) \end{bmatrix} + B_{\sigma(k)}, \quad k \geq \bar{k} \tag{2.5}$$

Now let us introduce the auxiliary variable

$$\xi(k) = \begin{bmatrix} \mathbf{x}'(k) \\ \mathbf{x}''(k) \end{bmatrix} - \begin{bmatrix} \mathbf{x}^*_{\text{opt}} \\ \mathbf{x}^*_{\text{opt}} \end{bmatrix}.$$

By exploiting the fact that, for $\sigma(k) \in \{1,\dots,N\}$,

$$\begin{bmatrix} \mathbf{x}^*_{\text{opt}} \\ \mathbf{x}^*_{\text{opt}} \end{bmatrix} = Q_{\sigma(k)} \begin{bmatrix} \mathbf{x}^*_{\text{opt}} \\ \mathbf{x}^*_{\text{opt}} \end{bmatrix} + B_{\sigma(k)} \tag{2.6}$$

we have that the variable ξ satisfies the following $2N$-dimensional recursive equation

$$\xi(k+1) = Q_{\sigma(k)}\xi(k). \tag{2.7}$$

Observe that $\hat{\mathbf{x}}(k) \to \mathbf{x}^*_{\text{opt}} + \alpha\mathbf{1}$ if and only if $\xi(k) \to \alpha\mathbf{1}$. Moreover, since $Q_{\sigma(k)}$ is a stochastic matrix for any $\sigma(k) \in \{1,\dots,N\}$, we have that (2.7) represents a $2N$-dimensional time-varying consensus algorithm.

In next sections, we analyze the convergence properties and the robustness to delays and packet losses of the a-CL algorithm by studying system (2.7) resorting to the mathematical tools developed in the literature of the consensus algorithms. In particular we will provide our results considering two different scenarios which are formally described in the following definitions.

Definition 2.6 (Randomly persistent comm. network) A network of N nodes is said to be a *randomly persistent communicating network* if there exists a N-upla $(\beta_1,\dots,\beta_N)$ such that $\beta_{\sigma(k)} > 0$, for all $\sigma(k) \in \{1,\dots,N\}$, and $\sum_{\sigma(k)=1}^N \beta_{\sigma(k)} = 1$, and such that, for all $k \in \mathbb{N}$,

$$\mathbb{P}\left[\text{the transmitting node at iteration } k \text{ is node } \sigma(k)\right] = \beta_{\sigma(k)}.$$

Definition 2.7 (Uniformly persistent comm. network) A network of N nodes is said to be a *uniformly persistent communicating network* if there exists a positive integer number τ such that, for all $k \in \mathbb{N}$, each node transmits the value of its estimate to its neighbors at least once within the time interval $[k, k+\tau)$.

2.5 Performance analysis of a-CL algorithm under randomly persistent communications

The following result characterizes the convergence properties of the a-CL when the network is randomly persistent communicating.

Proposition 2.8 Consider a *randomly persistent communicating network* of N nodes running the a-CL algorithm over a weakly connected measurement graph $\mathcal{G}_m$. Let ϵ be such that $0 < \epsilon < 1/(2d_{\max} R_{min}^{-1})$. Moreover let $\hat{x}_i$, $i \in \{1, \dots, N\}$, $\hat{x}_j^{(i)}$, $j \in \mathcal{N}_i$, be initialized to any real number. Then the following facts hold true

(i) the evolution $k \to \hat{\mathbf{x}}(k)$ converges almost surely to an optimal solution $\mathbf{x}_{\text{opt}} \in \chi$, i.e., there exists $\alpha \in \mathbb{R}$ such that
$$\mathbb{P}\left[\lim_{k\to\infty} \hat{\mathbf{x}}(k) = \mathbf{x}_{\text{opt}}^* + \alpha\mathbf{1}\right] = 1.$$

(ii) the evolution $k \to \hat{\mathbf{x}}(k)$ is exponentially convergent in mean-square sense, i.e., there exist $C > 0$ and $0 \le \rho < 1$ such that
$$\begin{aligned}\lim_{k\to\infty} \mathbb{E}&\left[\|\hat{\mathbf{x}}(k) - (\mathbf{x}_{\text{opt}}^* + \alpha\mathbf{1})\|^2\right] \\ &\le C\rho^k \mathbb{E}\left[\|\hat{\mathbf{x}}(0) - (\mathbf{x}_{\text{opt}}^* + \alpha\mathbf{1})\|^2\right].\end{aligned}$$

Proof The proof of Proposition 2.8 is based on proving the convergence to consensus of (2.7) using the mathematical tools developed in [44]. Let σ be the random process such that $\sigma(k)$ denotes the node performing the transmission action at the beginning of the $k+1$-th iteration. Clearly, in the randomized scenario we are considering, we have that, for $i \in \{1, \dots, N\}$, $\mathbb{P}[\sigma(k) = i] = \beta_i$ for all k. Let
$$S(k) = \prod_{h=0}^{k} Q_{\sigma(h)}.$$

Observe that $S(k)$ inherits the same block structure of the matrices $\{Q_i\}_{i=1}^N$, namely we can write
$$S(k) = \begin{bmatrix} S_{11}(k) & S_{12}(k) \\ S_{21}(k) & S_{22}(k) \end{bmatrix}$$

As consequence of Theorem 3.1 in [44] the a-CL reaches almost surely consensus if and only if, for every i and j in V
$$\mathbb{P}\left[\mathcal{E}_{ij}\right] = 1, \tag{2.8}$$
where
$$\mathcal{E}_{ij} = \{\exists \ell, \exists k \,|\, S_{i\ell}(k) S_{j\ell}(k) > 0\}.$$

Now observe that, since the measurement graph is weakly connected, then the communication graph is a connected undirected graph. This fact together with the fact the diagonal elements of

$Q_{11}^{(i)}$ are all positive for any $i \in \{1, \dots, N\}$ implies that there exists almost surely $\bar{k}$ such that, for all $k' \geq \bar{k}$, all the elements of the matrix $S_{11}(k')$ are strictly greater than 0. Assume now, without loss of generality, that $\sigma(k') = i$, for $k' \geq k$. Then, since the i-th row of $S_{21}(k'+1)$ is equal to $e_i e_i^T S_{11}(k')$, it turns out that, all the elements of the i-th row of $S_{21}(k'+1)$ are strictly greater than 0. Moreover, it is easy to see that they will remain strictly greater than 0 also for any $k'' \geq k'$. Hence we can argue that, there exists almost surely, also a $\bar{k}'$ such that for all $k' \geq \bar{k}'$, all the elements of the matrix $S_{21}(k')$ are strictly greater than 0. It follows that the property stated in (2.8) is satisfied for any $k \geq \bar{k}'$ and for any $\ell \in \{1, \dots, N\}$. This concludes the proof of item (i).

Concerning item (ii), we again resort to the results in [44]. Let $\Omega = I - \frac{1}{2N}\mathbf{1}\mathbf{1}^T$ where in this expression we assume that I is the $2N$-dimensional identity matrix and the vector $\mathbf{1}$ is $2N$-dimensional. From the results in [44], it follows that to study the rate of convergence of $\mathbb{E}\left[\|\xi(k) - \alpha\mathbf{1}\|^2\right]$ is equivalent to study the convergence rate of $\mathbb{E}\|\Omega\xi(k)\|^2$ and in particular of the linear recursive system

$$\Delta(t+1) = \mathbb{E}\left[Q_{\sigma(0)}^T \Delta(t) Q_{\sigma(0)}\right]$$

where $\Delta(0) = \Omega$. Observe that $\Delta(t)$ is the evolution of a linear dynamical system which can be written in the form

$$\Delta(t+1) = \mathcal{L}(\Delta(t))$$

where $\mathcal{L} : \mathbb{R}^{2N\times 2N} \to \mathbb{R}^{2N\times 2N}$ is given by

$$\mathcal{L}(M) = \mathbb{E}\left[Q_{\sigma(0)}^T M Q_{\sigma(0)}\right].$$

As highlighted in [44], the linear operator $\mathcal{L}$ can be represented by the matrix $\mathbf{L} = \mathbb{E}[\mathbf{Q}_{\sigma(\mathbf{0})} \otimes \mathbf{Q}_{\sigma(\mathbf{0})}]^{\mathbf{T}}$ where $\otimes$ denotes the Kronecker product of matrices. Following the proof of Proposition 4.3 of [44], one can see that $\mathbf{L}^{\mathbf{T}}$ is a primitive stochastic matrix which, therefore, has the eigenvalue 1 with algebraic multiplicity 1. Moreover, $\mathbf{L}^{\mathbf{T}}(\mathbf{1} \otimes \mathbf{1}) = (\mathbf{1} \otimes \mathbf{1})$ and $(\mathbf{1} \otimes \mathbf{1})(\Omega \otimes \Omega) = 0$, from which it follows that $\mathbb{E}\|\Omega\xi(k)\|^2 \leq C\rho_{ess}(\mathbf{L}^{\mathbf{T}})\mathbb{E}\|\mathbf{\Omega}\xi(\mathbf{0})\|^{\mathbf{2}}$ where $\rho_{ess}(\mathbf{L}^{\mathbf{T}})$ denotes the essential spectral radius of $\mathbf{L}^{\mathbf{T}}$. ♢

2.5.1 Bounds on the convergence rate of the a-CL algorithm

In this section we provide some insights on the convergence rate of the a-CL algorithm in the randomly persistent communicating scenario. To do so, we consider Eqn. (2.7) whose performance in terms of rate of convergence to the consensus can be analyzed following again the treatment in [44]. Typically, one would like to study the convergence rate of a randomized consensus algorithm by providing a mean-square analysis of the behavior of the distance between the state and the asymptotic consensus point, namely, by analyzing the rate of convergence of the quantity $\mathbb{E}\left[\|\xi - \alpha\mathbf{1}\|^2\right]$. Unfortunately, this is not a trivial task in general. To overcome this difficulty we study the evolution of $\Omega\,\xi$. The first consequence of the results obtained in [44] is that the quantities $\mathbb{E}\left[\|\xi - \alpha\mathbf{1}\|^2\right]$ and $\mathbb{E}\left[\|\Omega\xi\|^2\right]$ have the same exponential convergence rate to zero, or, more formally, given any initial condition $\xi(0)$,

$$\limsup_{k\to\infty} \mathbb{E}\left[\|\xi(k) - \alpha\mathbf{1}\|^2\right]^{1/k} = \limsup_{k\to\infty} \mathbb{E}\left[\|\Omega\xi(k)\|^2\right]^{1/k}.$$

For this reason, in what follows we study the right-hand expression, which turns out to be simpler to analyze. In order to have a single performance metric not dependent on the initial condition, we focus on this worst case exponential rate of convergence

$$R = \sup_{\xi(0)} \limsup_{k\to\infty} \mathbb{E}\left[\|\Omega\xi(k)\|^2\right]^{1/k}$$

It has been proved in Proposition 4.4 of [44] that

$$\left[\rho_{\text{ess}}(\bar{Q})\right]^2 \leq R \leq sr(\mathbb{E}(Q_i^T \Omega Q_i)). \tag{2.9}$$

where $\bar{Q}$ is the average consensus matrix, namely, $\bar{Q} = \mathbb{E}[Q_i] = \sum_{i=1}^{N} \beta_i Q_i$, and where $sr(\mathbb{E}(Q_i^T \Omega Q_i))$ denotes the spectral radius of the semidefinite positive matrix $\mathbb{E}(Q_i^T \Omega Q_i)$, i.e., its largest eigenvalue. Unfortunately, it turns out from a numerical inspection over significant families of graphs, like Cayley graphs (see [40]) and random geometric graphs, that the upper bound $sr(\mathbb{E}(Q_i^T \Omega Q_i))$ is greater than 1, that is, it is not informative for our analysis. However we have run a number of MonteCarlo simulations randomized over graphs of different topology and size and over different initial conditions, and it always resulted that $\limsup_{k\to\infty} \mathbb{E}\left[\|\Omega\xi(k)\|^2\right]^{1/k} \leq \rho_{\text{ess}}(\bar{Q})$. Based on this experimental evidence we formulate the following conjecture.

Conjecture 2.9 The quantity $\rho_{ess}(\bar{Q})$ is an upper bound for the exponential convergence rate R, i.e.,

$$R \leq \rho_{\text{ess}}(\bar{Q})$$

The above conjecture and the fact that $\left[\rho_{\text{ess}}(\bar{Q})\right]^2 \leq R$ motivates to study $\rho_{ess}(\bar{Q})$.

Remark 2.10 Notice that equation (2.7) describes a higher order consensus algorithm, for which few analytic tools are available. In fact, the few works available in the literature which address the rate of convergence of randomized higher order consensus algorithms are limited to the convergence in expectation [36].

2.5.2 Rate Analysis of a-CL algorithm for regular graphs

In this section we assume that the measurements graph $\mathcal{G}_m = (V, \mathcal{E}_m)$ is a strongly connected bidirected regular graph such that, for $i \in \{1, \ldots, N\}$, $|\mathcal{N}_i| = \nu$. Moreover we assume the following properties.

Assumption 2.11 We have that

(i) the error measurements covariances are all identical, i.e., $R_{ij} = R$ for all $(i, j) \in \mathcal{E}_m$;

(ii) $\epsilon = R/(2(\nu + 1))$;

(iii) the probabilities $\{\beta_1, \ldots, \beta_N\}$ are uniform, i.e., $\beta_1 = \ldots = \beta_N = 1/N$.

Observe that, from properties (i) and (ii) of Assumption 2.11, it turns out that the matrix $P = I - \epsilon A^T R^{-1} A$, associated to the s-CL algorithm, is symmetric and such that $P_{ij} = 1/(\nu + 1)$ for $j \in \mathcal{N}_i \cup \{i\}$. Let $\lambda_1(P) = 1 > \lambda_2(P) \geq \ldots \geq \lambda_N(P)$ be the eigenvalues of P. Then $\rho_{\text{ess}}(P) = \max\{|\lambda_2(P)|, |\lambda_N(P)|\}$. The following Lemma illustrates how the $2N$ eigenvalues of $\bar{Q}$ are related to those of P.

Lemma 2.12 Consider the a-CL algorithm running over a bidirected regular graph $\mathcal{G}_m = (V, \mathcal{E}_m)$ such that, for $i \in \{1, \ldots, N\}$, $|\mathcal{N}_i| = \nu$. Assume Assumption 2.11 holds true. Then the $2N$ eigenvalues of $\bar{Q}$ are the solutions of the following N second-order equations

$$f(s; \lambda_i, N, \nu) = s^2 + (a + b)s + (ab + c) \tag{2.10}$$

where

$$a = -\left[\frac{N-\nu}{N} + \frac{\lambda_i}{N} + \frac{\nu - 1}{N(\nu + 1)}\right]$$
$$b = -\frac{N-1}{N}, \quad c = -\frac{\nu - 1}{N^2}(\lambda_i - \frac{1}{\nu + 1})$$

Now let $s_1^{(i)}$ and $s_2^{(i)}$ denote the two solutions of $f(s;\lambda_i,N,\nu)$. It easy to see that $s_1^{(1)}=1$ and $s_2^{(1)}=1-\frac{\nu^2+1}{N(\nu+1)}$. The following result restricts the search of $\rho_{\text{ess}}(\bar{Q})$ among the values $|s_1^{(2)}|$, $|s_2^{(2)}|$ and $1-\frac{\nu^2+1}{N(\nu+1)}$.

Theorem 2.13 Consider the a-CL algorithm running on an bidirected regular graph $\mathcal{G}_m=(V,\mathcal{E}_m)$ such that, for $i\in\{1,\ldots,N\}$, $|\mathcal{N}_i|=\nu$. Assume Assumption 2.11 holds true. Moreover let
$\gamma^*=\frac{\nu-1+N(\nu+1)-\sqrt{N^2(\nu+1)^2-2N(\nu^3+\nu+2)+(\nu-1)^2+(\nu^2+1)^2}}{\nu+1}$
then

(i) if $1-\rho_{\text{ess}}(P)\leq\gamma^*\implies\rho_{\text{ess}}(\bar{Q})=\max(|s_1^{(2)}|,|s_2^{(2)}|)$;

(ii) if $1-\rho_{\text{ess}}(P)>\gamma^*\implies\rho_{\text{ess}}(\bar{Q})=s_2^{(1)}=1-\frac{\nu^2+1}{N(\nu+1)}$.

The proofs of Lemma 2.12 and Theorem 2.13 follows from standard algebraic manipulations. Due to space constraints we do not include them here, but we refer the interested reader to the document [45].

We provide now an asymptotic result on $\rho_{\text{ess}}(\bar{Q})$. To do so, consider a sequence of connected undirected regular graphs $\mathcal{G}_N$ of increasing size N, and fixed degree ν. Assume Assumption 2.11 holds true for any $\mathcal{G}_N$. Then to any $\mathcal{G}_N$ we can associate a stochastic matrix P_N such that $(P_N)_{ij}=1/(\nu+1)$ for all $j\in\mathcal{N}_i\cup\{i\}$. Let us assume the following property.

Assumption 2.14 Consider the sequence of matrices P_N associated to the sequence of graphs $\mathcal{G}_N$ above described and assume that

$$\rho_{\text{ess}}(P_N) \quad = \quad 1-\varepsilon(N)+o(\varepsilon(N)) \tag{2.11}$$

where $\varepsilon:\mathbb{N}\to\mathbb{R}$ is a positive function such that $\varepsilon(N)\to 0$ as $N\to\infty$.

Important families of matrices satisfying the above assumption (2.11) are given by the matrices built over the d-dimensional tori and the Cayley graphs (see [40]). It is worth remarking that the tori and the Cayley graphs have been shown to exhibit important spectral similarities with the random geometric graphs [46], which is a family of graphs that, during the last years, has been successfully used to model wireless communication in many applications [47]. Now, let the matrix $\bar{Q}_N$ represent the average matrix associated to the a-CL algorithm running over $\mathcal{G}_N$. The following result characterizes the asymptotic behavior of $\rho_{\text{ess}}(\bar{Q}_N)$, with respect to $\rho_{\text{ess}}(P_N)$.

Proposition 2.15 Consider the sequence of graphs $\mathcal{G}_N$ described above. Consider the a-CL algorithm running over $\mathcal{G}_N$. Assume Assumption 2.11 and Assumption 2.14 hold true. Then

$$\rho_{\text{ess}}(\bar{Q}_N) \quad = \quad 1-\frac{\nu(\nu+1)}{N(\nu^2+1)}\varepsilon(N)+o\left(\frac{\varepsilon(N)}{N}\right). \tag{2.12}$$

Proof Let $\gamma_i=1-\lambda_i$, then we can rewrite Eqn. (2.10) as:

$$f(s;\lambda_i,N,\nu)=d(s;N,\nu)+\gamma_i n(s;N,\nu)\triangleq g(s;\gamma_i,N,\nu)$$

so that g is an explicit function of γ_i, where

$$\begin{aligned} d(s;N,\nu) &= s^2 - \frac{2N(\nu+1)-(\nu^2+1)}{N(\nu+1)}s + 1 - \frac{\nu^2+1}{N(\nu+1)} \\ n(s;sN,\nu) &= \frac{s}{N} + \frac{\nu - N}{N^2} \end{aligned}$$

Note that

$$\lim_{N\to\infty} \gamma^*(\nu,N) = \frac{\nu^2+1}{(\nu+1)^2}$$

therefore, according to Theorem 2.13 and assumption 2.14, for N sufficiently large, $\rho_{ess}(\bar{Q}_N)$ is given by (i). Since $\gamma_2 = 1 - \lambda_2 + \epsilon(N) + o(\epsilon(N))$, then $s_1^{(2)} = \bar{s}_1^{(2)} + \alpha\varepsilon(N) + o(\varepsilon(N))$ and $s_2^{(2)} = \bar{s}_2^{(2)} + \beta\varepsilon(N) + o(\varepsilon(N))$ for some scalar α, β, where $\lambda_2 = \rho_{ess}(P_N)$ and $\bar{s}_1^{(2)}, \bar{s}_2^{(2)}$ are the solutions of second order equation $g(s;0,N,\nu) = 0$. It is easy to verify that $\bar{s}_1^{(2)} = 1$ and $\bar{s}_2^{(2)} = 1 - \frac{\nu^2+1}{N(\nu+1)}$. Since $|\bar{s}_1^{(2)}| > |\bar{s}_2^{(2)}|$, then for N sufficiently large and by continuity we have $\rho_{ess}(\bar{Q}_N) = |s_1^{(2)}|$. We are therefore interested in explicitly computing the scalar α. Since

$$\begin{cases} g(1;0,N,\nu) = 0 \\ \frac{\partial g}{\partial s}\Big|_{(1,0,N,\nu)} \neq 0 \end{cases}$$

it is possible to exploit the implicit function theorem that allows us to write:

$$\begin{aligned} s_1^{(2)} &= 1 - \frac{\partial g}{\partial \gamma_i}\left(\frac{\partial g}{\partial s}\right)^{-1}\Bigg|_{(1,0,N,\nu)} (\epsilon(N) + o(\epsilon(N))) \\ &= 1 - \frac{\nu(\nu+1)}{N(\nu^2+1)}\epsilon(N) + o\left(\frac{\epsilon(N)}{N}\right) \end{aligned}$$

which means that $\rho_{ess}(\bar{Q})$ can be expressed as

$$\rho_{\text{ess}}(\bar{Q}) = 1 - \frac{\nu(\nu+1)}{N(\nu^2+1)}\epsilon(N) + o\left(\frac{\epsilon(N)}{N}\right) \tag{2.13}$$

◇

Thank to [44], we know that the rates of convergence are lower bounded by $\rho_{\text{ess}}(\bar{Q}_N)^2$, while we recall we conjecture that $R \leq \rho_{\text{ess}}(\bar{Q}_N)$. From the above Proposition we can get the following result which compares the convergence rate R of the a-CL algorithm with respect to the convergence rate of the s-CL algorithm.

Corollary 2.16 Consider a sequence of graphs $\mathcal{G}_N$ as in Proposition 2.15. Consider the a-CL algorithm running over $\mathcal{G}_N$. Assume Assumption 2.11 and Assumption 2.14 hold true. Then, for $N \gg 1$,

$$\frac{1 - \rho_{\text{ess}}(\bar{Q}_N)}{1 - \rho_{\text{ess}}(P_N)} \simeq \frac{\nu(\nu+1)}{N(\nu^2+1)}$$

and

$$\frac{1 - \left[\rho_{\text{ess}}(\bar{Q}_N)\right]^2}{1 - \rho_{\text{ess}}(P_N)} \simeq 2\frac{\nu(\nu+1)}{N(\nu^2+1)}$$

namely, assuming Conjecture 2.9 holds true, the a-CL algorithm slows down of a factor $1/N$ with respect to the synchronous implementation.

Observe that the fact the rate of convergence in expectation is reduced by a factor N, is not surprising because in the a-CL there is only one node transmitting information at each iteration.

Remark 2.17 It is worth remarking that also standard memoryless asynchronous consensus algorithms based on asymmetric broadcast communication protocols, slow down their convergence rate by a factor $1/N$ with respect to the standard synchronous consensus implementations, see [48]. In other words, the presence of memory storage in the a-CL does not further deteriorate the convergence rate with respect to standard memoryless asynchronous consensus algorithms.

Remark 2.18 A similar analysis can be provided also for other relevant families of regular graphs like the complete graphs and, more in general, the Ramanujan graphs [49]. Let us recall the asymptotic lower bound proved by Alon and Boppana for doubly stochastic matrices built over ν-regular bidirected graphs. Specifically, if A denotes the adjacency matrix of a ν-regular bidirected graph, let P be the doubly stochastic matrix defined as $P = \nu^{-1}A$, then

$$\liminf_{N\to\infty} \rho_{\text{ess}}(P) \geq \frac{2\sqrt{\nu-1}}{\nu}$$

where the $\liminf$ is taken along the family of all ν-regular bidirected graphs having N vertices. Ramanujan graphs are those ν-regular bidirected graphs which achieves the previous bound, i.e., such that $\rho_{\text{ess}}(P) = \frac{2\sqrt{\nu-1}}{\nu}$. Hence through the Ramanujan graphs it is possible to keep the essential spectral radius bounded away from 1, while keeping the degree fixed. Exploiting Theorem 2.13, it is possible to prove that, for the a-CL algorithm running over Ramanujan graphs, it holds $\rho_{ess}(\bar{Q}) = 1 - \alpha(\nu)/N$, where $\alpha(\nu) \leq 1$ depends only on the degree ν, and, in turn,

$$\frac{1-\rho_{\text{ess}}(\bar{Q})}{1-\rho_{\text{ess}}(P)} \simeq \frac{\alpha(\nu)\,\nu}{N\sqrt{\nu+1}}$$

In other words also for the Ramanujan graphs, the a-CL algorithm is slowered down by a factor of $1/N$ with the respect to the synchronous implementation. Concerning the complete graphs we have that $\rho_{\text{ess}}(P) = 0$ and, again from Theorem 2.13, that $\rho_{ess}(\bar{Q}) = \frac{2(N-1)}{N}$. Hence it follows that, for $N \gg 1$,

$$\frac{1-\rho_{\text{ess}}(\bar{Q})}{1-\rho_{\text{ess}}(P)} \simeq 1.$$

namely, the a-CL algorithm is not slowered down by a factor N. This is due to the fact that, when a complete graph is employed, the number of neighbors of each node linearly increases with the size of the graph. Due to space constraints, we do not include here all the details of the analysis related to the Ramanujan graphs which, however, can be found in [45].

It is worth remarking that, even though there are plenty of Ramanujan graphs, it is still an open problem if for any pair N and ν there exist Ramanujan graphs with N vertices and of degree ν. Moreover, even if they exist, their construction is quite complex, thus making them of marginal interest from an application standpoint.

Remark 2.19 Following Remark 2.4, it is worth stressing that also the Jacobi-like strategy introduced in [2] is amenable of asynchronous implementation, see [1]. However, to the best of our knowledge, no theoretical analysis of the rate of convergence of the asynchronous version, introduced in [1], has been proposed in the literature.

2.6 Robustness properties of the a-CL algorithm with respect to packet losses and delays

In section 2.4 we have introduced the a-CL algorithm assuming that the communication channels are reliable, i.e., no packet losses occur, and that the transmission delays are negligible. In this section we relax these assumptions and we show that the a-CL algorithm still converges provided that the network is uniformly persistent communicating and the transmission delays and the frequencies of communication failures satisfy mild conditions which we formally describe next.

Assumption 2.20 (Bounded packet losses) There exists a positive integer L such that the number of consecutive communication failures between every pair of neighboring nodes in the communication graph $\mathcal{G}_c$ is less than L.

Assumption 2.21 (Bounded delay) Assume node i broadcasts its estimate to its neighbors at the beginning of iteration k, and, assume that, the communication link (i, j) does not fail. Then, there exists a positive integer D such that the information $\hat{x}_i(k)$ is used by node j to perform its local update not later than iteration $k + D$.

Loosely speaking Assumption 2.20 implies that there can be no more than L consecutive packet losses between any pair of nodes i, j belonging to the communication graph. Differently, Assumption 2.21 consider the scenario where the received packets are not used instantaneously, but are subject to some delay no greater than D iterations.

Clearly, in this more realistic scenario, it turns out that the implementation of the a-CL is slightly different from the description provided in Section 2.4. Specifically, consider the k-th iteration of the a-CL algorithm and, without loss of generality, assume node i is the transmitting node during this iteration. Due to the presence of packet losses and delays, it might happen that the set of updating nodes is, in general, different from the set $\mathcal{N}_i$. In fact, for $j \in \mathcal{N}_i$, node j does not perform any update since the packet $\hat{x}_i(k)$ from node i is lost or simply because the update is delayed. Moreover there might be a node $h \notin \mathcal{N}_i$ which, during iteration k, decides to perform an update since it received a packet $\hat{x}_s$, $s \in \mathcal{N}_h$, within the last D iterations. This scenario can be formally represented by the set of nodes $V'(k) \subseteq V$ which decide to perform an update at iteration k. Then, Eqn. (2.3) can be rewritten as

$$\hat{x}_j(k+1) := p_{jj}\hat{x}_j(k) + \sum_{h\in\mathcal{N}_j} p_{jh}\hat{x}_h(k'_h) + b_j, \tag{2.14}$$

for all $j \in V'(k)$, where $k - (\tau L + D) \leq k'_h \leq k$, i.e. loosely speaking when an update is performed, the local estimate of the neighbouring nodes cannot be older than $\tau L + D$ iterations[2]. Indeed, if $L = D = 0$, then we recover the standard a-CL algorithm where $V'(k) = \mathcal{N}_i$.

The following result characterizes the convergence properties of the a-CL in presence of delays, packet losses and when the network is uniformly persistent communicating.

Proposition 2.22 Consider a *uniformly persistent communicating network* of N nodes running the a-CL algorithm over a weakly connected measurement graph $\mathcal{G}_m$. Let Assumptions 2.20 and 2.21 be satisfied. Let ϵ be such that $0 < \epsilon < 1/(2d_{\max}R_{min}^{-1})$. Moreover let $\hat{x}_i$, $i \in \{1, \dots, N\}$, $\hat{x}_j^{(i)}$, $j \in \mathcal{N}_i$, be initialized to any real number. Then the following facts hold true

[2] Recall we are assuming the network is uniformly persistent communicating, namely, for all $k \in \mathbb{N}$, each node performs at least one transmission within the time interval $[k, k + \tau)$.

(i) the evolution $k \to \hat{\mathbf{x}}(k)$ asymptotically converges to an optimal estimate $\mathbf{x}_{\text{opt}} \in \chi$, i.e., there exists $\alpha \in \mathbb{R}$ such that
$$\lim_{k\to\infty} \hat{\mathbf{x}}(k) = \mathbf{x}^*_{\text{opt}} + \alpha\mathbf{1};$$

(ii) the convergence is exponential, namely, there exists $C > 0$ and $0 \leq \rho < 1$ such that
$$\|\hat{\mathbf{x}}(k) - \left(\mathbf{x}^*_{\text{opt}} + \alpha\mathbf{1}\right)\| \leq C\rho^k \|\hat{\mathbf{x}}(0) - \left(\mathbf{x}^*_{\text{opt}} + \alpha\mathbf{1}\right)\|.$$

Proof First we review the result stated in Proposition 1 in [50]. In [50], the authors consider the following consensus algorithm with delays[3]
$$x^i(k+1) = \sum_{j=1}^{m} a^i_j(k) x^j(k - t^i_j(k)) \tag{2.15}$$
where x^i denotes the state of node i, $i \in \{1, \dots, M\}$, the scalar $t^i_j(k)$ is nonnegative and it represents the delay of a message from agent j to agent i, while the scalar $a^i_j(k)$ is a nonnegative weight that agent i assigns to a delayed estimate $x^j(s)$ arriving from agent j at time k. It is assumed that the weights $a^i_j(k)$ satisfy the following assumption.

Assumption 2.23 There exists a scalar η, $0 < \eta < 1$ such that

(i) $a^i_i(k) \geq \eta$ for all $k \geq 0$;

(ii) $a^i_j(k) \geq \eta$ for all $k \geq 0$, and all agents j whose (potentially delayed) information $x^j(s)$ reaches agent i during the k-th iteration;

(iii) $a^i_j(k) = 0$ for all $k \geq 0$ and j otherwise.

(iv) $\sum_{j=1}^{m} a^i_j(k) = 1$ for all i and k.

For any k let the information exchange among the agents may be represented by a directed graph $(\mathcal{V}, \mathcal{E}_k)$, where $V = \{1, \dots, m\}$ with the set $\mathcal{E}_k$ of directed edges given by $\mathcal{E}_k = \left\{(j,i) | a^i_j(k) > 0\right\}$. The authors impose a connectivity assumption on the agent system, which is stated as follows.

Assumption 2.24 The graph $(V, \mathcal{E}_\infty)$ is connected, where $\mathcal{E}_\infty$ is the set of edges (j,i) representing agent pairs communicating directly infinitely many times, i.e., $\mathcal{E}_\infty = \{(j,i) | (j,i) \in \mathcal{E}_k \text{ for infinitely many indices } k\}$.

Additionally it is assumed that the intercommunication intervals are bounded for those agents that communicate directly. Specifically,

Assumption 2.25 There exists an integer $B \geq 1$ such that for every $(j,i) \in \mathcal{E}_\infty$, agent j sends information to its neighbor i at least once every B consecutive iterations.

Finally, it is assumed that the delays $t^i_j(k)$ in delivering a message from an agent j to any neighboring agent i is uniformly bounded at all times. Formally

[3] We adopt the notations of paper [50].

Assumption 2.26 Let the following hold:

(i) $t_i^i(k) = 0$ for all agents i and all $k \geq 0$.

(ii) $t_j^i(k) = 0$ for all agents j communicating with agent i directly and whose estimates x^j are not available to agent i during the k-th iteration.

(iii) There is an integer B_1 such that $0 \leq t_j^i(k) \leq B_1 - 1$ for all agents i, j, and all k.

The result illustrated in Proposition 1 of [50] is recalled in the following Proposition.

Proposition 2.27 Let Assumption 2.23, 2.24, 2.25, 2.26 hold. Then the sequences $\left\{x^{(i)}(k)\right\}$, $i = 1, \dots, m$, generated by Equation (2.15) converge exponentially to a consensus.

Thanks to this fact we show that a-CL algorithm in presence of delays and packet losses can be rewritten as a consensus with delays that satisfies Assumptions 2.23, 2.24, 2.25, and 2.26 beforehand reported.

To this aim, let $\delta_j(k) = \hat{x}_j(k) - [\mathbf{x}^*_{\text{opt}}]_\mathbf{j}$ where $[\mathbf{x}^*_{\text{opt}}]_j$ denotes the j-th component of the vector $\mathbf{x}^*_{\text{opt}}$. Recalling that $\mathbf{x}^*_{\text{opt}} = P\mathbf{x}^*_{\text{opt}} + b$ and, according to (2.14) we have that, if $j \in V'(k)$

$$\delta_j(k+1) := p_{jj}\delta_j(k) + \sum_{h \in \mathcal{N}_j} p_{jh}\delta_h(k'_h), \tag{2.16}$$

otherwise

$$\delta_j(k+1) = \delta_j(k).$$

The above equations describe a consensus algorithm on the variables $\delta_1, \dots, \delta_N$ which satisfies Assumptions 2.23, 2.24,2.25,2.26. Indeed Assumption 2.23 on the weights is trivially satisfied. Assumption 2.24 follows from the facts that the communication graph $\mathcal{G}_c$ is connected, the network is uniformly persistent communicating and from Assumptions 2.20 and 2.21. Assumption 2.25 is a consequence of the fact that the network is uniformly persistent communicating and Assumption 2.20; in our setup we have $B = L\tau$. Finally Assumption 2.26 follows from Assumption 2.21 and equation (2.16). Hence the variables $\delta_1, \dots, \delta_N$ converge exponentially to a consensus value α which, in turn, implies that $\hat{\mathbf{x}}$ converge exponentially to $\mathbf{x}^*_{\text{opt}} + \alpha\mathbf{1}$. ♢

Remark 2.28 We believe that the analysis of the robustness to packet losses of the a-CL algorithm might be performed also in the randomized scenario considered in Section 2.5 assuming that each transmitted packet might be lost with a certain probability. We leave this analysis as future research. However in the numerical section, specifically in Example 2.30, we show the effectiveness of the a-CL algorithm also in presence of random communication failures when the network is randomly persistent communicating.

Remark 2.29 Also the Jacobi-like strategy has been shown to be robust to packet losses, see [1]. Instead, concerning the other algorithms recently proposed in the literature, see [7, 10], to the best of our knowledge, no analysis considering the non-idealities introduced in this section has been proposed in the literature.

2.7 Numerical Results

In this Section we provide some simulations implementing the localization consensus-based strategy introduced in this paper.

Example 2.30 In this example we consider a random geometric graph generated by choosing $N = 100$ points randomly placed in the interval $[0, 1]$. Two nodes are connected and take measurements if they are sufficiently close, i.e more specifically, both measurements z_{ij} and z_{ji} are available provided that $|x_i - x_j| \leq 0.15$. This choice resulted in networks with an average number of neighbours per node of about 7. Every measurement was corrupted by Gaussian noise with covariance $\sigma^2 = 10^{-4}$. In this example we assumed that the network is randomly persistent communicating with uniform communication probabilities $(\beta_1, \ldots, \beta_N)$, namely, $\beta_1 = \ldots = \beta_N = 1/N$. Moreover the possibility of communication failure is taken into account. Specifically, supposing node i is transmitting, each node $j \in \mathcal{N}_i$ with a certain probability i.e., p_f, can not receive the sent packet.

In Figure 2.1 we plotted the behavior of the error

$$J(k) = \log\left(\|A(\hat{x}(k) - x^*)\|\right)$$

for different values of the failure probability p_f.

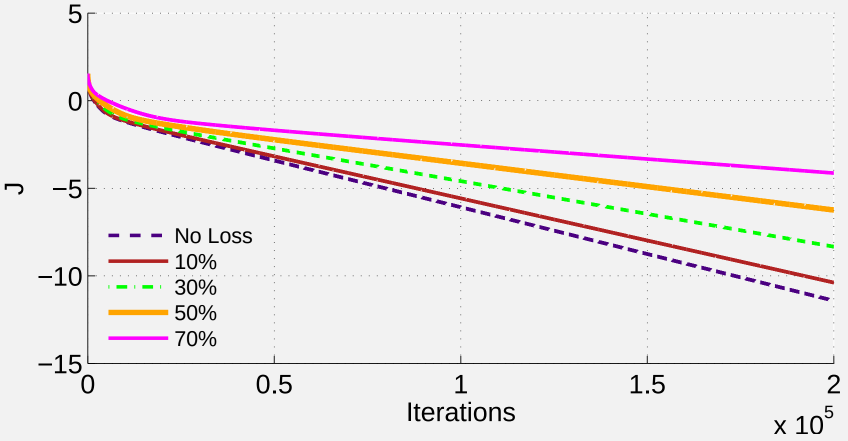


Figure 2.1: Behavior of J for a randomly persistent communicating network in a random geometric graph, for different values of the probability failure p_f.

The plot reported is the result of the average over 1000 Monte Carlo runs, randomized with respect to both the measurement graph[4] and the initial conditions. Observe that the trajectory of J decreases exponentially.

Example 2.31 In this example we tested the validity of conjecture 2.9. In Figure 2.2 (*top panel*) it is show the simulation considering a set of 2-dimensional torus graphs of increasing size N, while in Figure 2.2 (*bottom panel*) the simulation is performed considering a family of Random Geometric graphs. What we show is a comparison between the empirical rate of convergence of the algorithm, its lower bound, represented by $esr(\bar{Q})^2$, and the $esr(\bar{Q})$.

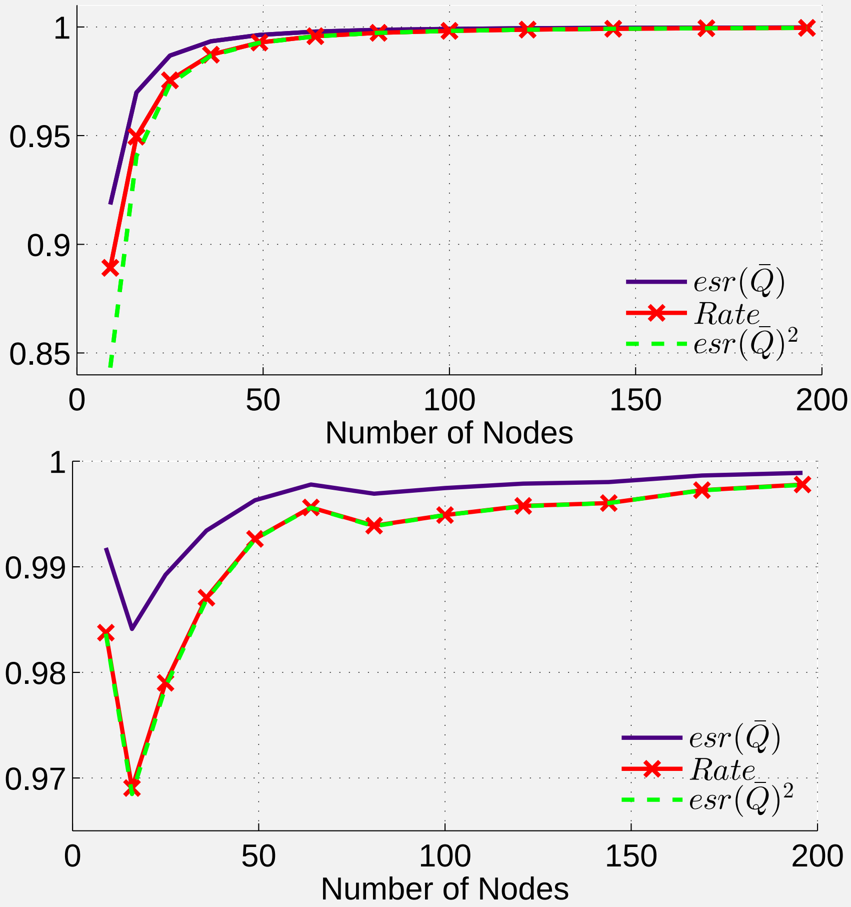


Figure 2.2: Trend of the rate of convergence, of the $esr(\bar{Q})^2$ and of the $esr(\bar{Q})$ for $2-d$ torus (*top panel*) and for random geometric graphs (*bottom panel*) of increasing size N.

Example 2.32 In this example we provide a numerical comparison with some well known algorithms proposed in literature which, for sake of the completeness, we briefly recall (Table 2.1).

Algorithm	**Sent packets per iteration**
a-CL	1
a-GL	1
BC	$\|N_i\|+1$
REK	$\|N_j\|+5$

Table 2.1: Number of sent packets per iteration for each algorithm.

The first algorithm considered, hereafter called **a-GL algorithm**, is proposed in [3]. Similarly to the a-CL algorithm, during its k-th iteration one node, say h, transmits its variable $\hat{x}_h$ to all its neighbors. For $l \in \mathcal{N}_h$, node l, based on the information received from node h, performs the following update

$$\begin{aligned}\hat{x}_l(k+1) &= 1/2\,(\hat{x}_l(k)+\hat{x}_h(k)+1/2(z_{lh}-z_{hl}))\\ &= \hat{x}_l(k)+1/2\,(\hat{x}_l(k)-\hat{x}_h(k)+1/2(z_{lh}-z_{hl}))\end{aligned}$$

while for $l \notin \mathcal{N}_h$ the state remains unchanged, i.e., $\hat{x}_l(k+1) = \hat{x}_l(k)$. Note that just one packet is transmitted at each iteration. Moreover, since this algorithm is known to reach mean square convergence [8], then its ergodic mean has been proposed as a possible estimator of the state.
The second algorithm, denoted hereafter as **BC algorithm**, is proposed in [7]. It requires a

coordinated broadcast communication protocol meaning that, during k-th iteration one node, say h, asks the variable $\hat{x}_l$ to all its neighbors $l \in \mathcal{N}_h$. When it receives the current state values, it performs the following greedy local optimization based on the current status of the network:

$$\begin{aligned} \hat{x}_h(k+1) \quad &:= \quad \operatorname{argmin}_{\hat{x}_h} \textstyle\sum_{(i,j)\in\mathcal{E}}^{|\mathcal{E}|} \|\hat{x}_i(k) - \hat{x}_j(k) - z_{ij}\|^2 \\ &= \quad \tfrac{1}{2|\mathcal{N}_h|} \textstyle\sum_{l\in\mathcal{N}_h} (2\hat{x}_l(k) - z_{lh} + z_{hl}) \end{aligned}$$

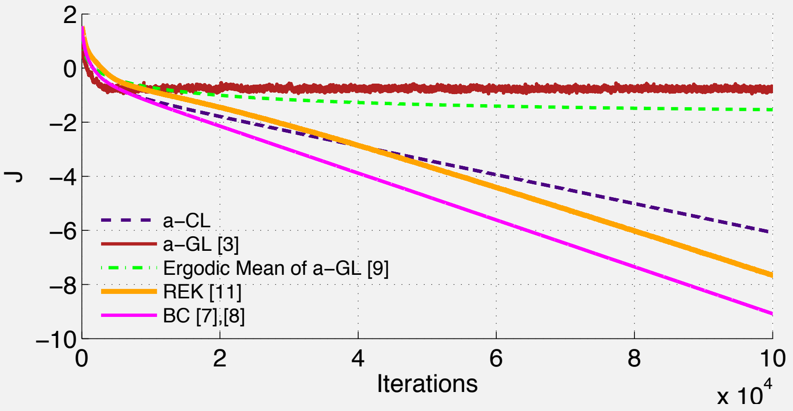
Note that the number of communications performed during one iteration are $|\mathcal{N}_h|+1$, since there is a broadcast packet sent by node h, and $|\mathcal{N}_h|$ packets sent by all its neighbours. We stress the fact that the Jacobi-like algorithm proposed [1] is indeed the same algorithm proposed in [7]. The last algorithm that we considered is the Randomized Extended Kaczmarz, hereafter called **REK algorithm**, presented in [10], consisting of two different update steps. The first step is an orthogonal projection of the noisy measurements onto the column space of the incidence matrix A in order to bound the measurements error. The second step is similar to the standard Kaczmarz update. Since a distributed implementation is not formally presented in [10], we propose the following. More specifically, let $s \in \mathbb{R}^M$ be the current projection of the noisy measurements onto the column space of A. Similarly as above, we denote with a little abuse of notation the e-th entry of s with the corresponding edge, i.e. $s_e = s_{ij}$. Then, the REK algorithm proposed in [10] for general least-squares problems, reduces in our setting to randomly and independently selecting a node h and an edge (i, j) at each iteration k according to the following probabilities:

$$p_h = \frac{|\mathcal{N}_h| + 1}{2M}; \quad p_{ij} = \frac{1}{M}$$

and then to performing the following local updates:

$$\begin{aligned} s_{\ell h}(k+1) &= s_{\ell h}(k) + \frac{\sum_{m\in\mathcal{N}_h}(s_{hm}(k) - s_{mh}(k))}{|\mathcal{N}_h|+1}, \quad \forall \ell \in \mathcal{N}_h \\ s_{h\ell}(k+1) &= s_{h\ell}(k) - \frac{\sum_{m\in\mathcal{N}_h}(s_{hm}(k) - s_{mh}(k))}{|\mathcal{N}_h|+1}, \quad \forall \ell \in \mathcal{N}_h \\ \hat{x}_i(k+1) &= \hat{x}_i(k) + \frac{z_{ij} - s_{ij}(k) - (\hat{x}_i(k) - \hat{x}_j(k))}{2} \\ \hat{x}_j(k+1) &= \hat{x}_j(k) - \frac{z_{ij} - s_{ij}(k) - (\hat{x}_i(k) - \hat{x}_j(k))}{2} \end{aligned}$$

We point out that, since in the updating step only local information is required, the algorithm is implemented in a distributed fashion and it exactly requires $|N_j| + 5$ communication rounds to perform an iteration. Specifically the first $|N_j| + 2$ are due to the update of the variable s and the last 3 are needed to update $\hat{x}$.



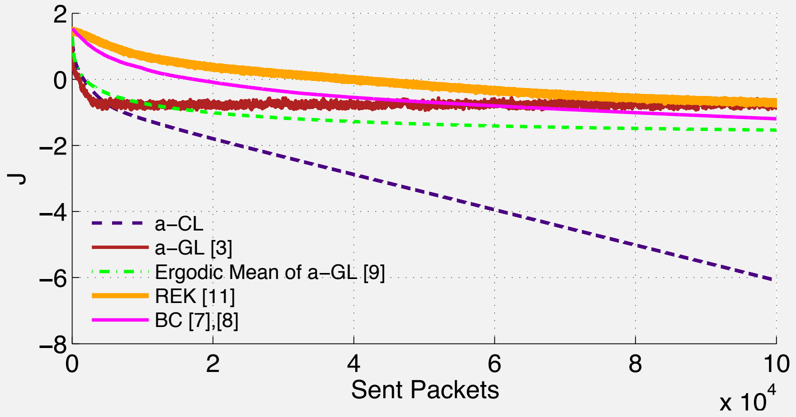


Figure 2.3: Comparison of various algorithms considering the number of iteration (*top panel*) and number of sent packets (*bottom panel*).

In this example we consider a random geometric measurement graph $\mathcal{G}$ built as in the previous example. In Figure 2.3 we plot the behavior of J respect to the number of iterations and sent packets. From these simulations, we observe that from an energy point of view the a-CL algorithm is the most convenient since the effective number of sent packets to achieve a certain estimation error, is lower. On the other hand if no energy constraint is imposed, then BC is the fastest algorithm, although not substantially faster than REK and a-CL.

As observed in [3] the local estimates of a-GL algorithm do not converge to the optimal solution, but they oscillate around it. However, a-GL exhibits the fastest transient among all algorithms and it is also energetically efficient. In our recent work, we thus proposed to combine the a-CL algorithm with the a-GL algorithm in order to have fast transient as well as guaranteed exponential asymptotic convergence by using suitable switching strategies [38].

3 Multi-Robot Localization via GPS and Relative Measurements

In this chapter we address the problem of multi-vehicle localization where the estimate has to be performed in real-time and communication is achieved via wireless communication. We propose to integrate less precise global sensors (GPS and compass) with more precise relative positioning sensors (range and bearing sensors) in order to achieve global high accuracy. Intuitively, precise range and bearing sensors would allow for the reconstruction of a relative formation but provides no information about the global position and orientation of the formation. Differently, compass and GPS installed in multiple vehicles can provide estimation of the centroid and orientation of the whole formation. The fusion of these two types of information would allow an accurate global positioning of all vehicles. The chapter is organized as follows. In Section 3.1 we introduce some mathematical notation useful later on. In Section 3.2 we present the measurement model and we formulate the maximum likelihood estimator and a possible linear approximation. In Section 3.3 we present a distributed and asynchronous solution of the problem highlighting its resilience to packet losses. Section 3.4 reports the numerical results.

3.1 Mathematical Preliminaries

Resorting to standard graph theory, the estimation problem can be naturally associated with an *undirected measurement graph* $\mathcal{G}_m = (V; \mathcal{E}_m)$ where $V \in \{1, \dots, N\}$ represents the nodes and $\mathcal{E}_m \subset V \times V$ contains the unordered pairs of nodes $\{i, j\}$ which are connected to and measure each other. We denote with $\mathcal{N}_i \subseteq V$ the set $\{j \mid \{i, j\} \in \mathcal{E}_m\}$, i.e. the neighboring set of node i. An undirected graph $\mathcal{G}_m$ is said to be connected if for any pair of vertices $\{i, j\}$ a path exists, connecting i to j. In the problem at hand, we consider a communication graph among the nodes which coincides with the measurements graph $\mathcal{G}_m$. Moreover, broadcast and asynchronous communications are assumed among the nodes. We denote with $| \cdot |$ the modulus of a scalar. Assuming M to be the cardinality of $\mathcal{E}_m$, the incidence matrix $A \in R^{M \times N}$ of $\mathcal{G}_m$ is defined as $A = [a_{ei}]$, where $a_{ei} = \{1, -1, 0\}$, if edge e is incident on node i and directed away from it, is incident on node i and directed toward it, or is not incident on node i, respectively. We denote with the symbol $\| \cdot \|$ the vector 2-norm and with $[\cdot]^T$ the transpose operator. The symbol $\odot$ represents the *Hadamard* product. Given a vector $\mathbf{v} \in \mathbb{R}^2$, the function $\text{atan2}(\cdot) : \mathbb{R}^2 \to [0, 2\pi]$ returns its angle, i.e., $\mathbf{v} = \|\mathbf{v}\| e^{j\,\text{atan2}(\mathbf{v})}$. Given a matrix $\mathbf{v} \in \mathbb{R}^{2 \times n}$, with $v_{\text{ctr.}}$ we denote the vector centroid, i.e., $v_{\text{crt.}} = \frac{1}{n} \sum_{i=1}^{n} v_i$, where v_i is the i- th row of the matrix. The symbol σ_x denotes the standard deviation of the generic measurement x. The operator $\mathbb{E}[\cdot]$ denotes the expected value, while $\text{proj}(\cdot) : \mathbb{R} \to \mathbb{R}^2$ denotes the function $\text{proj}(\theta) = \begin{bmatrix} \cos\theta & \sin\theta \end{bmatrix}^T$. Finally, $\mathcal{I}$ denotes the identity matrix of suitable dimensions.

3.2 Problem Formulation

Consider the problem of estimating the 2D positions, expressed in a common reference frame, of N nodes of a sensor network. Each node of the network is endowed with a set of sensors that provide both relative and absolute measurements.

In the following, firstly, we introduce the statistical models exploited for each type of measurements. Secondly, we formulate the non linear *Maximum-Likelihood* estimation problem. Thirdly, we introduce a suitable linear and convex reformulation.

3.2.1 Measurement Model

We assume that the N nodes are provided with a GPS module, a compass, a relative range sensor, and a relative bearing sensor. We denote with $p_i = (x_i, y_i),\ i \in V$, the 2D position of node i in a common inertial frame, and with θ_i its orientation with respect to the inertial North axis, which in the following we assume to coincide with the x-axis. Each sensor is described by the following statistical model:

(i) The GPS measurement $p_i^{\mathrm{GPS}} = (x_i^{\mathrm{GPS}}, y_i^{\mathrm{GPS}})$ represents a noisy measurement of $p_i = (x_i, y_i)$. We assume a normal distribution of the GPS measurements, that is $p_i^{\mathrm{GPS}} \sim \mathcal{N}(p_i, \sigma_p^2 \mathcal{I})$.

(ii) The compass provides a noisy measurement θ_i^C of θ_i. This is modelled according to an angular Gaussian distribution (see, e.g., [51]) which approximates the *Langevin* distribution [52]. This reads as $\mathrm{proj}(\theta_i^C) \sim\ \mathcal{N}\left(\mathrm{proj}(\theta_i), \sigma_\theta^2 \mathcal{I}\right)$.

(iii) The range sensor returns a noisy measurement r_{ij} of the distance between nodes i and j, which is modelled according to a normal distribution, that is $r_{ij} \sim \mathcal{N}(\|p_i - p_j\|, \sigma_r^2)$.

(iv) The bearing sensor returns a noisy measurement δ_{ij} of the bearing angle of the node j in the local frame of node i. For δ_{ij} we adopt an angular Gaussian distribution model which reads as $\mathrm{proj}(\delta_{ij}) \sim\ \mathcal{N}\left(\mathrm{proj}(\mathrm{atan2}(p_j - p_i) - \theta_i), \sigma_\delta^2 \mathcal{I}\right)$.

Remark 3.1 Observe that, in order to reduce the set-up cost, each node has access to highly noisy absolute measurements together with relative measurements that are less prone to noise than the absolute ones. In particular, the GPS sensors are usually characterized by a standard deviation $\sigma_p = 2$ [m] [53],[54], while the compass by a standard deviation $\sigma_\theta =\ 0.05$ [rad][55]. To retrieve information about range and bearing different methods can be used, e.g., depth-camera, laser, ultrasound. Acceptable values for the standard deviation of these measurements might be $\sigma_r = 0.1$ [m] and $\sigma_\delta = 0.03$ [rad]. Due to the variability in the accuracy of the available sensors, we will test our algorithm in a sufficiently wide range of standard deviation values.

For the sake of simplicity, we consider that all the nodes are endowed with a GPS module. However, a simple reformulation of the problem would still guarantee that all the results hold even if a reduced number of nodes are provided with a GPS.

3.2.2 Maximum-Likelihood Estimator

We assume that all the measurements are independent and their probability distributions are given in the previous section. It is possible to formulate the localization problem as a *Maximum-Likelihood* (ML) estimation problem [56]. Let us define the state and measurements sets, respectively, as

$$\begin{aligned} \mathbf{x} &= \{\mathbf{p}, \theta\} = \{p_i,\ \theta_i \text{ with } i \in V\}, \\ \mathbf{y} &= \left\{p_i^{\mathrm{GPS}},\ \theta_i^C,\ r_{hk},\ \delta_{hk} \text{ with } i \in V,\ (h,k) \in \mathcal{E}_m\right\}, \end{aligned}$$

where $\mathbf{p} := [p_1, \ldots, p_N]^T$ and $\theta := [\theta_1, \ldots, \theta_N]^T$. Then, the negative log-likelihood cost function can be written as

$$J(\mathbf{x}) := -\log f\big(\mathbf{y} \,|\, \mathbf{x}\big) = J_p + J_\theta + J_r + J_\delta + c, \tag{3.1}$$

where

$$\begin{aligned}
J_p &= \sum_{i=1}^{N} \frac{\|p_i - p_i^{\text{GPS}}\|^2}{2\sigma_p^2}, \\
J_\theta &= \sum_{i=1}^{N} \frac{\left\|\text{proj}(\theta_i^C) - \text{proj}(\theta_i)\right\|^2}{2\sigma_\theta^2}, \\
J_r &= \sum_{(i,j)=1}^{M} \frac{(r_{ij} - \|p_i - p_j\|)^2}{2\sigma_r^2}, \\
J_\delta &= \sum_{(i,j)=1}^{M} \frac{\|\text{proj}(\delta_{ij}) - \text{proj}\left(\text{atan2}(p_j - p_i) - \theta_i\right)\|^2}{2\sigma_\delta^2},
\end{aligned}$$

and c is a constant term that does not depend on $\mathbf{x}$ and $\mathbf{y}$. The minimization of the function in (3.1) would provide the maximum-likelihood estimator for the nodes absolute positions and orientations, i.e.:

$$\widehat{\mathbf{x}}^{\text{ML}} = \text{argmin}_{\mathbf{x}}\; J(\mathbf{x}). \tag{3.2}$$

The ML estimator benefits of some properties regarding its mean and its asymptotic behavior. In particular, consider the following equivalent parametrization of agents' positions using their centroid $p_{\text{ctr.}}$ and corresponding deviation Δp_i. This reads as

$$p_i = p_{\text{ctr.}} + \Delta p_i, \qquad \sum_i \Delta p_i = 0, \tag{3.3}$$

Let us also define $\Delta\mathbf{p} = (\Delta p_1, \ldots, \Delta p_N)$. Thanks to the new parametrization, equation (3.2) is equivalent to:

$$\begin{aligned}
\left\{\widehat{p}_{\text{ctr.}}^{\text{ML}}, \Delta\widehat{\mathbf{p}}^{\text{ML}}, \widehat{\boldsymbol{\theta}}^{\text{ML}}\right\} = \quad &\underset{\{p_{\text{ctr.}}, \Delta\mathbf{p}, \boldsymbol{\theta}\}}{\text{argmin}} \quad J(p_{\text{ctr.}}, \Delta\mathbf{p}, \boldsymbol{\theta}), \\
&\text{s.t.} \qquad \sum_i \Delta p_i = 0.
\end{aligned} \tag{3.4}$$

The previous reformulation allows us to prove the following lemma, which suggests how the ML estimator exploits the GPS information to solve for the absolute positioning of the formation centroid:

Lemma 3.2 Consider the negative log-likelihood cost function (3.1). Then, the maximum likelihood solution $\widehat{\mathbf{x}}^{\text{ML}}$ which solves (3.4) is such that

$$\widehat{p}_{\text{ctr.}}^{\text{ML}} = p_{\text{ctr.}}^{\text{GPS}}, \tag{3.5}$$

where $\widehat{p}_{\text{ctr.}}^{\text{ML}} := \frac{1}{N}\sum_{i=1}^{N} \widehat{p}_i$ and $p_{\text{ctr.}}^{\text{GPS}} := \frac{1}{N}\sum_{i=1}^{N} p_i^{\text{GPS}}$.

Proof Observe that only the term J_p of the log-likelihood cost function depends on $p_{\text{ctr.}}$. Indeed, J_θ is not a function of p_i; while, both J_r and J_δ depend only on the difference between p_i and p_j which, thanks to the equation (3.3) reads as

$$p_i - p_j = p_{\text{ctr.}} + \Delta p_i - p_{\text{ctr.}} - \Delta p_j = \Delta p_i - \Delta p_j.$$

It is then possible to consider only the log-likelihood relative to the GPS measurements. Specifically, if we define $p_i^{\mathrm{GPS}} = p_{\mathrm{ctr.}}^{\mathrm{GPS}} + \Delta p_i^{\mathrm{GPS}}$, it is possible to write

$$\begin{aligned} 2\sigma_p^2 J_p &= \sum_{i=1}^N \|p_{\mathrm{ctr.}} + \Delta p_i - (p_{\mathrm{ctr.}}^{\mathrm{GPS}} + \Delta p_i^{\mathrm{GPS}})\|^2 \\ &= \sum_{i=1}^N \left(\|p_{\mathrm{ctr.}} - p_{\mathrm{ctr.}}^{\mathrm{GPS}}\|^2 + \|\Delta p_i - \Delta p_i^{\mathrm{GPS}}\|^2 + 2(\Delta p_i - \Delta p_i^{\mathrm{GPS}})^T (p_{\mathrm{ctr.}} - p_{\mathrm{ctr.}}^{\mathrm{GPS}})\right) \\ &= N\|p_{\mathrm{ctr.}} - p_{\mathrm{ctr.}}^{\mathrm{GPS}}\|^2 + \sum_{i=1}^N \|\Delta p_i - \Delta p_i^{\mathrm{GPS}}\|^2, \end{aligned}$$

where we used the facts $\sum_i \Delta p_i = 0$ and $\sum_i \Delta p_i^{\mathrm{GPS}} = 0$. To minimize the first term on the right hand side we must have

$$p_{\mathrm{ctr.}} = p_{\mathrm{ctr.}}^{\mathrm{GPS}},$$

which proves the lemma. ♢

We can also state some limit behavior in a scenario where range, bearing and compass noises are very large or very small:

Lemma 3.3 For fixed GPS variance σ_p we have

(i) $\displaystyle \lim_{\max\{\sigma_\theta,\sigma_r,\sigma_\delta\}\to 0} \widehat{p}_i^{\mathrm{ML}} = p_{\mathrm{ctr.}}^{\mathrm{GPS}} + \Delta p_i$,

(ii) $\displaystyle \lim_{\min\{\sigma_r,\sigma_\delta\}\to +\infty} \widehat{p}_i^{\mathrm{ML}} = p_i^{\mathrm{GPS}}$.

Proof In the first scenario $\max\{\sigma_\theta, \sigma_r, \sigma_\delta\} \to 0$. This implies that the distributions for compass, range and bearing measurements converge to delta distributions, implying that

$$r_{ij} \to \|p_i - p_j\|, \quad \theta_i^C \to \theta_i, \quad \delta_{ij} \to \theta_i + atan2(p_j - p_i).$$

From these expressions it easily follows that

$$\widehat{p}_j - \widehat{p}_i \to p_j - p_i = r_{ij} e^{j(\delta_{ij} - \theta_i^C)}, \quad \{j, i\} \in \mathcal{E}_m \ ,$$

i.e., the relative vectorial distances among the communicating nodes are perfectly known. Since the graph is connected, it is possible to compute the exact vectorial difference among any two agents in the network, and therefore also the exact distance of any agent from the true centroid since:

$$\Delta\widehat{p}_i = \widehat{p}_i - \frac{1}{N}\sum_j \widehat{p}_j = \frac{1}{N}\sum_j (\widehat{p}_i - \widehat{p}_j) \to \frac{1}{N}\sum_j (p_i - p_j) = \Delta p_i.$$

Since $\widehat{p}_i = \widehat{p}_{\mathrm{ctr.}} + \Delta\widehat{p}_i$ and from Lemma 3.2 we have $\widehat{p}_{\mathrm{ctr.}} = p_{\mathrm{ctr.}}^{\mathrm{GPS}}$, then it follows the first part of the lemma.
In the second scenario when $\min\{\sigma_r, \sigma_\delta\} \to +\infty$ becomes arbitrary large, the probability distribution of range and bearing degenerate into an uniform distribution with infinite support. As so, the terms J_r and J_δ become negligible as compared to J_p and J_θ. Since the positions p_i do not appear in J_θ, it follows that $\widehat{p}_i$ results from the minimization of J_p, which gives $\widehat{p}_i = p_i^{\mathrm{GPS}}$ and, therefore, the claim of the lemma. ♢

Scenario 1) of Lemma 3.3 states that in the case where $\max\{\sigma_\theta, \sigma_r, \sigma_\delta\} \to 0$, the shape of the formation is perfectly retrieved. In this case the only source of error between the estimated formation and the ground-truth is given by the error between GPS centroid and the true centroid. Scenario 2) states that if the relative measurements accuracies deteriorate, the ML estimator will "trust" the GPS measurements only.
Unfortunately problem (3.2) is highly non linear and hard to solve. In particular, it is known that, if the angles are noise-free, the problem is linear [2]. Conversely, if the angles are not known, the problem presents many local minima [57, 58]. One possible way to tackle it, is using a standard gradient descent approach since the gradient vector of the log-likelihood function can be computed in closed form using (3.1). However, such approach heavily suffers of bad initialization. In fact, the presence of multiple local minima in the cost function (3.1) causes the algorithm to stop in the wrong minimizer.
In the following, we resort to a suitable approximation which let us reformulate the problem in a classical linear-least square framework.

3.2.3 An Approximated Linear Least-Squares Formulation

An approximated solution for the problem stated in (3.2), which exploits a suitable model linearization, is now presented. The idea is to move from the polar coordinate system to the equivalent Cartesian representation.
Indeed, assuming a perfect knowledge of range, bearing and compass, it is possible to express the *displacement* d_{ij} between agent i and j as

$$d_{ij} := p_i - p_j = r_{ij} \begin{bmatrix} \cos(\delta_{ij} + \theta_i) \\ \sin(\delta_{ij} + \theta_i) \end{bmatrix}. \tag{3.6}$$

Since the measurements are affected by noise, it is necessary to map the noise of range, bearing and compass into the equivalent noise in Cartesian coordinates. Namely, given the noisy version of (3.6), that is

$$d_{ij} = p_i - p_j + n_{ij}, \tag{3.7}$$

where n_{ij} is the noise in Cartesian coordinate, we want to find the expression for its covariance, $\mathbb{E}[n_{ij} n_{ij}^T] = \Sigma_{ij}$, in terms of the statistical description of range, bearing and compass measurements noises. After a first order expansion we obtain

$$\Sigma_{ij} = \begin{bmatrix} \sigma_x^2(i,j) & \sigma_{xy}(i,j) \\ \sigma_{yx}(i,j) & \sigma_y^2(i,j) \end{bmatrix}, \tag{3.8}$$

where

$$\begin{aligned} \sigma_x^2(i,j) &= \sigma_r^2 \cos^2(\delta_{ij} + \theta_i) + r_{ij}^2(\sigma_\delta^2 + \sigma_\theta^2) \sin^2(\delta_{ij} + \theta_i), \\ \sigma_y^2(i,j) &= \sigma_r^2 \sin^2(\delta_{ij} + \theta_i) + r_{ij}^2(\sigma_\delta^2 + \sigma_\theta^2) \cos^2(\delta_{ij} + \theta_i), \\ \sigma_{xy}(i,j) &= \left(\sigma_r^2 - r_{ij}^2(\sigma_\delta^2 + \sigma_\theta^2)\right) \sin(\delta_{ij} + \theta_i) \cos(\delta_{ij} + \theta_i). \end{aligned}$$

Remark 3.4 Since the linear approximation introduced is based on a first order expansion, its validity holds under the assumption of sufficiently small measurement errors.

Remark 3.5 Note that Σ_{ij} is a function of the true values of range, bearing and compass. Since it is not possible to have access to these data, in a real setup these quantities must be replaced by their corresponding measured values.

Once computed the displacements, it is possible to define the weighted residuals as

$$J_d = \frac{1}{2} \sum_{\{i,j\} \in \mathcal{E}_m} \|p_i - p_j - d_{ij}\|^2_{\Sigma_{ij}^{-1}}.$$

Thanks to this, it is possible to define an approximation of the negative log-likelihood in (3.1), which accounts for the GPS measurements and the displacements, as

$$J_{\text{LS}}(\mathbf{p}) = J_p + J_d. \tag{3.9}$$

The minimization problem becomes

$$\widehat{\mathbf{p}}^{\text{LS}} = \text{argmin}_{\mathbf{p}}\ J_{\text{LS}}(\mathbf{p})\ , \tag{3.10}$$

which is a linear least-squares problem, thus convex, which can be solved in closed form. Specifically, assuming $\mathcal{G}_m$ connected, the optimal estimate is given by

$$\widehat{\mathbf{p}}^{\text{LS}} = (\Sigma_{\text{GPS}}^{-1} + A^T\Sigma^{-1}A)^{-1}(\Sigma_{\text{GPS}}^{-1}\mathbf{p}^{\mathbf{GPS}} + A^T\Sigma^{-1}\mathbf{d}), \tag{3.11}$$

where $\Sigma_{\text{GPS}} = \sigma_p^2\mathcal{I}$, Σ is the matrix which accounts for all the Σ_{ij}, and $\mathbf{d}$ and $\mathbf{p}^{\text{GPS}}$ are the vectors obtained stacking together all the relative distances defined in (3.7) and the GPS absolute positions, respectively.

Remark 3.6 Note that the LS estimates only the absolute positions $\mathbf{p}$ without providing any estimate of the absolute orientations. These are retrieved using the compass and exploited to project the noise in rectangular coordinates.

Remark 3.7 Observe that, even if the linear least-squares problem returns an approximate solution for the problem of equation (3.2), since the problem of equation (3.10) is convex, its solution is unique.

For the LS estimator it is possible to show an optimal result similar to the one stated in Lemmas 3.2 and 3.3 for the ML estimator. We state the following:

Lemma 3.8 Consider the cost function (3.9). Then, the optimal solution $\widehat{\mathbf{p}}^{\text{LS}}$ which solves (3.10) is such that

$$\widehat{p}^{\text{LS}}_{\text{ctr.}} = p^{\text{GPS}}_{\text{ctr.}}. \tag{3.12}$$

Moreover, for fixed GPS variance σ_p we have

$$\lim_{\max\{\sigma_\theta,\sigma_r,\sigma_\delta\}\to 0} \widehat{p}_i^{\text{LS}} = p^{\text{GPS}}_{\text{ctr.}} + \Delta p_i^{\text{LS}}\ ,$$

$$\lim_{\min\{\sigma_r,\sigma_\delta\}\to +\infty} \widehat{p}_i^{\text{LS}} = p_i^{\text{GPS}}\ .$$

Proof The result follows with arguments similar to those used in Lemma 3.2 and 3.3. ♢

Observe that, to compute $\widehat{\mathbf{p}}^{\text{LS}}$ as in equation (3.10), one needs all the measurements, their covariances and the topology of $\mathcal{G}_m$ to be available to a central computation unit. In the following section we present a solution which is amenable for a distributed and asynchronous implementation. We assume that a nodes i and j can communicate with each other only if $\{i, j\} \in \mathcal{E}_m$. Remarkably, the solution is robust to packet losses and delays in the communication channel.

3.3 Distributed and Asynchronous Algorithm

In this section we present a distributed and asynchronous solution for the minimization problem (3.10), which is robust to communication delays and packet losses. The implementation presented is inspired by [37], where is shown that this strategy is efficient both in terms of number of iterations and number of sent packets per communication round, compared to existing alternative strategies.
In the following:

(i) by *distributed*, we mean that there is no central unit gathering all the measurements $\mathbf{p}^{\mathrm{GPS}}$ and $\mathbf{d}$, having global knowledge of the graph $\mathcal{G}_m$ and computing $\widehat{\mathbf{p}}^{\mathrm{LS}}$ directly; instead, each node has limited computational and memory resources, and can communicate only with its neighbors;

(ii) by *asynchronous*, we mean that there is no common reference time (generated, e.g., by a centralized clock source) which keeps all the updating/transmitting actions synchronized among all the nodes.

The algorithm we propose is based on a standard gradient descent strategy and employs an *asynchronous broadcast* communication protocol; specifically during each iteration of the algorithm there is only one node which transmits information to all its neighbors in the graph $\mathcal{G}_m$. Furthermore, the time between two consecutive iterations does not have to be constant. We refer to this algorithm as the *asynchronous gradient-based localization* algorithm (denoted hereafter as a-GL algorithm). For the sake of simplicity, from now on, the superscript LS in the single node estimates will be dropped.
We assume that every node has access to its own measurements and the ones of its neighbors nodes, as well as the associated covariances. Additionally we assume that node i, $i \in V$, stores in memory an estimate $\widehat{p}_{\mathrm{i}}$ of p_i and, for $j \in \mathcal{N}_i$, an estimate $\widehat{p}_j^{(i)}$ of $\widehat{p}_j$.
The a-GL algorithm is shown in Algorithm 3. Let $t_0, t_1, t_2, \ldots$ be the time instants in which the iterations of the a-GL algorithm occur.

Algorithm 3 a-GL Algorihtm

```
Require: Node $i \in V$ store in memory the measurements $p_i^{\mathrm{GPS}}$, $d_{ij}$, $j \in \mathcal{N}_i$, the variances $\sigma_p$, $N_{ij}$ and the neighbors estimates $\widehat{p}_j^{(i)}$, $j \in \mathcal{N}_i$.
for $t = t_0, t_1, t_2, \ldots$ do
    # Random node selection
    Node $i \in V$ wakes-up
    # Node $i$ self update
    $\widehat{p}_i \leftarrow \widehat{p}_i - \alpha(i) \odot \frac{\partial J_{\mathrm{LS}}}{\partial p_i}$
    # Self-update broadcasting
    $\widehat{p}_i$ broadcast to $j$, $j \in \mathcal{N}_i$
    # neighbors memory update
    $\widehat{p}_i^{(j)} \leftarrow \widehat{p}_i \, , \forall j \in \mathcal{N}_i$
```

In Algorithm 3, $\alpha(i) = [\alpha_x(i) \ \alpha_y(i)]^T$ is a suitable scale factor for the gradient step. Through standard algebraic computations, one can see that:

$$\frac{\partial J_{\mathrm{LS}}}{\partial p_i} = \frac{p_i - p_i^{\mathrm{GPS}}}{\sigma_p^2} + \sum_{j \in \mathcal{N}_i} \Sigma_{ij}^{-1} (p_i - p_j - d_{ij}) \ .$$

Observe that in order to compute $\frac{\partial J_{\mathrm{LS}}}{\partial p_i}$, node i requires information only from its neighbors. This makes the algorithm amenable for a distributed implementation. Since every node has available in memory a copy of the neighbors estimate, a natural way to evaluate the gradient is

$$\frac{\partial J_{\mathrm{LS}}}{\partial p_i} = \frac{\widehat{p}_i(t) - p_i^{\mathrm{GPS}}}{\sigma_p^2} + \sum_{j \in \mathcal{N}_i} \Sigma_{ij}^{-1} (\widehat{p}_i(t) - \widehat{p}_j^{(i)}(t) - d_{ij}) \ ,$$

It is possible to show that J_{LS} does not increase if

$$0 < \alpha_x(i) \leq \left(\frac{1}{\sigma_p^2} + \sum_{j \in \mathcal{N}_i} (\gamma_x(i,j) + \gamma_x(j,i)) \right)^{-1}, \tag{3.13a}$$

$$0 < \alpha_y(i) \leq \left(\frac{1}{\sigma_p^2} + \sum_{j \in \mathcal{N}_i} (\gamma_y(i,j) + \gamma_y(j,i)) \right)^{-1}, \tag{3.13b}$$

where $\gamma_x(h,k)$ and $\gamma_y(h,k)$ represent the diagonal elements of Σ_{ij}^{-1}. In particular, if $\alpha(i)$ coincides with the RHS of (3.13a) then the minimum of J_{LS} is attained.

In the following we analyze the convergence properties and the robustness to packet losses and delays of the a-GL algorithm.

3.3.1 Convergence Analysis in Presence of Packet Losses and Communication Delays

In Algorithm 3 is presented a way to compute the linear least-squares solution of (3.10) in a distributed and asynchronous fashion. In this section we consider an even more realistic scenario: *presence of delays and packet losses in the communication channel.* Convergence of the a-GL algorithm to the optimal LS solution is proven, provided that the network is uniformly persistent communicating and the transmission delays and the frequencies of communication failures satisfy mild conditions which we formally describe next. We introduce the following definition.

Definition 3.9 (Uniformly persistent comm. network) A network of N nodes is said to be a *uniformly persistent communicating network* if there exists a positive integer number τ such that, for all $t \in \mathbb{N}$, each node perform lines 5 and 7 of the a-GL algorithm at least once within the iteration-interval $[t, t+\tau)$.

Moreover, the following assumptions characterize the communication non-idealities.

Assumption 3.10 (Bounded packet losses) There exists a positive integer L such that the number of consecutive communication failures between every pair of neighboring nodes in the graph $\mathcal{G}_m$ is less than L.

Assumption 3.11 (Bounded delays) Assume node i broadcasts its estimate to its neighbors during iteration t, and, assume that, the communication link (i,j) does not fail. Then, there exists a positive integer D such that the information $\widehat{p}_i(t+1)$ is used by node j to perform its local update not later than iteration $t+D$.

Loosely speaking, Assumption 3.10 implies that there can be no more than L consecutive packet losses between any pair of nodes i, j belonging to the communication graph. Differently, Assumption 3.11 considers the scenario where the received packets are not used instantaneously, but are subject to some delay no greater than D iterations.

The following result characterizes the convergence properties of the a-GL algorithm in the scenario described by Definition 3.9 under Assumptions 3.10 and 3.11.

Proposition 3.12 (Proposition V.3 in [59]) Consider a *uniformly persistent communicating network* of N nodes running the a-GL algorithm over a connected measurement graph $\mathcal{G}_m$. Let Assumptions 3.10 and 3.11 be satisfied. Assume the weights $\alpha(i)$ satisfy Equations (3.13a)–(3.13b). Moreover, assume that $\widehat{p}_i$, $i \in \{1, \ldots, N\}$, $\widehat{p}_j^{(i)}$, $j \in \mathcal{N}_i$, be initialized to $\mathbf{p}^{\mathrm{GPS}}$. Then the following facts hold true

(i) the evolution $t \to \widehat{\mathbf{p}}(t)$ asymptotically converges to the optimal estimate $\widehat{\mathbf{p}}^{\mathrm{LS}}$, i.e.,

$$\lim_{t\to\infty} \widehat{\mathbf{p}}(t) = \widehat{\mathbf{p}}^{\mathrm{LS}};$$

(ii) the convergence is exponential, namely, there exists $C > 0$ and $0 \leq \rho < 1$ such that

$$\|\widehat{\mathbf{p}}(t) - \widehat{\mathbf{p}}^{\mathrm{LS}}\| \leq C\rho^t \|\widehat{\mathbf{p}}(0) - \widehat{\mathbf{p}}^{\mathrm{LS}}\|. \tag{3.14}$$

Proof The proof can be found in [59]. ♢

3.4 Simulations

In this section, we test the effectiveness of the proposed algorithm. We consider a group of robots:

- placed on a $2D$ lattice formation;
- regularly spread with an inter-node distance of 4 meters.

We assume each agent to be endowed with:

- a GPS sensor characterized, according to [53],[54], by $\sigma_p = 2$ [m];
- a compass sensor characterized by, according to [55], by $\sigma_\theta = 0.05$ [rad];
- a range and a bearing sensors with standard deviations σ_r and σ_δ, respectively. Acceptable values are $\sigma_r = 0.1$ [m] and $\sigma_\delta = 0.03$ [rad]. However, due to their variability, we test our algorithm in a sufficiently wide range of standard deviation values.

The reminder of the section is organized as follows:

(i) in Section 3.4.1, we briefly describe the performance measures used, later on, to test our algorithms;

(ii) in Section 3.4.2, we analyze the steady state behavior of the a-GL algorithm with respect to the ground truth, for increasing number of nodes N and for different values of σ_r and σ_δ;

(iii) in Section 3.4.3, we analyze the transient behavior (convergence analysis) of the a-GL algorithm in terms of number of iterations with respect to the optimal configuration obtained from (3.11).

3.4.1 Performance Measures

For the steady state analysis of Section 3.4.2, the estimated positions are compared with the ground truth in terms of Mean Squared Error (MSE). Specifically, by denoting the generic vector of positions estimate $\widehat{\mathbf{p}} = [\widehat{p}_1, \ldots, \widehat{p}_N]^T$ where $\widehat{p}_i = (\widehat{x}_i, \widehat{y}_i)$, the MSE of the positions is equal to

$$MSE(\widehat{\mathbf{p}}, \mathbf{p}) = \mathbb{E}\left[\|\widehat{\mathbf{p}} - \mathbf{p}\|^2\right]. \tag{3.15}$$

By defining the centroids of the estimated x and y coordinates as

$$\widehat{x}_{\text{ctr.}} := \frac{1}{N}\sum_{i=1}^{N}\widehat{x}_i \ , \qquad \widehat{y}_{\text{ctr.}} := \frac{1}{N}\sum_{i=1}^{N}\widehat{y}_i \ ,$$

and those of the true x and y coordinates as

$$x_{\text{ctr.}} := \frac{1}{N}\sum_{i=1}^{N}x_i \ , \qquad y_{\text{ctr.}} := \frac{1}{N}\sum_{i=1}^{N}y_i \ ,$$

the MSE can be rewritten as

$$MSE(\widehat{\mathbf{p}},\mathbf{p}) = \mathbb{E}\Big[\sum_{i=1}^{N}\big((\widehat{x}_i - \widehat{x}_{\text{ctr.}}) - (x_i - x_{\text{ctr.}}) + (\widehat{x}_{\text{ctr.}} - x_{\text{ctr.}})\big)^2 + \\ \big((\widehat{y}_i - \widehat{y}_{\text{ctr.}}) - (y_i - y_{\text{ctr.}}) + (\widehat{y}_{\text{ctr.}} - y_{\text{ctr.}})\big)^2\Big] \ .$$

It is convenient to define the displacements from the centroid and the difference between the centroids for the x coordinate as

$$\Delta x_i := x_i - x_{\text{ctr.}}, \qquad \Delta\widehat{x}_i := \widehat{x}_i - \widehat{x}_{\text{ctr.}}, \qquad \Delta x_{\text{ctr.}} := \widehat{x}_{\text{ctr.}} - x_{\text{ctr.}} \ ,$$

and similarly Δy_i, $\Delta\widehat{y}_i$ and Δy_C those for the y coordinate. We recall the fact that

$$\sum_{i=1}^{N}\Delta x_i = \sum_{i=1}^{N}\Delta y_i = \sum_{i=1}^{N}\Delta\widehat{x}_i = \sum_{i=1}^{N}\Delta\widehat{y}_i = 0 \ .$$

After some algebraic manipulations it is possible to write

$$MSE(\widehat{\mathbf{p}},\mathbf{p}) = MSE_{\text{Ctr.}} + MSE_{\text{Rel.Disp.}} \ ,$$

where

$$MSE_{\text{Ctr.}} := \mathbb{E}\left[\Delta x_C^2 + \Delta y_C^2\right] \ , \tag{3.16a}$$

$$MSE_{\text{Rel.Disp.}} := \mathbb{E}\left[\sum_{i=1}^{N}(\Delta\widehat{x}_i - \Delta x_i)^2 + (\Delta\widehat{y}_i - \Delta y_i)^2\right] \ , \tag{3.16b}$$

represent the MSE of the centroids and of the relative displacement from the centroid, respectively. Note that the

$$MSE_{\text{Ctr.}} = \frac{\sigma_p^2}{N} \ ,$$

so, it scales with the number of nodes and tends to zero as $N \to \infty$.

For the transient analysis of Section 3.4.3, we compare the performance of the a-GL algorithm with the steady state estimate obtained with the LS centralized algorithm, i.e.,

$$\|\widehat{\mathbf{p}}(t) - \widehat{\mathbf{p}}^{\text{LS}}\|. \tag{3.17}$$

As shown in equation (3.14), the a-GL exponentially converges to the centralized solution.

Remark 3.13 (Numeric MSE) Observe that the theoretic MSE cannot be exactly computed. In the following, we plot the numeric MSE computed via Monte Carlo simulations.

Remark 3.14 (Dependence between σ_r and σ_δ) In the following we test the proposed algorithm as a function of the relative measurements standard deviations, σ_r and σ_δ. We vary only the range standard deviation since the bearing measurements accuracy is assumed to depend on the range accuracy as $\sigma_\delta = atan2(\sigma_r, \frac{4}{3})$ which let us approximately draw samples in a ball centered in the true positions.

3.4.2 Steady State Analysis

In this section, we analyze the steady state behavior of the a-GL algorithm for increasing N and for different values of σ_r and σ_δ.
Figure 3.1 shows the absolute positions of the GPS measurements, the a-GL estimate and the minimizer of the log-likelihood, respectively. It can be seen how, thanks to the additional relative information, the estimates outperforms the GPS measurements.

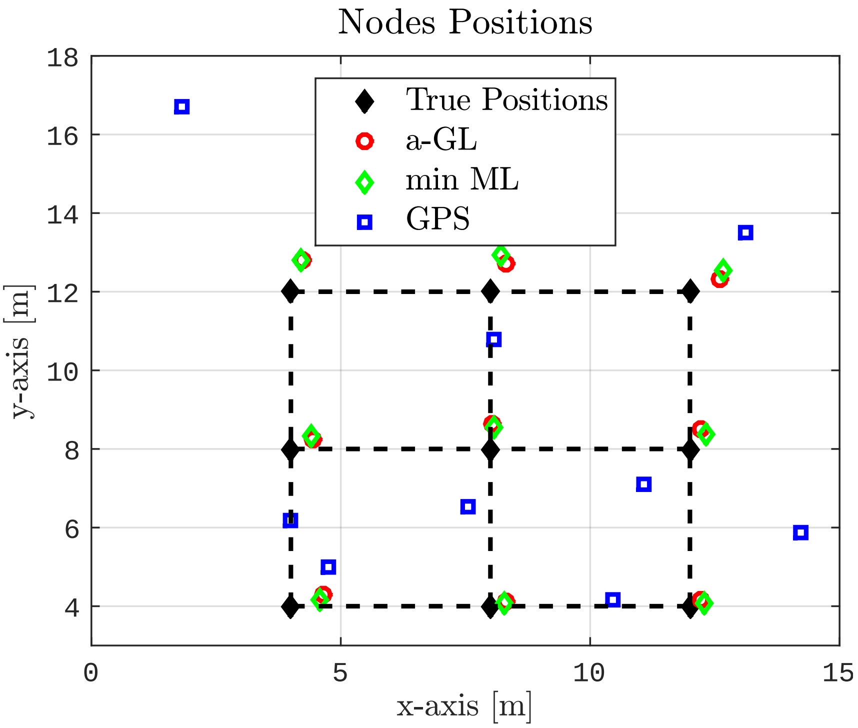


Figure 3.1: Absolute positions for a formation of robots with $N = 9$, $\sigma_r = 0.1$ [m] and $\sigma_\delta = 0.03$ [rad]. The black dashed line highlights the shape of the real formation.

Remark 3.15 As outlined, the ML estimation problem is highly non-linear and characterized by many local minima. Then, the ML estimate has been computed by exhaustive search around the ground truth.

Figure 3.2 shows the behavior of the MSE of equation (3.15) for increasing N. Specifically, the MSE has been split into its components related to the centroid and the relative displacement, equations (3.16a)–(3.16b), respectively. It can be seen how the $MSE_{\text{Ctr.}}$ tends to zero for $N \to \infty$, while $MSE_{\text{Rel.Disp.}}$ remains almost constant and on the same order of magnitude of σ_r^2. From the plot, it can be understood that there are mainly two sources of error: one related to the absolute position reconstruction, which is obtained from the GPS information; one depending on the relative information. Thanks to accurate relative information, it is possible to reconstruct the shape of the formation with an error comparable to that of the relative measurements. The absolute formation position, which is recovered from the GPS, for small number of agents is the greater source of error, but improves with the number of robots as $\frac{1}{N}$. As already outlined, the proposed solution can be used seamlessly in scenario where not all the robots are equipped with GPS sensors. In this case, the absolute positions error scales with the number of agents equipped with a GPS module.

Figure 3.3 shows the absolute positions MSE, equation (3.15), for increasing values of σ_r. The plot shows the behavior of the a-GL algorithm (red line) compared with the behavior of the maximum likelihood estimator (green line). Moreover, some limit behaviors are plotted: the MSE

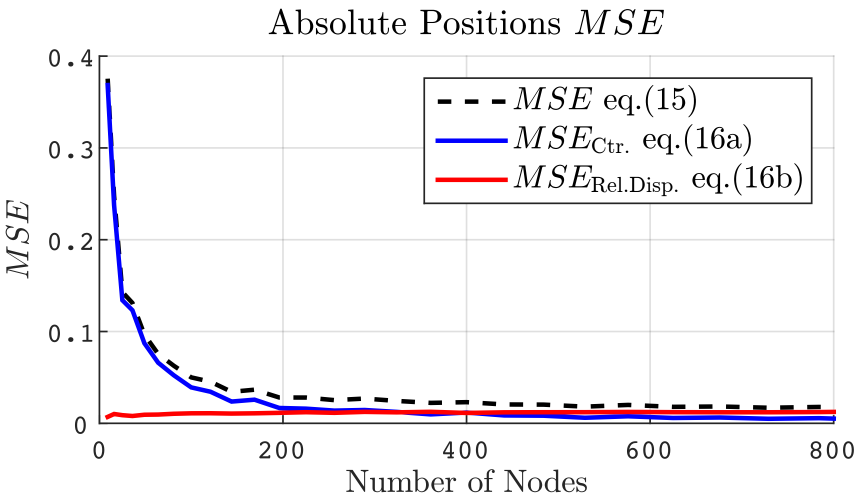


Figure 3.2: Absolute positions MSE, computed via Monte Carlo simulations, as function of the number of nodes for $\sigma_r = 0.1$ [m] and $\sigma_\delta = 0.03$ [rad].

of the GPS measurements (blue dashed line); the MSE of the mean of the GPS measurements (black dashed line). The limit behaviors, according to Lemma 3.8, are due to the following facts:

- for increasing values of σ_r the relative sensors information becomes useless and the estimator will "trust" mainly the GPS measurements;
- for small values of σ_r the shape of the formation is "perfectly" known. So, the only source of error is due to the displacement of the GPS mean from the ground truth mean.

Figure 3.3 shows how the a-GL algorithm behaves similarly to the ML estimator for the whole range of σ_r. In addition to this, for values of σ_r within $0.1 \div 0.5$ [m], which characterize practical operating sensors range, the a-GL algorithm mimics almost perfectly the ML estimator.

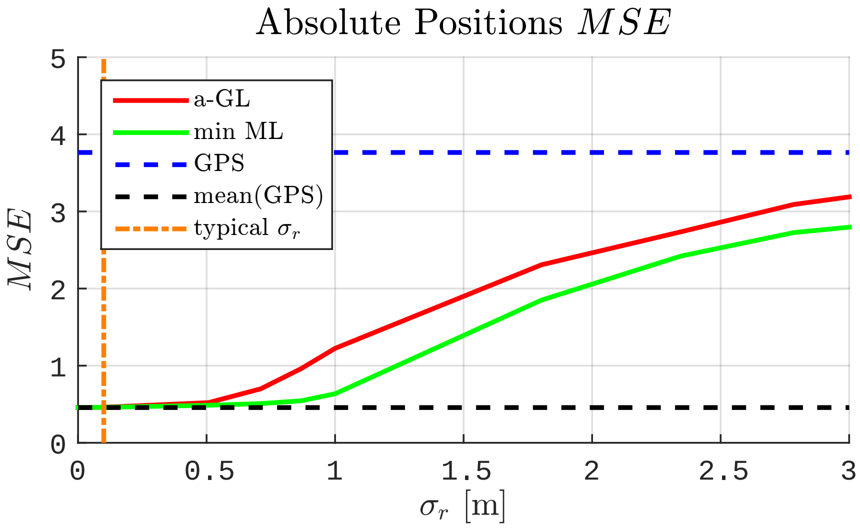


Figure 3.3: Absolute positions MSE, computed via Monte Carlo simulations, as function of σ_r $\left(\sigma_\delta = atan2(\sigma_r, \frac{4}{3})\right)$ for $N = 9$. The dark orange vertical dashed-dotted line highlights the behavior corresponding to $\sigma_r = 0.1$ [m].

3.4.3 Transient Analysis

In this section we analyze the transient behavior of the a-GL algorithm in presence of packet losses and communication delays. At each iteration, a node, randomly chosen, wakes up, updates its state and communicates its estimate to node $j \in \mathcal{N}_i$. We assume independent communication links between neighboring nodes, each of them characterized by a certain failure probability. Figure 3.4 plots, in logarithmic scale, the error in (3.17) between the a-GL estimated formation and the optimal LS solution, computed using (3.11). The different lines correspond to different percentages of packet losses. As expected, the higher is the losses the slower is the convergence. Note that, in a real set-up, different nodes could wake up and update their estimates at the same time. This could increase the possibility of communication collision but at the same time could speed up the convergence rate.

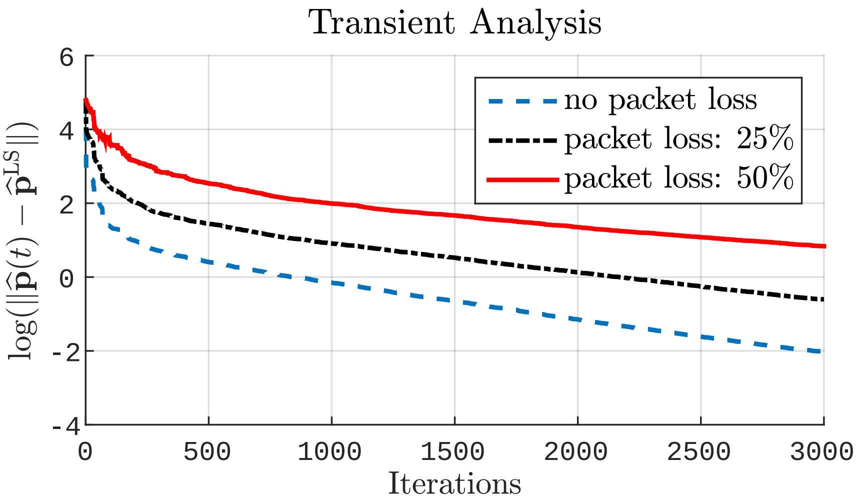


Figure 3.4: Comparison between the a-GL solution and the optimal centralize LS solution, for different percentages of packet losses.

4 Coverage

In this chapter we analyze the framework of coverage the area of interest while estimating the non-uniform measurable field of event appearance from noisy measurements collected by the robots. We consider a realistic scenario where communication is performed over an unreliable wireless network subject to packet losses or delays. In particular we start from the Voronoi optimization problem, which can be solved using the Lloyd algorithm, however it requires the knowledge of a sensory function that it may not be known in advance. Under a client-server communication scheme we present an algorithm capable of estimating the unknown sensory function and perform the coverage in a complete robust fashion. The estimate is proved to converge to the real sensory function and the final configuration of the robots can be arbitrarily close to an optimal partitioning that would be obtained if the sensory function would be known since the beginning. The estimation is performed through the classical Gaussian regression and to overcome its limitation on the computational complexity we also present an approximated solution based on a grid. This approach is shown to be much more efficient and also guarantees the convergence of the estimated sensory function to the real one in all the points of the grid. The chapter is organized as follows. In Section 4.1 some mathematical preliminaries and notions on Voronoi partitioning and coverage control are recalled. In Section 4.2 we formulate the problem at hand, and in Section 4.3 we briefly review Gaussian non-parametric estimation. Section 4.4 presents the main algorithm and its convergence analysis. In Section 4.5 we discuss numerical considerations and we present an approximated algorithm and its properties. In Section 4.6 we test the proposed algorithms via extensive numerical simulations.

4.1 Mathematical Preliminaries

In this section we recall some mathematical preliminaries which will be useful later on.

4.1.1 Bernoulli Trial

In probability theory a *Bernoulli trial* is a random experiment with exactly two possible outcomes, “success” and “failure” , in which the probability of success is the same every time the experiment is conducted [60]. More formally, given a value $\overline{\eta} \in \ [0,1]$, then a Bernoulli trial $\eta \sim B(\overline{\eta}), \eta \in \{0,1\}$ is defined as

$$\eta = \begin{cases} 1 \text{ with probability } \overline{\eta} \ , \\ 0 \text{ with probability } 1 - \overline{\eta}. \end{cases}$$

4.1.2 Voronoi Partitions

Let $\mathcal{X}$ be a compact and convex polygon in $\mathbb{R}^2$ an let $\|\cdot\|$ denote the Euclidean distance function. Let $\mu : \mathcal{X} \to \mathbb{R}_{>0}$ be a distribution sensory function defined over $\mathcal{X}$. Within the context of this paper, a *partition* of $\mathcal{X}$ is a collection of N convex polygons $\mathcal{P} = (\mathcal{P}_1, \dots, \mathcal{P}_N)$ with disjoint interiors whose union is $\mathcal{X}$. Given the list of N distinct points in $\mathcal{X}$, $\mathbf{x} = (x_1, \dots, x_N)$, we define the *Voronoi partition* $\mathcal{W}(\mathbf{x}) = \{\mathcal{W}_1(\mathbf{x}), \dots, \mathcal{W}_N(\mathbf{x})\}$ generated by $\mathbf{x}$ as

$$\mathcal{W}_i(\mathbf{x}) = \{q \in \mathcal{X} \mid \|q - x_i\| \leq \|q - x_j\|, \ \forall j \neq i\} . \tag{4.1}$$

which can be shown to be convex [61]. Given a partition $\mathcal{P} = (\mathcal{P}_1, \dots, \mathcal{P}_N)$, for each region $\mathcal{P}_i$, $i \in \{1, \dots, N\}$, we define its *centroid* with respect to the sensory function μ as

$$c_i(\mathcal{P}_i) = \left(\int_{\mathcal{P}_i} \mu(q) dq \right)^{-1} \int_{\mathcal{P}_i} q\mu(q) dq.$$

We denote by

$$\mathbf{c}(\mathcal{P}) = (c_1(\mathcal{P}_1), \dots, c_N(\mathcal{P}_N))$$

the vector of regions centroids. A partition $\mathcal{P} = (\mathcal{P}_1, \dots, \mathcal{P}_N)$ is said to be a *Centroidal Voronoi partition* of the pair $(\mathcal{X}, \mu)$ if

$$\mathcal{P} = \mathcal{W}(\mathbf{c}(\mathcal{P})) ,$$

namely if $\mathcal{P}$ coincides with the Voronoi region generated by $\mathbf{c}(\mathcal{P})$.
Given a partition $\mathcal{P} = \{\mathcal{P}_1, \dots, \mathcal{P}_N\}$ and a sensory function μ we introduce the *Coverage function* $H(\mathcal{P}; \mu)$ defined as

$$H(\mathcal{P}; \mu) = \sum_{i=1}^{N} \int_{\mathcal{P}_i} \|q - c_i(\mathcal{P}_i)\|^2 \mu(q) dq .$$

For a fixed sensory function μ, it can be shown that the set of local minima of H coincides with the Centroidal Voronoi partitions of the pair $(\mathcal{X}, \mu)$ [61].

4.1.3 Coverage Control Algorithm

Let $\mathcal{X}$ be a convex and closed polygon in $\mathbb{R}^2$ and let μ be a sensory function defined over $\mathcal{X}$. Consider the following optimization problem

$$\min_{\mathcal{P}} \ H(\mathcal{P}; \mu).$$

The *coverage algorithm* we consider is a version of the classic Lloyd algorithm [62] based on "centering and partitioning" for the computation of Centroidal Voronoi partitions. Given an initial condition $\mathcal{P}(0)$ the algorithm cycles iteratively the following two steps:

(i) computes the centroids of the current partition, namely $\mathbf{c}(\mathcal{P})$;

(ii) updates $\mathcal{P}$ to the partition $\mathcal{W}(\mathbf{c}(\mathcal{P}))$.

In mathematical terms, for $k \in \mathbb{N}$, the algorithm is described as

$$\mathcal{P}^L(k+1) = \mathcal{W}(\mathbf{c}(\mathcal{P}^L(k))) , \tag{4.2}$$

where the upperscript L indicates the sequence generated by the Lloyd algorithm. Clearly, by construction $\mathcal{P}^L(k)$ are all Voronoi partitions for $k \geq 1$. It can be shown, [63], that the function $H(\mathcal{P}; \mu)$ is monotonically non-increasing along the solutions of (4.2) and that all the solutions of (4.2) converge asymptotically to the set of centroidal Voronoi partitions. It is well known, [63], that the set of centroidal Voronoi partitions of the pair $(\mathcal{X}, \mu)$ are the critical points of the coverage function $H(\mathcal{P}; \mu)$.

4.2 Problem Formulation

We consider a group of N robots, all with sensing capabilities and communicating with a Base Station (BS). They are displaced and allowed to move in an area represented by the convex set $\mathcal{X}$. Their goal is to simultaneously estimate an unknown map $\mu : \mathcal{X} \to \mathbb{R}$ and to provide a good partitioning $\mathcal{P}$ for minimizing $H(\mathcal{P};\mu)$. To be more precise μ is modelled as the realization of a zero-mean Gaussian random field with covariance $K:\mathcal{X}\times\mathcal{X}\to\mathbb{R}$. Without loss of generality, we restrict our attention to radial Mercer kernels, i.e. $K(a,b)=h(\| a-b \|)$, such that if $\| a-b \|\leq\| c-d \|$ then $h(\| a-b \|)\leq h(\| c-d \|)$ and $K(x,x)=\lambda$, $\forall x\in\mathcal{X}$. The function μ is assumed to be unknown to both the robots and the BS. Each agent $i\in\{1,\dots,N\}$ is required to have the following basic computation, communication and sensing capabilities:

(C1) it can identify itself to the BS and can send information to the BS, however the transmission can fail with probability $1-\overline{\beta}$ with $\overline{\beta}>0$;

(C2) it can measure the function μ in the position it occupies; specifically, if x_i denotes its current position, it can take the noisy measurement

$$y(x_i)=\mu(x_i)+\nu_i,$$

where $\nu_i \backsim \mathbb{N}(0,\sigma^2)$ is independent of the unknown function μ and of all other measurement noises ν_j;

(C3) it can move to a target-point b_i transmitted by the BS.

The BS must have the following capabilities:

(C4) it can store all the measurements taken by all the robots;

(C5) it can store the last position of all the robots;

(C6) it can compute centroids and Voronoi regions of all the robots;

(C7) it can send information periodically to all robots every T seconds;

(C8) it can send information to each robot i, however the transmission can fail with probability $1-\overline{\gamma}$ with $\overline{\gamma}>0$;

(C9) it can compute and store an estimate $\widehat{\mu}$ of the function μ and its posterior variance V.

We also assume a simple robots' dynamics with the following discrete update law:

$$x_{i,k+1}=x_{i,k}+u_{i,k},\quad \forall i\in\{1,\dots,N\}, \tag{4.3}$$

where $x_{i,k}=x_i(kT)$, i.e., each robot can move from location $x_{i,k}$ at time $t=kT$ to any desired location $x_{i,d}=x_{i,k+1}$ at time $t=(k+1)T$. The ultimate goal is to position the robots in the centroids of a good partition that minimizes $H(\mathcal{P};\mu)$. This is a challenging problem since we have to deal with the well known *exploration-exploitation dilemma*:

(i) to have a good partition with respect to μ we need a good estimate of μ which implies having the robots moving around the environment in order to explore it;

(ii) as mentioned in the introduction, to minimize the expected weighted time to monitor a new event, it is necessary to position the robots in the centroids of a specific partition and to keep them idle.

Intuitively, a good strategy should initially promote exploration and later, as the estimate $\widehat{\mu}$ of the true map μ becomes more accurate, the robots should transition to exploitation. As shown later, the strategy that we propose smoothly transitions from the exploration phase to the exploitation phase as suggested by this intuition. We address this problem in the context of a *client-server architecture*, see Figure 4.1, under a communication model which takes into account possible packet losses on both ways, i.e. from robots to BS and BS to robots.

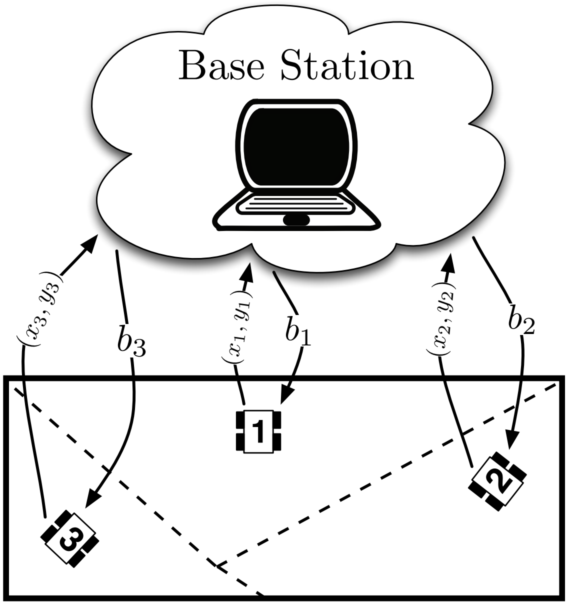


Figure 4.1: The figure illustrates the setup used in the simulations.

4.3 Function Estimation and Posterior Variance Computation

We assume that each robot takes only one measurement within the time window $(kT, (k+1)T)$ we indicate this measurement as $y_{i,k} = y(x_{i,k})$. Once the measurement is taken, it is immediately transmitted to the BS. We also define with $\beta_{i,k} \in \{0, 1\}$ and $\gamma_{i,k} \in \{0, 1\}$ the variables that indicate whether the k-th message from the i-th robot to the BS, and k-th message from the BS to the i-th robot have been received, respectively, i.e., $\beta_{i,k} = 1$ (resp. $\gamma_{i,k} = 1$) if the k-th message from the i-th robot to the BS (resp. from the BS to the i-th robot) has been received successfully and $\beta_{i,k} = 0$ (resp. $\gamma_{i,k} = 0$) otherwise. We define with J_k the set of measurements successfully received by the BS, i.e

$$J_k := \{(x_{i,k}, y_{i,k}) \,|\, \beta_{i,k} = 1, i = 1, \dots, N\},$$

and the complete information set I_k available at the base station as follows:

$$I_k = I_{k-1} \cup J_k, \;\; \forall k \geq 1,$$

where $I_0 = \emptyset$.
Let m_k be the cardinality of the set I_k, i.e., $m_k = |I_k|$, and, for notational convenience, let us relabel I_k as

$$I_k = \left\{ \left(x^{(h)}, y^{(h)} \right) \mid h \in \{1, \dots, m_k\} \right\},$$

where $\left(x^{(h)}, y^{(h)}\right) = (x_{i,t}, y_{i,t})$ for some $i \in \{1, \dots, N\}$ and some $t \in \{1, \dots, k\}$ such that $\beta_{i,t} = 1$. As shown by [64] and [65], the minimum variance estimate of μ given I_k can be computed as:

$$\widehat{\mu}_k(x) = \mathbb{E}\left[\mu(x)|I_k\right] = \sum_{i=1}^{m_k} c^{(i)}\, K(x^{(i)}, x), \quad x \in \mathcal{X}, \tag{4.4}$$

where

$$\begin{bmatrix} c^{(1)} \\ \vdots \\ c^{(m_k)} \end{bmatrix} = (\bar{K}_k + \sigma^2 \mathbb{I})^{-1} \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m_k)} \end{bmatrix},$$

and

$$\bar{K}_k = \begin{bmatrix} K(x^{(1)}, x^{(1)}) & \dots & K(x^{(1)}, x^{(m_k)}) \\ \vdots & & \vdots \\ K(x^{(m_k)}, x^{(1)}) & \dots & K(x^{(m_k)}, x^{(m_k)}) \end{bmatrix}. \tag{4.5}$$

The a posteriori variance of the estimate, in a generic input location $x \in \mathcal{X}$, is

$$V_k(x) = \mathrm{Var}\left[\mu(x)|I_k\right] = K(x,x) - \\ \begin{bmatrix} K(x^{(1)}, x) & \dots & K(x^{(m_k)}, x) \end{bmatrix} (\bar{K}_k + \sigma^2 \mathbb{I})^{-1} \begin{bmatrix} K(x^{(1)}, x) \\ \vdots \\ K(x^{(m_k)}, x) \end{bmatrix}. \tag{4.6}$$

4.4 r-EC Algorithm

To achieve the goal described in Section 4.2, we propose the *robust Estimation and Coverage algorithm* described in Algorithm 4 (denoted hereafter as r-EC algorithm). The algorithm is divided in two parts, the first is the code executed by the BS and the second is the code executed by the robots.

For each $k \in \mathbb{N}$ the BS stores in memory the estimate $\widehat{\mu}_k(x)$ of the function $\mu(x)$ and its posterior $V_k(x)$, a partition $\mathcal{P}_k = \{\mathcal{P}_{1,k}, \dots, \mathcal{P}_{N,k}\}$, the corresponding list of centroids $\widehat{\mathbf{c}}_k = \{\widehat{c}_{1,k}, \dots, \widehat{c}_{N,k}\}$, the positions of the robots $(x_{1,k}, \dots, x_{N,k})$ and all the measurements received up to k by the robots, I_k. The BS collects the new positions $x_{i,k}$ and the sensed function μ in $x_{i,k}$, i.e. $y_{i,k} = \mu(x_{i,k}) + \nu_{i,k}$, for all robots i who transmit successfully. The BS, based on the newly gathered measurements and on the past measurements, computes a new estimate $\widehat{\mu}_k$ of μ exploiting the technique explained in Section 4.3. Additionally, the BS updates the partition $\mathcal{P}_k$, setting $\mathcal{P}_k = \mathcal{W}(\widehat{\mathbf{c}}_{k-1})$ and the corresponding new centroids $\widehat{\mathbf{c}}_k$. It is quite intuitive that in order to improve the quality of the estimate of the function μ, the measurements have to be taken to reduce as much as possible the posterior variance $V_k(x)$. To do so the r-EC algorithm uses a monotonically increasing function $F(M_{i,k})$, as for example $F(x) = x$, where $M_{i,k}$ is the maximum of the posterior in the Voronoi region of the agent i at time k. The algorithm decides whether a robot has to perform exploration or to move to the centroid, using a randomized strategy based on the value $F(M_{i,k})$, i.e. the higher the value of $F(M_{i,k})$, the higher is the probability to perform an exploration step, and vice-versa. If an agent i is selected to perform exploration, the BS identifies the next target point for this agent by determining the point with maximum posterior variance in its current patrolling region $\mathcal{P}_{i,k}$. The target points are then sent by the BS to each robot every period T. The communication between the BS to a single robot is not realiable and packets can be lost. If the agent receives the target point, then it moves there to take the next sample measurement, otherwise it takes no measurement and remains still till the next target point arrives, therefore robots do not need to be synchronized with the BS.

4.4.1 Convergence Analysis

In this section we provide the mathematical characterisation of the proposed r-EC algorithm. The following proposition states that, if the packet loss probability is smaller than unity and under a mild condition on the threshold policy $F(\cdot)$ which determines whether an agent has to perform either an exploitation or an exploration step, the estimated map $\widehat{\mu}_k$ converges to the true map μ.

Proposition 4.1 Let us consider the r-EC algorithm. Let $F(M) : [0,1] \to [0,1]$ be a continuous and monotonically increasing function such that $F(M) > 0$ for $M > 0$. If $\mathbb{P}[\beta_{i,k} = 1] = \overline{\beta} > 0$ and $\mathbb{P}[\gamma_{i,k} = 1] = \overline{\gamma} > 0$. Then

$$\widehat{\mu}_k \xrightarrow{P} \mu,$$

where the convergence is in the space of the continuous functions (sup-norm).

Algorithm 4 r-EC Algorithm

```
BASE STATION
Require: The BS stores in memory $\widehat{\mu}_k$,$V_k(x), I_k$, and has a clock that triggers an event every $T$
    seconds.
if $k = nT,\ n \in \mathbb{N}$ then
    Listen:

    for i=1,...,N do
        if BS $\leftarrow (x_{i,k}, y_{i,k})$ then
            $I_k = I_k \cup \{x_{i,k}, y_{i,k}\}$

    Estimate update:
    $\mathcal{P}_{i,k} = \mathcal{W}_i(\widehat{\mathbf{c}}_{k-1})$,                                    eq.(4.1)

    $\widehat{\mu}_k(x) = \mathbb{E}\,[\mu(x) \,|\, I_k]$,                                    eq.(4.4)

    $V_k(x) = \mathrm{Var}\,[\mu(x) \,|\, I_k]$,                                    eq.(4.6)

    $\widehat{c}_{i,k} = (\int_{\mathcal{P}_{i,k}} \widehat{\mu}_k(q)dq)^{-1} \int_{\mathcal{P}_{i,k}} q\widehat{\mu}_k(q)dq.$

    Target-Points computation:

    $M_{i,k} = \frac{\max_{x \in \mathcal{P}_{i,k}} V_k(x)}{\lambda},\ \forall i$

    $p_{i,k} = F(M_{i,k})$

    $\eta_{i,k} \sim \mathcal{B}(p_{i,k})$
    if $\eta_{i,k} = 1$ then
        $b_{i,k} = \mathrm{argmax}_{x \in \mathcal{P}_{i,k}} V_k(x)$
    else
        $b_{i,k} = \widehat{c}_{i,k}$

    Target-Points Transmission:

    $x_{i,k+1} = b_{i,k}$ (i.e. $u_{i,k} = b_{i,k} - x_{i,k}$), $\forall i$


ROBOTS
Require: A clock with sample time $T$ or a submultiple of $T$.
if $k = nT,\ n \in \mathbb{N}$ then
    Measurement collection: $y_{i,k} = \mu(x_{i,k}) + v_{i,k}$

    Measurement transmission: $(x_{i,k}, y_{i,k}) \longrightarrow$ BS

    Listen:
    if BS $\rightarrow b_i$ then
        $x_{i,k+1} = b_i$ (alternatively $u_{i,k} = b_{i,k} - x_{i,k}$)

    Move to the new target-point.
```

Proof Without loss of generality, a system with only one agent is considered. For every $\epsilon > 0$, define the process $\bar{x}_k$ as follows:

$$\bar{x}_k = \begin{cases} x_k & \text{if } M_k \geq \epsilon \\ C_k & \text{with probability } (1-p_\epsilon)\overline{\beta} \quad \text{if } M_k < \epsilon \\ E_k & \text{with probability } p_\epsilon\overline{\beta} \quad \text{if } M_k < \epsilon \\ \bar{x}_{k-1} & \text{with probability } 1-\overline{\beta} \quad \text{if } M_k < \epsilon \end{cases} \tag{4.7}$$

where $p_\epsilon = \min_{x\in[\epsilon,1]} F(x)$, $\overline{\beta}$ is the probability that an agent transmits successfully a packet to the base station, C_k is the location of the centroid and E_k is the location where the posterior variance attains its maximum. We also define $\bar{I}_k$ as the set of measurements collected by the process $\bar{x}_k$ up to instant k, $\bar{V}_k(x) = \text{Var}\left[\mu(x)|\bar{I}_k\right]$ the posterior variance at the input location x associated to the process $\bar{x}_k$ and $\bar{M}_k$ the maximum of $\bar{V}_k$ w.r.t. x. First, we prove that $\forall\epsilon \geq 0, \forall\delta \in (0,1]$ there exists k_0 such that, $\forall k \geq k_0$, one has:

$$\mathbb{P}\left[M_k \leq \epsilon\right] \geq 1-\delta.$$

Note that one always has

$$\mathbb{P}\left[\bar{M}_k \leq \epsilon\right] = \mathbb{P}\left[M_k \leq \epsilon\right], \tag{4.8}$$

since the two processes x_k and $\bar{x}_k$ coincide up to time instant k and, after k, both M_k and $\bar{M}_k$ are less than ϵ due to definition (4.7). In view of (4.8), our strategy is to prove that $\bar{x}_k$ satisfies the desired condition which then will immediately extend to x_k. As also clear in the sequel, the key advantage of using $\bar{x}_k$, in place of x_k, is that it avoids the introduction of conditional probability measures difficult both to define and to handle.

Now, consider the subsequence $\bar{x}_{k_j}$ extracted by $\bar{x}_k$ such that $k_i < k_{i+1}$for every i and for every k_j the agent is moving to E_{k_j}. The length of this subsequence is infinite with probability one since $\bar{x}_k$ can move to the the maximum posterior variance location with probability at least $p_\epsilon\overline{\beta}$ at every k. It is an elementary algebraic fact that, for every $\epsilon > 0$, there exists a pair $\bar{\alpha}$ and $\bar{m}$ such that:

$$\lambda - \frac{(\lambda-\bar{\alpha})^2}{\lambda + \frac{\sigma^2}{\bar{m}}} \leq \epsilon.$$

By the continuity of the kernel, there exists a finite partition, function of $\epsilon, \bar{\alpha}, \bar{m}$, given by all the subsets $\mathcal{D}_j \subseteq \mathcal{X}$ such that $K(x,x^*) \geq \lambda - \bar{\alpha}$, $\forall x, x^* \in \mathcal{D}_j$ (recall that $K(x,x) = \lambda$). Since there is a finite number of subsets $\mathcal{D}_j$, at least one of them is visited infinite times by the subsequence $\bar{x}_{k_j}$ with probability one. This implies that, with probability not smaller than $1-\delta$, there always exists a time instant k_a such that $\mathcal{D}_j$ has been visited at least $\bar{m}$ times and another instant $k_b > k_a$ where $\mathcal{D}_j$ is visited again. Now it is not restrictive consider only $\bar{m}$ measurements falling in $\mathcal{D}_j$, denoted by $z_1^j, \dots, z_{\bar{m}}^j$ and collected on the input locations $\bar{x}_1^j, \dots, \bar{x}_{\bar{m}}^j$. Let $\bar{K}_j$ be the $\bar{m} \times \bar{m}$ kernel matrix with (k,i) entry

$$[\bar{K}_j]_{ki} = K(\bar{x}_k^j, \bar{x}_i^j),$$

i.e. obtained sampling the kernel K on the input locations falling in $\mathcal{D}_j$. We have $Tr(\bar{K}_j) = \sum \Lambda(\bar{K}_j) = m\lambda$, where $\Lambda(\bar{K}_j)$ is the set of real and non negative eigenvalues of $\bar{K}_j$. Then, one has $\bar{K}_j \preceq \bar{m}\lambda\mathbb{I}$ so that

$$(\bar{K}_j + \sigma^2) \preceq (\bar{m}\lambda + \sigma^2)\mathbb{I} \Rightarrow (\bar{K}_j + \sigma^2)^{-1} \succeq (\bar{m}\lambda + \sigma^2)^{-1}\mathbb{I}.$$

It comes that, with probability at least $1-\delta$, for every input location $x \in \mathcal{D}_j$ one has

$$\text{Var}\left[\mu(x)|I_k\right] \leq \text{Var}\left[\mu(x)|z_1^j, \dots, z_{\bar{m}}^j\right] = K(x,x) -$$

$$\begin{bmatrix} K(\bar{x}_1^j, x) & \dots & K(\bar{x}_{\bar{m}}^j, x) \end{bmatrix} (\bar{K}_j + \sigma^2\mathbb{I})^{-1} \begin{bmatrix} K(\bar{x}_1^j, x) \\ \vdots \\ K(\bar{x}_{\bar{m}}^j, x) \end{bmatrix} \leq$$

$$\leq \lambda - \frac{\sum_{h=1}^{\bar{m}} K(\bar{x}_h^j, x)^2}{\bar{m}\lambda + \sigma^2} \leq \lambda - \frac{\bar{m}(\lambda-\bar{\alpha})^2}{\bar{m}\lambda + \sigma^2} = \lambda - \frac{(\lambda-\bar{\alpha})^2}{\lambda + \frac{\sigma^2}{\bar{m}}} \leq \epsilon.$$

The above equations show that, with probability not smaller than $1-\delta$, $\max_{x\in\mathcal{D}_j}\bar{V}_{k_a}(x)\leq\epsilon$ and $\bar{M}_{k_b}\leq\epsilon$. In fact, since at instant k_b the subset $\mathcal{D}_j$ is visited again, the input location where $\bar{V}_{k_b}$ is maximized falls again in $\mathcal{D}_j$ and can not exceed ϵ. In view of (4.8), this implies that $\mathrm{Var}\left[\mu(x)|I_k\right]$ is converging uniformly to zero in probability. Now, define $\widetilde{\mu}_k=\mu-\widehat{\mu}_k$. From the Chebyshev inequality, we have that for every x

$$\mathbb{P}\left[|\widetilde{\mu}_k(x)|\leq sd\mid\mathrm{Var}\left[\mu(x)|I_k\right]=d^2\right]\geq 1-\frac{1}{s^2}\quad\forall x,s. \tag{4.9}$$

Note that the conditioning event is the posterior variance of the reconstruction error which has been just proved to go uniformly to zero in probability. Now, fix another arbitrary $\epsilon>0$, $0<\delta<1$ and define $\bar{\delta}=1-\sqrt{1-\delta}$. Then, we can find k_0 such that $\forall k\geq k_0$, with probability at least $1-\bar{\delta}$, one has

$$\max_x\mathrm{Var}\left[\mu(x)|I_k\right]\leq\bar{\delta}\epsilon^2.$$

Setting $s^2=1/\bar{\delta}$ in the Chebyshev inequality above, since the conditioning event holds with probability at least $1-\bar{\delta}$, we obtain that $\forall k\geq k_0$ the event

$$|\widetilde{\mu}_k(x)|\leq sd=\frac{\epsilon\sqrt{\bar{\delta}}}{\sqrt{\bar{\delta}}}=\epsilon$$

has probability not smaller than $(1-s^{-2})(1-\bar{\delta})=(1-\bar{\delta})^2=1-\delta$. This shows that, in probability, $\widetilde{\mu}_k$ is going to zero in the sup-norm topology and concludes the proof. ♢

We remark that the choice for the function $F(\cdot)$ leaves a certain degree of freedom to the designer since it allows to regulate exploration versus exploitation, as it will be shown in the simulations. For example, the previous proposition include as a special case, the strategy that performs exploration only, i.e. $F(M)=1,\forall x$. As so, the proposed algorithm can also be interpreted as a cooperative strategy for optimal sampling. It is also possible to show that the trajectory generated by the proposed algorithm eventually behaves as the traditional Lloyd algorithm. This is formally stated in the following proposition:

Proposition 4.2 Let assumptions of Proposition 4.1 hold and $F(0)=0$. For any $0<\delta<1$, $\epsilon>0$ and integer N, there exists a sufficiently large $\bar{k}$ such that, if $\mathbf{c}^L_{\bar{k}}=\widehat{\mathbf{c}}_{\bar{k}}$, then

$$\mathbb{P}\left[\left\|\widehat{\mathbf{c}}_{\bar{k}+k}-\mathbf{c}^L_{\bar{k}+k}\right\|\leq\epsilon\right]\geq 1-\delta,\quad k=0\ldots,N. \tag{4.10}$$

Proof Let $\mathcal{U}$ be the set of all the continuously differentiable sensory functions defined over $\mathcal{X}$. Let $\mathcal{U}$ be equipped with a norm $\|\cdot\|$, for instance, $\|\mu\|=\max_{x\in\mathcal{X}}\mu(x)$.

Let us define $G\left(c;\mu\right)\;:\mathcal{X}^N\times\mathcal{U}\mapsto\mathcal{X}^N$ as

$$G\left(c;\mu\right)=\mathbf{c}\left(\mathcal{W}\left(c\right)\right),$$

where the operator $\mathbf{c}$ computes the centroids according to the sensory function μ. It is known that the map G, above defined, is continuous on both arguments (see [66]). Observe that $c^L_{k+1}=G\left(c^L_k;\mu\right)$ and $\hat{c}_{k+1}=G\left(\hat{c}_k;\hat{\mu}_k\right)$.

We can write

$$\begin{aligned}\|\hat{c}_{k+1}-c^L_{k+1}\|&=\|G\left(\hat{c}_k;\hat{\mu}_k\right)-G\left(c^L_k;\mu\right)\|\\&=\|G\left(\hat{c}_k;\hat{\mu}_k\right)-G\left(\hat{c}_k;\mu\right)+G\left(\hat{c}_k;\mu\right)-G\left(c^L_k;\mu\right)\|\\&\leq\|G\left(\hat{c}_k;\hat{\mu}_k\right)-G\left(\hat{c}_k;\mu\right)\|+\|G\left(\hat{c}_k;\mu\right)-G\left(c^L_k;\mu\right)\|.\end{aligned}$$

Since the operator G is continuous on $\mathcal{X}^N$ and since $\mathcal{X}^N$ is a compact set, it follows that there exists $L_\mu>0$ such that

$$\|G\left(\hat{c}_k;\mu\right)-G\left(c^L_k;\mu\right)\|\leq L_\mu\|\hat{c}_k-c^L_k\|.$$

On the other hand, since G is continuous on $\mathcal{U}$ it follows that there exists $L_{\mu;\hat{c}_k}$ such that

$$\|G\left(\hat{c}_k;\hat{\mu}_k\right)-G\left(\hat{c}_k;\mu\right)\|\leq L_{\mu;\hat{c}_k}\|\hat{\mu}_k-\mu\|.$$

Let $\bar{L}_\mu=\max_{c\in\mathcal{X}^N}L_{\mu;c}$, then,

$$\|G\left(\hat{c}_k;\hat{\mu}_k\right)-G\left(\hat{c}_k;\mu\right)\|\leq \bar{L}_\mu\|\hat{\mu}_k-\mu\|.$$

Summarizing we have obtained that

$$\|\hat{c}_{k+1}-c^L_{k+1}\|\leq L_\mu\|\hat{c}_k-c^L_k\|+\bar{L}_\mu\|\hat{\mu}_k-\mu\|.$$

Assuming that there exists a positive integer $\bar{k}$ such that $\|\hat{\mu}_k-\mu\|\leq\xi$ for all $k\geq\bar{k}$, and such that $\hat{c}_{\bar{k}}=c^L_{\bar{k}}$, then it follows, for $k\geq\bar{k}$,

$$\|\hat{c}_k-c^L_k\|\leq\sum_{j=\bar{k}}^{k}L_\mu^{(j-\bar{k})}\bar{L}_\mu\xi\,.$$

From Proposition 4.1, it follows that for any positive integer N, positive real number ϵ and real number δ such that $0<\delta<1$, there exist a positive real number $\xi>0$ and a positive integer number $\bar{k}$ such that the following two facts are verified

- $\mathbb{P}\left[\|\hat{\mu}_k-\mu\|\leq\xi\quad\text{for all } k\in\left\{\bar{k},\ldots,\bar{k}+N\right\}\right]\geq 1-\delta,$
- $\sum_{j=\bar{k}}^{N}L_\mu^{(j-\bar{k})}\bar{L}_\mu\xi\leq\epsilon.$

This concludes the proof. ♢

Proposition 4.2 states that for sufficiently large k, since the sensory function can be estimated with arbitrary accuracy, then the r-EC algorithm mimics in probability, with arbitrary accuracy, the Lloyd algorithm. Equivalently, the centroid sequence generated by Lloyd and the one generated by the r-EC algorithm evolve arbitrarily close to each other for an arbitrary long but finite time. See Figure 4.2 for an intuitive graphical representation.

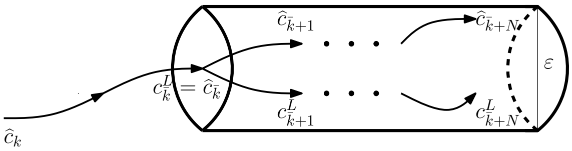


Figure 4.2: Graphical illustration of what stated in Proposition 4.2.

4.5 Numerical Considerations

In this section we outline some aspects regarding the numerical implementation of the estimation process of Section 4.3. First, we present a convenient procedure to speed up the Gaussian estimation process. Second, to further speed up the algorithm execution time, we introduce an approximated version of the r-EC algorithm which relies on a discretization of the working area $\mathcal{X}$. Accordingly, the result stated in Proposition 4.1 will be relaxed.

4.5.1 Online Gaussian Estimation

In this section we present a method to reduce the execution time of equations (4.4) and (4.6) used to estimate the sensory function μ and to compute the posterior variance, respectively. From a time computational complexity point of view, the most expensive operation is represented by the matrix inversion

$$(\bar{K}_k+\sigma^2\mathbb{I})^{-1}\,.$$

Observe that, at iteration k, the dimension of the sampled kernel, $\bar{K}$, is equal to $m_k \times m_k$ where m_k is the number of measurements taken by the robots up to iteration k. By recalling the fact that the computational complexity for the inversion of a $n \times n$ matrix is approximately $\mathcal{O}(n^3)$, it follows that the computational complexity of the r-EC algorithm grows as the cube of the number of measurements collected. In order to speed up the computational time, it is worth noting that the $\bar{K}$ matrix does not entirely change at each iteration. Indeed, recalling equation (4.5) and denoting with $\bar{K}_k$ the sampled kernel matrix at iteration k, we have that

$$\bar{K}_{k+1} = \begin{bmatrix} \bar{K}_k & \bar{K}_{k+1,12} \\ \bar{K}^T_{k+1,12} & \bar{K}_{k+1,22} \end{bmatrix}, \tag{4.11}$$

where

$$\bar{K}_{k+1,12} = \begin{bmatrix} K(x^{(1)}, x^{(m_k+1)}) & \cdots & K(x^{(1)}, x^{(m_{k+1})}) \\ & \vdots & \\ K(x^{(m_k)}, x^{(m_k+1)}) & \cdots & K(x^{(m_k)}, x^{(m_{k+1})}) \end{bmatrix},$$

$$\bar{K}_{k+1,22} = \begin{bmatrix} K(x^{(m_k+1)}, x^{(m_k+1)}) & \cdots & K(x^{(m_k+1)}, x^{(m_{k+1})}) \\ & \vdots & \\ K(x^{(m_{k+1})}, x^{(m_k+1)}) & \cdots & K(x^{(m_{k+1})}, x^{(m_{k+1})}) \end{bmatrix}.$$

Equation (4.11) highlights a recursive way to build the sampled kernel matrix $\bar{K}$ which, at each iteration, does not have to be entire recomputed. In order to take advantage of the same recursive structure for the inverse operation, it is convenient to exploit the Schur complement to compute the block matrix inversion which, given a block symmetric matrix

$$M = \begin{bmatrix} A & B \\ B^T & D \end{bmatrix},$$

lets us compute the inverse as

$$M^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B(D - B^TA^{-1}B)^{-1}B^TA^{-1} & -A^{-1}B(D - B^TA^{-1}B)^{-1} \\ -(D - B^TA^{-1}B)^{-1}B^TA^{-1} & (D - B^TA^{-1}B)^{-1} \end{bmatrix}. \tag{4.12}$$

In our case, the block A^{-1} corresponds to $(\bar{K}_k^{-1} + \sigma^2\mathbb{I})^{-1}$, while D is of dimension $(m_{k+1} - m_k) \times (m_{k+1} - m_k)$ and depends only on the new measurements taken during iteration $k+1$. So, to compute the inverse of the matrix $(\bar{K}_{k+1} + \sigma^2\mathbb{I})$ only the inversion of D plus some matrix multiplications are required. Observe that, in average, the number of new measurements taken during an iteration of the algorithm, is $\bar{\beta}N$ and, hence, $m_k \simeq k\bar{\beta}N$. Thanks to this implementation the computational complexity of the r-EC algorithm can be reduced from $\mathcal{O}(k^3\bar{\beta}N^3)$ to $\mathcal{O}(k^2\bar{\beta}^3N^3)$ where the latter, represents the computational complexity of the most expensive matrix multiplication in the inverse computation of equation (4.12). Finally, observe the computational complexity per iteration is not constant but grows quadratically in the number of iterations, i.e., the computational complexity is unbounded. To overcome this problem, in the next section we provide a modified version of the proposed algorithm whose computational complexity per iteration is constant.

4.5.2 Grid Based Approximation

In this section we present a modified version of the r-EC algorithm which is light and fast and then suitable for numerical implementation. We refer to it as the r-EC-grid algorithm. The idea relies on a spatial discretization of the continuous convex domain $\mathcal{X}$. In particular, we constrain the robots to collect measurements from a set of predetermined finite number of input locations which are obtained thanks to a discretization of $\mathcal{X}$. To do so, we now introduce a sampled version of the original space $\mathcal{X}$.

Definition 4.3 (Sampled Space) Consider the finite set of m input locations $\mathcal{X}_{\text{grid}} := \{x_{\text{grid},1}, \dots, x_{\text{grid,m}}\} \subset \mathcal{X}$ where $\mathcal{X} \subset \mathbb{R}^2$ is a convex and closed polygon. Given the scalar $\Delta > 0$, we say that set $\mathcal{X}_{\text{grid}}$ forms a *sampled space* of resolution Δ if

$$\min_{i=1,\dots,m} \|x_{\text{grid,i}} - x\| \leq \Delta, \quad \forall x \in \mathcal{X} \tag{4.13}$$

Moreover, we introduce the following projection operator that projects $x \in \mathcal{X}$ onto its closest point in $\mathcal{X}_{\text{grid}}$.

Definition 4.4 (Projection Operator) Given a convex and closed polygon $\mathcal{X}$ and its sampled version $\mathcal{X}_{\text{grid}}$, we define the *projection operator* of $x \in \mathcal{X}$ onto $\mathcal{X}_{\text{grid}}$ as

$$\mathcal{X} \longmapsto \mathcal{X}_{\text{grid}} \ : \ x \longmapsto \text{proj}(x) = \arg \min_{a \in \mathcal{X}_{\text{grid}}} \|x - a\| \,. \tag{4.14}$$

To force the evolution of the robots on $\mathcal{X}_{\text{grid}}$, Algorithm 4 must be changed. In particular, the BS has to compute a control input in order to drive the robots only on input locations which own to $\mathcal{X}_{\text{grid}}$. To accomplish this task, lines 19 and 21 of Algorithm 4 can be changed as follows:

$$\begin{aligned} \texttt{line 19} \quad &\longmapsto \quad b_{i,k} = \text{argmax}_{x \in \mathcal{P}_{i,k} \cap \mathcal{X}_{\text{grid}}} V_k(x)\,, \\ \texttt{line 21} \quad &\longmapsto \quad b_{i,k} = \text{proj}(\widehat{c}_{i,k})\,. \end{aligned}$$

Note that lines 19 and 21 are substantially different from each other. Indeed, line 21 simply says to project the centroids, $\widehat{c} \in \mathcal{X}$, onto the closer points owing to $\mathcal{X}_{\text{grid}}$. Conversely, exploiting line 19, the BS directly computes the input location owing to $\mathcal{X}_{\text{grid}}$ which maximizes the posterior variance restricted on the grid, for each Voronoi region. This is in general different from the projection onto the grid of the input location where the posterior variance maximum is located. That is,

$$\text{argmax}_{x \in \mathcal{P}_{i,k} \cap \mathcal{X}_{\text{grid}}} V_k(x) \neq \text{proj}\left(\text{argmax}_{x \in \mathcal{P}_{i,k}} V_k(x)\right) \,.$$

This precaution is fundamental to ensure convergence of the estimator of μ on $\mathcal{X}_{\text{grid}}$ which is stated in the following proposition.

Proposition 4.5 Let us consider the r-EC-grid algorithm. If $F(M) : [0,1] \to [0,1]$ is a continuous and monotonically increasing function such that $F(M) > 0$ for $M > 0$, then

$$\widehat{\mu}_k(x) \xrightarrow{P} \mu(x), \ \forall x \in \mathcal{X}_{\text{grid}}. \tag{4.15}$$

The proof of Proposition 4.5 can be found together with the proof of Proposition 4.6.
The following result instead characterizes the asymptotic performance of the estimator on a generic input location, possibly falling outside $\mathcal{X}_{\text{grid}}$. Before stating it, some useful notation is introduced. The $m \times m$ covariance matrix of the function μ sampled on $\mathcal{X}_{\text{grid}}$ is denoted by K_{grid}, i.e.

$$[K_{\text{grid}}]_{ki} = K(x_{\text{grid,k}}, x_{\text{grid,i}}).$$

Given a generic x, we also use $k_{\text{grid}}(x)$ to denote the row vector

$$k_{\text{grid}}(x) = [K(x, x_{\text{grid},1}) \ \dots \ K(x, x_{\text{grid,m}})] \,.$$

Below, recall also that $K(a,b) = h(\| \ a - b \ \|)$ and $\lambda = K(x,x)$.

Proposition 4.6 Let the assumptions of Proposition 4.5 hold, with the probabilities of successfully transmitting and receiving a packet given by $\overline{\beta}, \overline{\gamma} > 0$, respectively, and $F(0) = 0$. If $\mathcal{X}_{\text{grid}}$ is a sampled subset of the space $\mathcal{X}$ of resolution Δ, as in Definition 4.3, one has

$$\lim_{k\to\infty} V_k(x) = \lambda - k_{\text{grid}}(x) K_{\text{grid}}^{-1} k_{\text{grid}}(x)^{\top}, \tag{4.16}$$

where convergence is in probability and holds also uniformly w.r.t. x. In addition, the following simple Δ dependent bound holds

$$\lim_{k\to\infty} V_k(x) \leq \lambda - \frac{h^2(\Delta)}{\lambda}. \tag{4.17}$$

Finally, if K is the Gaussian kernel with $K(a,b) = \lambda e^{-\frac{\|a-b\|^2}{\zeta^2}}$, for small Δ we have

$$\lim_{k\to\infty} V_k(x) \leq \lambda - \frac{h^2(\Delta)}{\lambda} \approx \frac{\lambda\Delta^2}{\zeta^2}. \tag{4.18}$$

Proof As already noticed in Section 4.5.2, when measurements can be collected only over the input locations contained in the finite set $\mathcal{X}_{\text{grid}}$, the estimation process at a generic instant k is equivalent to reconstructing the function μ from measurements

$$w_{i,\ell_i} = \mu(x_{\text{grid,i}}) + \nu_i.$$

Above, conditional on the process history up to instant k, the noise variances ν_i are zero-mean Gaussian with variance $\sigma_i^2 = \frac{\sigma^2}{\ell_i}$ where ℓ_i is the number of visits at $x_{\text{grid,i}}$. The proof of Proposition 4.1 can be now followed just replacing the function domain $\mathcal{X}$ with $\mathcal{X}_{\text{grid}}$, with the Voronoi regions covering the entire $\mathcal{X}$ defined at every instant k by a map having as arguments only the estimates of the m random variables $\mu(x_{\text{grid,i}})$. One then obtains that $\ell_i \to \infty$ for $i = 1, \ldots, m$, i.e. the posterior variances of all the $\mu(x_{\text{grid,i}})$ go to zero. Hence, equations (4.15,4.16) immediately follow. To obtain equation (4.17), note the following two facts. First, given any $x \in \mathcal{X}$ there exists $x_{\text{grid,i}} \in \mathcal{X}_{\text{grid}}$ such that $\|x - x_{\text{grid,i}}\| \leq \Delta$. Second, the r.h.s. in (4.17) is exactly the posterior variance of $\mu(x)$ conditional on the perfect knowledge of $\mu(x_{\text{grid,i}})$ with $\|x - x_{\text{grid,i}}\| = \Delta$. Eq. (4.17) is then obtained recalling that, by assumption, if $\| a-b \| \leq \| c-d \|$ then $h(\| a-b \|) \leq h(\| c-d \|)$ and $K(x,x) = \lambda$, $\forall x \in \mathcal{X}$. Finally, eq. 4.18 is just the expansion of the r.h.s. of (4.17) around $\Delta = 0$ in the Gaussian kernel case. ♢

When adopting the grid-based strategy, measurements can be collected only over the input locations contained in the finite set $\mathcal{X}_{\text{grid}}$. Using basic results on estimation of Gaussian processes, one can see that equation (4.16) is the posterior variance of μ conditional on the perfect knowledge of the function on the grid. Hence, the above result shows that our update mechanism ensures convergence to the minimum possible error compatible with $\mathcal{X}_{\text{grid}}$. Equation (4.17) then shows how the posterior variance can be made uniformly and arbitrarily small by choosing a Δ sufficiently small. In particular, from eq. (4.18) one sees that the error converges to zero at least quadratically in Δ when a Gaussian kernel is adopted.

Since the grid is composed by a finite number of locations, new measurements can fall exactly over the same input location. Let $n \leq m$ be the number of distinct input locations on the grid visited at least once up to instant k, denoted simply by $x_1, ..., x_n$ (they coincide with the set $\{x_{\text{grid,i}}\}_{i=1}^{m}$ defined above when $n = m$). Let also y_i be the ℓ_i-th measurement taken in the input location $x_i \in \mathcal{X}_{\text{grid}}$ and define the *virtual measurement* w_{i,ℓ_i} as the average of the first ℓ_i measurements associated to the input location x_i. To avoid storing all the measurements up to y_i, it is possible to compute w_{i,ℓ_i} in a recursive way exploiting only the current measurement y_i and the previous virtual measurements w_{i,ℓ_i-1}. In particular, one has

$$w_{i,\ell_i} = \frac{\ell_i - 1}{\ell_i} w_{i,\ell_i-1} + \frac{1}{\ell_i} y_i.$$

Accordingly, we define the variance associated to the virtual measurement w_{i,ℓ_i} as $\sigma_i^2 = \frac{\sigma^2}{\ell_i}$ and with Σ a diagonal matrix collecting these noise variances, i.e. $\Sigma = \operatorname{diag}(\{\sigma_i^2\}_{i=1}^n)$. Once we collect two or more measurements in the same input location, the size of the sampled Kernel (4.5) does not vary. It instead increases when an input location on the grid is visited for the first time. When all of these locations are visited at least once, the sampled kernel does not change any more: it reaches its maximum possible size, becoming the matrix K_{grid} previously defined. Instead, every time a new measurement is achieved, one has to update the variance matrix Σ and the vector with the virtual measurements $w = [w_{1,\ell_1}, \ldots, w_{n,\ell_n}]^T$ with $n \leq m$. The function estimate can then be computed as follows:

$$\widehat{\mu}_k(x) = \mathbb{E}\left[\mu(x)|I_k\right] = \sum_{j=1}^{n} c_j K(x_j, x), \quad x \in \mathcal{X}, \quad n \leq m \tag{4.19}$$

where

$$\begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} = (\bar{K}_k + \Sigma)^{-1} \begin{bmatrix} w_{1,\ell_1} \\ \vdots \\ w_{n,\ell_n} \end{bmatrix}$$

and

$$\bar{K}_k = \begin{bmatrix} K(x_1,x_1) & \ldots & K(x_1,x_n) \\ \vdots & & \vdots \\ K(x_n,x_1) & \ldots & K(x_n,x_n) \end{bmatrix}. \tag{4.20}$$

The posterior variance becomes

$$V_k(x) = \operatorname{Var}\left[\mu(x)|I_k\right] = K(x,x) - \begin{bmatrix} K(x_1,x) & \ldots & K(x_n,x) \end{bmatrix} (\bar{K}_k + \Sigma)^{-1} \begin{bmatrix} K(x_1,x) \\ \vdots \\ K(x_n,x) \end{bmatrix}. \tag{4.21}$$

In Algorithm 4 lines 11 and 12 must be computed using equations (4.19) and (4.21) instead of (4.4) and (4.6), respectively. Finally, Figure 4.3 shows the behaviour of the bound of the posterior variance of equation (4.16).

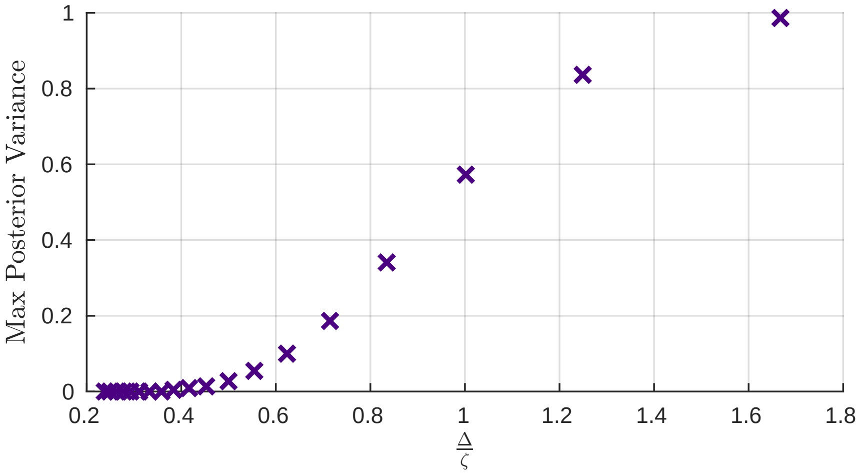


Figure 4.3: Evolution of the max of the posterior variance as function of $\frac{\Delta}{\zeta}$ for a Gaussian kernel with standard deviation ζ (Equation (4.16)).

4.6 Simulations

In this section we provide some simulations showing the performances of the r-EC algorithm. We run all the simulations in MATLAB on a desktop computer with a processor Intel Core i7-4790 and 8Gb of RAM. We consider a team of $N = 8$ robots and a squared domain $\mathcal{X} = [0, 1] \times [0, 1]$. Moreover, we use the Gaussian kernel

$$K(x, x') = e^{-\frac{\|x-x'\|^2}{0.002}} .$$

Unless differently specified, the unknown sensory function μ is a combination of two bi-dimensional Gaussian:

$$\mu(x) = 5 \left(e^{\frac{-\|x-\mu_1\|^2}{0.04}} + e^{\frac{\|x-\mu_2\|^2}{0.04}} \right) ,$$

where

$$\mu_1 = \begin{bmatrix} 0.8 \\ 0.2 \end{bmatrix} \quad \mu_2 = \begin{bmatrix} 0.5 \\ 0.7 \end{bmatrix} .$$

For computational reasons, the function μ and the posterior variance are evaluated over a grid with 65 points per side. However, the centroids and the input locations are not approximated. The Voronoi regions are always computed with respect to the last value of the centroids. Finally, we analyze the behaviour of the r-EC algorithm using

$$F_\alpha(M) = M^\alpha ,$$

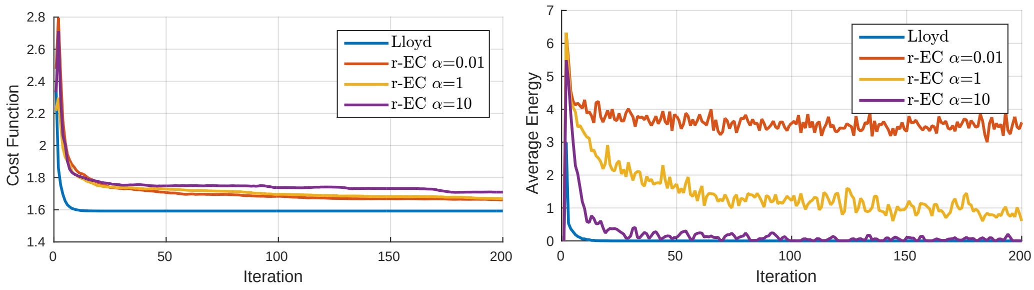
being the max of posterior variance M normalized between $[0, 1]$. The tuning parameter α lets us control the trade-off between exploration and exploitation. In particular, for $0 \leq \alpha < 1$ the robots are more prone to exploration; for $\alpha = 1$ the probability that the robots perform exploration or exploitation linearly varies with M; while for $\alpha > 1$ the robots are more prone to exploit the estimated sensory function to perform coverage control. In the following we choose a value of $\alpha = 2$, unless differently specified.

4.6.1 r-EC Algorithm Analysis

In this subsection we analyze the behaviour of the r-EC algorithm. Figure 4.4(a) shows the coverage cost function, $H(\mathcal{P}, \mu)$, for the standard Lloyd' algorithm and for the r-EC algorithm. It can be seen how the r-EC algorithm converges to a different value than the standard Lloyd algorithm. This is because the initial conditions in any single run are randomized and the robots can reach a configuration corresponding to a different local minimum. The behavior for three different values of α are shown. Observe that for small value of α the cost function converges more rapidly. This is due to the fact that the robots are more prone to explore the environment and, consequently, the estimated sensory function is a better approximation of the true function. So, since the cost function is computed with respect to the centroids of the Voronoi regions, a better estimate implies a better approximation of the centroids and consequently a lower value of the cost function. This trend is inverted increasing the value of α. On the other hand, the average energy spent by the robots at iteration k, i.e., $\frac{1}{N} \sum_{i=1}^{N} \|u_{i,k}\|^2$, decreases for increasing α, as shown in Figure 4.4(b). This is due to the following two facts:

(i) when a sufficiently accurate estimate of the true sensory function is reached, the function $F_\alpha(M)$ which describes the probability of the robots to perform exploration or coverage, forces the robots to be more prone to perform coverage control;

(ii) since the estimate is sufficiently accurate, the Voronoi's partition of the working area does not substantially change, so their centroids do not change as well. Consequently, the robots, which are forced to move towards the centroids, do not move.



(a) Cost function evolution.

(b) Average energy spent by the robots.

Figure 4.4: Comparison between the Llyod and r-EC algorithms. The plot shows a single run for different value of the tuning parameter α.

The posterior variance evolution shows how the estimate improves at each iteration. Figure 4.5 reports the max, the min and the average of the posterior variance, i.e.,

$$V_{\text{max}} = \max_{x \in \mathcal{X}} \; V_k(x) \, ; \; V_{\text{ave}} = \frac{\int_{\mathcal{X}} V_k(x) dx}{\int_{\mathcal{X}} dx} \; ; \; V_{\text{min}} = \min_{x \in \mathcal{X}} \; V_k(x) \, .$$

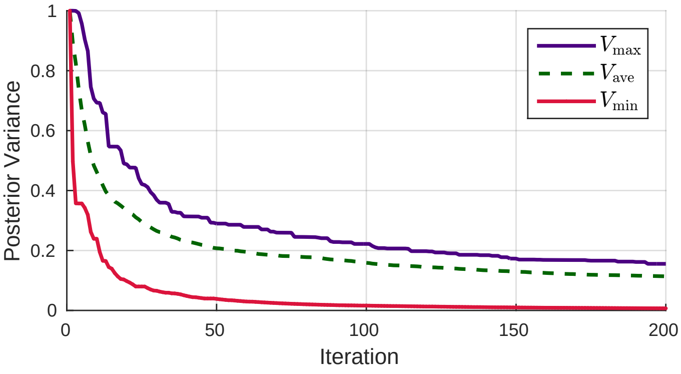


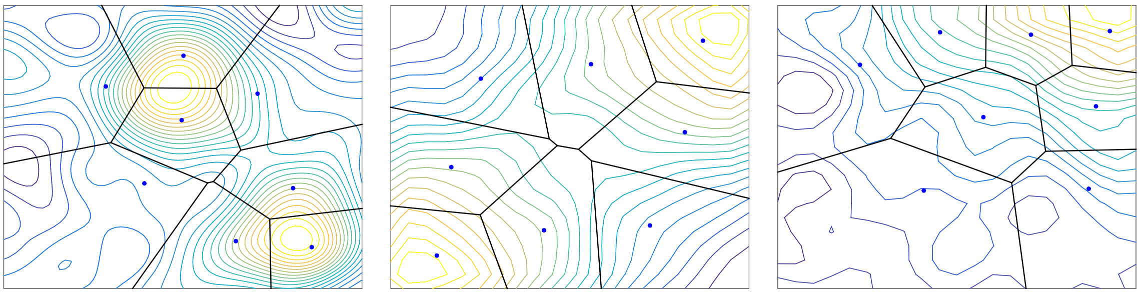
Figure 4.5: Evolution of the min, the average and the max of the posterior variance of the estimate.

Remark 4.7 Note that the curves of Figure 4.5 scale with the number of robots. That is increasing the number of robots, the max, the average and the min of the posterior variance tend to zero more rapidly. In particular the curves scale as $\frac{1}{\sqrt{N}}$.

Figure 4.6 reports a single realization of the r-EC algorithm for three different sensory functions μ. In particular, the sensory function to which correspond the plot in the left panel is spanned by the kernel since it is a combination of gaussians. Conversely, the sensory function used in the middle and right panels are obtained from sinusoidal curves thus they are not spanned by the kernel. However, the algorithm performs really well. This must not seem surprising since it is known that using a gaussian kernel it is possible to retrieve a very good estimate for almost any the sensory function. In the figure, the blue dots show the locations of the centroids obtained using the r-EC algorithm, while the black lines identify the Voronoi' partitions borders.

4.6.2 Packet Losses Analysis

In this section we show the effectiveness and the robustness of the r-EC algorithm against lossy communications. In particular, we test the algorithm for different percentage of packet losses



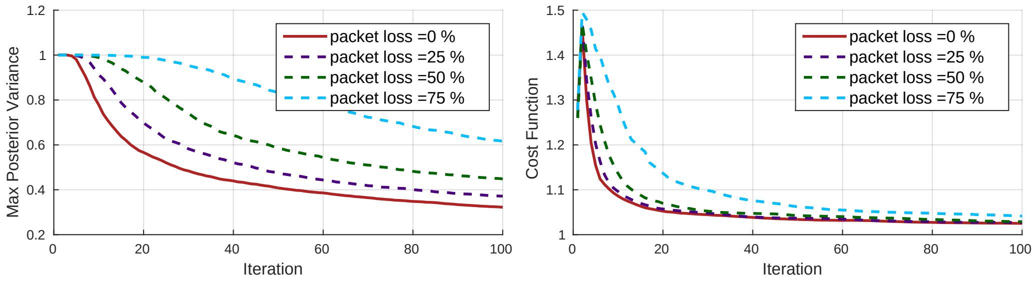
(a) $\mu(x) = 5\left(e^{\frac{-\|x-\mu_1\|^2}{0.04}} + e^{\frac{\|x-\mu_2\|^2}{0.04}}\right)$

(b) $\mu(x) = 10\cos(\pi x_1)\cos(\pi x_2) + 10$

(c) $\mu(x) = 10\sin\left(x_1 x_2^2\right)$

Figure 4.6: Voronoi partitions computed using the r-EC Algorithm (black lines) for different sensory function $\mu(x)$. Blue dots indicate the locations of the centroids obtained with the r-EC algorithm.

assuming, for the sake of simplicity, that the losses probabilities, $\overline{\beta}, \overline{\gamma}$, are equal among the nodes and within the same node. That is the probability of either successfully transmit to or receive data from the BS is the same. It is worth noting that this assumption does not compromise the well behavior of the algorithm which can be easily implemented assigning different losses probability. Figure 4.7(a) and 4.7(b) plot the evolution of the posterior variance and of the cost function, respectively, corresponding to four values of packet loss probability, namely 0%, 25%, 50% and 75%. It can be seen how the convergence rate is reduced due to the packet loss. However, the convergence of the r-EC algorithm holds.



(a) Max of the posterior variance.

(b) Cost function evolution.

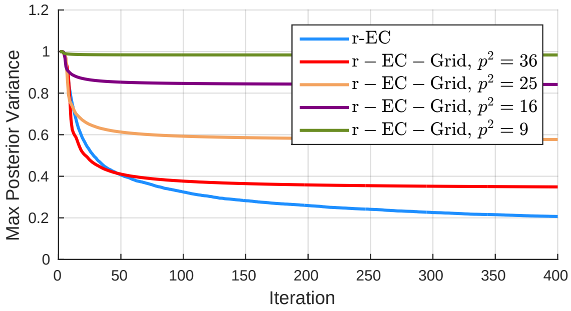
Figure 4.7: Behaviour of the r-EC algorithm for different values of the packet loss probability averaged over 100 Monte Carlo runs.

4.6.3 Grid Based Approximation Analysis

In this section we show a numerical comparison between the original r-EC algorithm of Section 4.4 and its grid based approximation presented in Section 4.5.2. Since $\mathcal{X}$ has been chosen to be equal to a a square set, $[0, \ell]^2$, with edge of length $\ell = 1$, it is convenient to let the sampled space $\mathcal{X}_{\text{grid}}$ be a uniform sampled grid defined as

$$\mathcal{X}_{\text{grid}} = \Big\{ x_{11}, \ldots, x_{pp} \mid x_{ij} = (x_{ij,1}, x_{ij,2}) \text{ with } x_{ij,1} = \frac{\ell}{p-1}(i-1), \\ x_{ij,2} = \frac{\ell}{p-1}(j-1)\ ,\ i, j = 1, \ldots, p\ \Big\}$$

where the integer $p > 0$ represents the desired number of points per edge. In particular, Figure 4.8 (left panel) shows the performance, in terms of max of the posterior variance, of the algorithms for different level of space discretization, i.e., different total number of points p^2. It can be seen that the grid based approximation is slightly faster during the first iterations but it reaches convergence in more or less 150 iterations. Conversely, in the original implementation the max of the posterior variance keeps decreasing since it converges to zero asymptotically. However, in terms of execution time the r-EC-grid algorithm is much lighter than the r-EC algorithm, see table in Figure 4.8 (right panel). This represents the major advantage of using this implementation since given a desired final accuracy on the max of the posterior variance, according to Proposition 4.6, the execution time can be reduced of different order of magnitudes. Finally, as stated by Proposition 4.6, in the grid based approximation, the finer is the grid the lower is the final value of the max of the posterior variance, see Figure 4.8 (left panel).



p^2	$\frac{\Delta}{\zeta}$	Steady State (after 400 iterations)	Exe. Time [sec.]
9	1.65	0.9836	2.3
16	1.25	0.8418	2.4
25	1	0.5769	3.9
36	0.83	0.3489	4.7
r-EC	–	0.2	865.4

Figure 4.8: Comparison between the original r-EC algorithm and the grid based approximation for different total number of points p^2. The left panel shows the evolution of the max of the posterior variance averaged over 100 Monte Carlo runs. The table in the right panel reports the steady state value after 400 iterations and the execution times obtained using the grid based approximation and classic algorithms.

4.6.4 Parameters tuning and scaling

The previous algorithms require the choice of some design parameters. Here we provide some sideline based on the extensive simulative experiments that we performed. The first design choice is the Kernel to be used. A common choice is to use Gaussian Kernels, i.e.

$$K(a,b) = \lambda e^{\frac{\|a-b\|^2}{\varsigma^2}}$$

where λ represents the a-priori uncertainty variance of the map μ, and ζ is the standard deviation of the Kernel. The choice of Gaussian Kernels is motivated by their good representative features, although other choices are possible [67]. The parameters λ and ζ based on the a-priori knowledge of the map to be learned. For example, if the map to be identified is a temperature map, one might know in advance that it must be within a certain range, therefore λ could be set to the square half of the maximum range, i.e.

$$\lambda \approx \frac{\max \mu^2(x)}{4}$$

The parameter ζ is related to how rapidly the map $\mu(x)$ changes over the space, therefore if one knows the typical distance beyond which the temperature is uncorrelated with the temperature taken at a specific point, let us call it δ_μ, then a reasonable choice is given by:

$$\zeta \approx \frac{\delta_\mu}{10}$$

For what concern the size of the sampling grid Δ in the grid-based approximated algorithm, this is to be chosen based on the desired a-posteriori steady state error variance that one wants to have. Based on Fig. 4.3, then Δ should be chosen around

$$\Delta \approx \frac{\zeta}{3}$$

although a good partitioning is obtained already for $\Delta \approx \zeta$.

Another important design choice is given by the desired tradeoff between exploration and exploration. We found that the suggested function

$$F(M) = M^\alpha$$

is sufficiently flexible and easy to use, where $\alpha = 1$ would provide a good general compromise in terms of this trade-off.

Finally, we would like to summarize the scaling properties of the proposed algorithm in terms of its ability to estimate the unknown map $\mu(x)$. In fact the average estimation error scales approximately as follows in terms of the different parameters of the system:

$$|\mu(x) - \hat{\mu}_k(x)| \approx \sqrt{\lambda} \left(\frac{1}{\sqrt{Nk}} + \frac{\Delta}{\zeta} \right) \tag{4.22}$$

where the first term is obtained by assuming to divide the exploration region in N subregions of equal size and then perform uniform sampling, while the second term is due to the error given by the approximating grid given in equation (4.18). This expression clearly shows the benefit of a larger number of agents N, how the error scales with time k, and the impact of the size of the approximating grid.

5

Kalman Filter meets Gaussian Regression

In this chapter we address the problem of designing algorithms that are capable to obtain the optimal estimate of a Gaussian process in an efficient manner. In particular we define a class of Kernels for which the exact estimate can be obtained through a Kalman filter and we also propose an approximate algorithm to estimate processes generated by Kernels that do not belong to aforementioned class. In Section 5.1 we introduce the key aspects of the non-parametric estimation while in Section 5.2 we introduce the main aspects of the Kalman filtering. In Section 5.3 we present our main results and we show its effectiveness through some simulations in Section 5.4.

5.1 Nonparametric Estimation

In this section we review some key aspects regarding the nonparametric estimation in the Gaussian regression framework.

Let $f : \mathcal{A} \mapsto \mathbb{R}$ be a zero-mean Gaussian field with covariance, also called kernel, $K : \mathcal{A}\times\mathcal{A} \to \mathbb{R}$, where $\mathcal{A}$ is a compact set. Assume to have a set of $N \in \mathbb{N}$ noisy measurements, i.e. $\{a_i, y_i\}_{i=1}^N$, of the form

$$y_i = f(a_i) + v_i, \tag{5.1}$$

where v_i is a zero-mean Gaussian noise with variance σ^2, i.e. $v_i \sim \mathcal{N}(0, \sigma^2)$, independent of the unknown function. Given the data-set $\{a_i, y_i\}_{i=1}^N$, exploiting known results on estimation of Gaussian random fields , one obtains that the estimate of f is a linear combination of the kernel sections $K(a_i, \cdot)$. In particular, for any a it holds that

$$\hat{f}(a) = \mathbb{E}\left[f(a)|\{a_i, y_i\}_{i=1}^N\right] = \sum_{i=1}^N c_i K(a_i, a), \tag{5.2}$$

where the expansion coefficients c_i are obtained as

$$\begin{bmatrix} c_1 \\ \vdots \\ c_N \end{bmatrix} = (\bar{K} + \sigma^2 \mathbb{I})^{-1} \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix},$$

while the matrix $\bar{K}$ is obtained evaluating the kernel at the input-locations, i.e.

$$\bar{K} = \begin{bmatrix} K(a_1, a_1) & \cdots & K(a_1, a_N) \\ \vdots & \ddots & \vdots \\ K(a_N, a_1) & \cdots & K(a_N, a_N) \end{bmatrix}.$$

Finally, the posterior variance of the estimate at a generic input location $a \in \mathcal{A}$ is

$$V(a) = \mathrm{Var}\left[f(a)|\{a_i, y_i\}_{i=1}^N\right] = K(a, a) - \begin{bmatrix} K(a_1, a) & \cdots & K(a_N, a) \end{bmatrix} (\bar{K} + \sigma^2 \mathbb{I})^{-1} \begin{bmatrix} K(a_1, a) \\ \vdots \\ K(a_N, a) \end{bmatrix}.$$

To apply this approach in a context where new data arrive as time increases, one can see that more and more input-locations and output values must be stored and processed at each iteration to compute the estimate. Indeed, the most expensive operation is the inversion of a $N \times N$ matrix which requires $O(N^3)$ operations. This eventually leads to a memory and computational consuming process growing unbounded with N.

Observe that, in spatial Gaussian processes, a denotes a spatial variable defining a spatial location. Analogously, in temporal Gaussian processes, a typically represents time. In the rest of the paper, we are interested in spatio-temporal processes, where time-varying functions are considered so that $f(a) = f(x,t)$ with a now incapsulating both a spatial and a temporal quantity. The domain set can thus be decomposed as $\mathcal{A} := \mathcal{X} \times \mathbb{R}_+$, with $\mathcal{X}$ and $\mathbb{R}_+$ indicating the spatial and temporal domain, respectively. We will also assume $\mathcal{X}$ to be a compact set.

5.2 Kalman Filter for Finite Dimensional State Linear Estimation

In this section we briefly recall some basic notions on Kalman filtering in finite-dimensional state-space linear systems. Consider the following discrete-time linear dynamical system

$$\begin{aligned} s_{k+1} &= A s_k + w_k, \\ y_k &= C_k s_k + v_k, \end{aligned} \tag{5.3}$$

where $s_k \in \mathbb{R}^n$ is the state vector, $y_k \in \mathbb{R}^m$ the output vector, $w_k \in \mathbb{R}^n$ and $v_k \in \mathbb{R}^m$ are i.i.d. zero-mean Gaussian random vectors with covariance matrices $Q \geq 0$ and $R > 0$, respectively. $A \in \mathbb{R}^{n \times n}$ is the state matrix and $C_k \in \mathbb{R}^{m \times n}$ is the time-varying output matrix. As commonly done, we assume both the process and measurement noise to be uncorrelated with respect to each other, i.e. $\mathbb{E}\left[w_k^T v_s\right] = 0 \ \forall_{k,s}$. We also assume that the initial condition s_0 is a Gaussian vector of zero-mean and covariance Σ_0. Furthermore, we define

$$\hat{s}_{k+1|k+1} = \mathbb{E}\left[s_{k+1} | y_0, \ldots, y_{k+1}\right].$$

Under the stated assumptions, the Kalman Filter is then described by the following equations

$$\begin{aligned} &\hat{s}_{k+1|k} = \mathbb{E}\left[s_{k+1} | \hat{s}_{k|k}, y_k\right] = A \hat{s}_{k|k} \\ &\Sigma_{k+1|k} = A \Sigma_{k|k} A^T + Q \\ &\hat{s}_{k+1|k+1} = \mathbb{E}\left[s_{k+1} | \hat{s}_{k+1|k}, y_k\right] = \hat{s}_{k+1|k} + K_{k+1}\left(y_{k+1} - C_k \hat{s}_{k+1|k}\right) \\ &\Sigma_{k+1|k} = \Sigma_{k+1|k} - K_{k+1} C_k \Sigma_{k+1|k} \\ &K_{k+1} = \Sigma_{k+1|k} C_k^T \left(C_k \Sigma_{k+1|k} C_k^T + R\right)^{-1} \end{aligned} \tag{5.4}$$

where we set $\hat{s}_{0|-1} = \mathbb{E}[s_0] = 0$ and $\Sigma_{0|-1} = Cov[s_0] = \Sigma_0$. It is well known, [68], that if we assume the output matrix constant, i.e. $C_k = C$, under the hypothesis of stabilizability of the pair (A, Q) and detectability of the pair (A, C) the estimation error covariance of the Kalman filter converges to a unique value from any initial condition.

5.3 Kalman Regression on a Finite Dimensional Grid

In this section we show how to combine the two approaches described in the previous sections to estimate a time-varying function on a finite dimensional grid. The next section will be then devoted to extend the results to perform estimation on any point of the spatial domain.

As already mentioned, Gaussian regression turns out appealing, especially when no precise information on f is available, e.g. just smoothness. However, the computational and storage burden is high, both scaling as $O(N^3)$. On the other hand the Kalman filter algorithm is recursive and it requires to store only the last state estimate, poster variance and measurements. Our next developments illustrate a way to bridge nonparametric estimation and Kalman filtering.

Consider a function $f : \mathcal{X} \times \mathbb{R}_+ \to \mathbb{R}$ modeled as a zero-mean Gaussian Process with covariance K. Hereby, to simplify notation, $f_t(x)$ is used in place of $f(x,t)$. Let us also define a spatial discretization of $\mathcal{X}$, as described below.

Definition 5.1 (Sampled Space) Letting $\mathcal{X} \subset \mathbb{R}^2$, we denote with $\mathcal{X}_{\text{grid}} \subset \mathcal{X}$ a finite set containing M input locations, i.e. $\mathcal{X}_{\text{grid}} := \{x_1, \dots, x_{\text{M}}\}$.

With this definition in mind, unless differently specified, in the rest of this section all the spatial input locations where data are collected are assumed constrained on $\mathcal{X}_{\text{grid}}$ whereas the time variable t can assume any value in $\mathbb{R}_+$. Note however that, even under this restriction, the Gaussian estimator in general suffers of high computational and storage burden. This is essentially due to the non Markovian nature of the process f w.r.t. the temporal variable. To overcome this problem, our main idea is to define a new class of Markovian kernels through a suitable yet rich class of state space models. This will pave the way to the use of Kalman filtering for function reconstruction. For this purpose, it is useful to define $\mathbf{f}_t$ as the vector with components given by $f_t(\text{x})$ sampled on $\mathcal{X}_{\text{grid}}$, i.e. $\mathbf{f}_t = [f_t(x_1), \dots, f_t(x_{\text{M}})]^T$.

Assumption 5.2 The covariance of the Gaussian process $f_t(x)$ is separable in time and space and stationary in time, i.e.

$$K(x_1, x_2, t_1, t_2) = \lambda K_1(x_1, x_2) K_2(\tau), \quad \tau = t_2 - t_1.$$

In addition, the power spectral density of $K_2(\tau)$, denoted by

$$S(\omega^2) = \mathcal{F}\left[K_2(\tau)\right],$$

is a rational function of order r.

Example 5.3 To help the reader's intuition we now introduce two examples of stationary kernels K_2 which model time correlation, the first one will satisfy Assumption 5.2 while the second one will not admit a rational representation.
First, consider the exponential kernel

$$K_2(\tau) = e^{-\sigma_t |\tau|}.$$

The associated spectral density is given by

$$S(\omega^2) = \mathcal{F}\left[K_2(\tau)\right] = \frac{2\lambda\sigma_t}{\omega^2 - \sigma_t^2} = \frac{\sqrt{2\lambda\sigma_t}}{(\omega - \sigma_t)} \frac{\sqrt{2\lambda\sigma_t}}{(\omega + \sigma_t)}$$

which is rational of order 1. Instead, if we consider the squared exponential kernel

$$K_2(\tau) = \lambda e^{-\sigma_t^2 \tau^2},$$

its spectral density is not rational, i.e.

$$\mathcal{F}\left[K_2(\tau)\right] = \sqrt{\pi} \frac{\lambda}{\sigma_t} e^{-(\frac{w}{2\sigma_t})^2}.$$

So this kernel does not satisfy Asssumption 5.2.

We are now ready to present our first result which links the process $f_t(x)$ to an equivalent state space representation.

Proposition 5.4 Let $\lambda K_1(x_1, x_2)K_2(\tau)$ a kernel which satesifies Assumption 5.2 and with spectral density of $K_2(\tau)$ of the form

$$S(\omega^2) = W(i\omega)W(-i\omega)$$

where

$$W(i\omega) = \frac{b_m(i\omega)^m + b_{m-1}(i\omega)^{m-1} + \cdots + b_0}{a_n(i\omega)^n + b_{a-1}(i\omega)^{n-1} + \cdots + a_0}.$$

Then the process $f_t(x)$ admits, constrained on $\mathcal{X}_{\text{grid}}$, the following strictly proper state-space representation

$$\begin{cases} \dot{s}_t^j = Fs_t^j + Gw_t^j \\ z_t^j = Hs_t^j \qquad\qquad j = 1, \dots, M, \\ \mathbf{f}_t = K^{1/2}Z_t \end{cases} \tag{5.5}$$

where j is an index cycling through all the inputs locations of $\mathcal{X}_{\text{grid}}$, $\mathbb{E}\left[s_t^i s_t^j\right] = 0\ \forall t$ with $i \neq j$, $Z_t = \left[z_t^1, \dots, z_t^M\right]^T$, K is such that its (i,j)-th entry is $K_{ij} = K_1(x_i, x_j)$ with $x_i, x_j \in \mathcal{X}_{\text{grid}}$, w is zero-mean white Gaussian noise, $s(0) \sim \mathcal{N}(0, \lambda(H^T H)^\dagger H^T H (H^T H)^\dagger)$ where λ is the variance of z and the model matrices are

$$F = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & a_n \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

$$H = \begin{bmatrix} b_0 & b_1 & b_2 & \dots & b_n \end{bmatrix}.$$

Proof The process $\mathbf{f}_t$ is a Gaussian process, because w_t is Gaussian and the solution of a linear differential equation is a linear operation on the input. Now we need to prove that the covariance of $\mathbf{f}_t$ is $KK_2(\tau)$. The first two equations of model (5.5) are the state space representation of the rational power spectral density $S(\omega)$ thus $\mathbb{E}\left[z_{t+\tau}^j z_t^j\right] = K_2(\tau)$. So we can state that the covariance of the process $\mathbf{f}_t$ is $\mathbb{E}\left[\mathbf{f}_{t+\tau}\mathbf{f}_t\right] = K^{1/2}\left[\mathbb{I} \cdot K_2(\tau)\right]\left(K^{1/2}\right)^T = KK_2(\tau)$. ♢

Example 5.5 We continue the previous example and we see how to compute the matrices F, H and $K^{1/2}$ necessary to implement the state space model of the process to estimate. Consider a zero-mean Gaussian process $f_t(x)$ with covariance

$$K(x_1, x_2, \tau) = \lambda K_1(x_1, x_2)K_2(\tau) = \lambda e^{-\sigma_x(x_1 - x_2)^2} e^{-\sigma_t|\tau|} \tag{5.6}$$

which, as we showed in Example 5.3, satisfies Assumption 5.2. Then, thanks to Proposition 5.4, we know that process $f_t(x)$ admits a state space representation on $\mathcal{X}_{\text{grid}}$. So, to get the state and the output matrices F and H, we compute the state space representation of the spectral density of K_2, i.e. $S(\omega^2) = \mathcal{F}\left[K_2(\tau)\right]$, which in this case is

$$F = -\sigma_t,\ H = \sqrt{2\lambda\sigma_t}.$$

Finally we compute the matrix $K^{1/2}$ as the Cholesky factorization of the kernel K_1 sampled on the input location of $\mathcal{X}_{\text{grid}}$. However, even if the kernel who generates $f_t(x)$ does not satisfies

Assumption 5.2, it is possible to get a state-space process which approximated the original function $f_t(x)$. Consider in fact the squared exponential kernel that has a spectral density which does not satisfy Assumption 5.2, its density can always be approximated with a rational function exploiting to Taylor expansion or Pade approximation. One example is given by

$$\mathcal{F}\left[K_2(\tau)\right](\omega_t) = \sqrt{\pi}\frac{\lambda}{\sigma_t}e^{-(\frac{w}{2\sigma_t})^2} \approx \frac{\sqrt{\pi}\frac{\lambda}{\sigma_t}}{1+(\frac{w}{2\sigma_t})^2+\frac{1}{2}(\frac{w}{2\sigma_t})^4},$$

and with this approximation we can write a second order state space system.

Assume to have a sampling time of T seconds, the discretization of (5.5) is

$$\begin{cases} s_{k+1}^j = \bar{F}s_k^j + w_k^j \\ z_k^j = \bar{H}s_k^j & j = 1, \dots, M, \\ \mathbf{f}_k = K^{1/2}Z_k \end{cases} \tag{5.7}$$

where

$$\bar{F} = e^{FT},\ \bar{H} = H$$

and w_k^j is a zero-mean white Gaussian noise and $s_0 \sim \mathcal{N}(0, \lambda(H^TH)^\dagger H^TH(H^TH)^\dagger)$.

Example 5.6 We complete Example 5.5 giving the discrete state space representation of a zero-mean Gaussian process $f_k(x)$ with covariance given by (5.6), with sampling time T. In particular Q, H and $K^{1/2}$ do not change while the state amatrix becomes

$$F = e^{-\sigma_t T}.$$

To complete the state model of equation (5.7) and bring it to the form of equation (5.3), we need to provide an explicit measurement model, like the one of equation (5.1). In particular, at every iteration k, we assume to be able to collect m_k measurements of the form (5.1) from different input locations on $\mathcal{X}_{\text{grid}}$. We define the set of selected input locations as $\mathcal{M}(k) := \{x_{\mathrm{j}_1}, \dots, x_{\mathrm{j}_{\mathrm{m_k}}}\} \subseteq \mathcal{X}_{\text{grid}}$. This leads to

$$y_k = C_k\mathbf{f}_k + v_k, \tag{5.8}$$

where $C_k \in \{0,1\}^{m_t \times M}$ is the time-varying output matrix used to select m_k measurements corresponding to the $\mathcal{M}(k)$ input locations. Finally, v_k is an i.i.d. zero-mean Gaussian random vector with covariance $R = \sigma^2\mathbb{I}$, independent from w_k. In Figure 5.1 is depicted the an example of measurments collection on a grid of 5 sensors while in Figure 5.2 is reported a scheme which explain the process formation. Now it is possible to apply Kalman filtering to the state space model of equations (5.7) and (5.8) to compute best unbiased linear estimation of $\mathbf{f}_k$ which exactly correspond to the nonparametric estimate given by equation (5.2). This result is concisely presented in the following

Corollary 5.7 Consider Proposition 5.4 and the state space model of equations (5.7) and (5.8). Then,

$$\hat{\mathbf{f}}_k := \mathbb{E}\left[\mathbf{f}_k|\{x_i, y_i\}_{i=1}^N\right] = \mathbb{E}\left[\mathbf{f}_k|\hat{\mathbf{s}}_{k-1}, y_k\right].$$

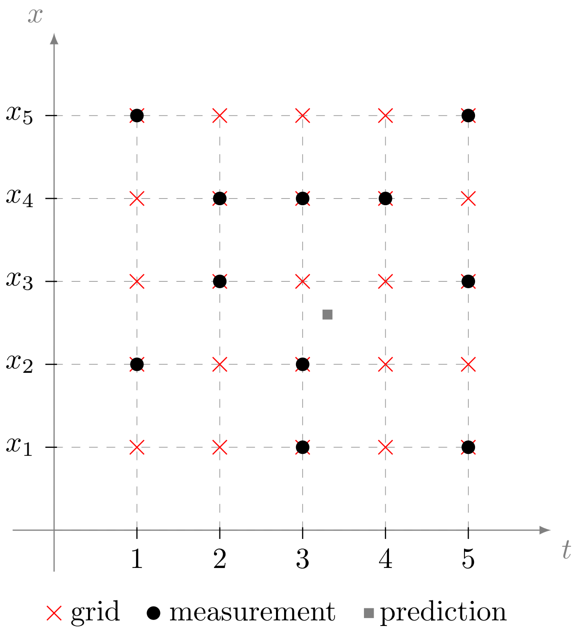


Figure 5.1: This plot shows time versus space. In particular the red crosses highlight the grid input locations, the black circles the measurements taken at each iteration and the gray square an example of prediction off-time and off-grid.

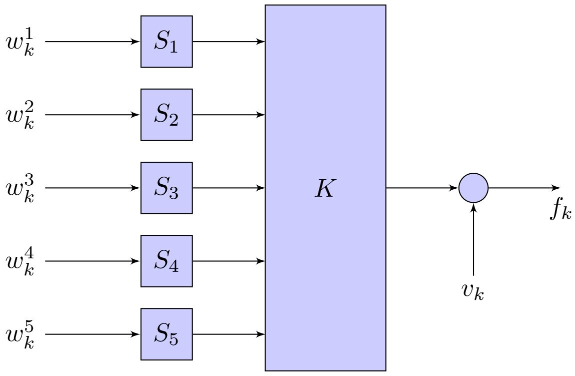


Figure 5.2: This plot shows the process formation in a grid with five input locations.

Remark 5.8 Exploiting the Kalman filter for the estimation makes the complexity cubic on the number of the state of the linear system incited of cubic in the number of measurements. This is incredibly efficient and energy saving when the dealing with big data sets or long time series.

5.4 Numerical Results

In this section we provide some numerical results to validate the performances of the estimation methods proposed. We run all the simulation in MATLAB on a laptop with a 2.4 GHz Intel Core 2 Duo processor and 8 Gb of RAM. For all the simulations we choose a grid with a cardinality of $M = 100$ points equally spaced over the interval $[1, 100]$, a sampling time of $T = 0.2$ seconds.

Example 5.9 First we analyze the estimation of a process that is generate by a kernel which satisfies Assumption 5.2. In particular the kernel used is a sum of exponentials for modeling the time covariance multiplied by a squared exponential for modeling the space covariance, e.g.

$$K(x_1, x_2, \tau) = \lambda(e^{-\sigma_1|\tau|} + e^{-\sigma_2|\tau|})e^{-(x_1-x_2)^2\sigma_x},$$

where $\lambda = 10$, $\sigma_1 = 0.01$, $\sigma_2 = 0.05$ and $\sigma_x = 0.2$. Figure 5.3 shows that the estimation of the last iteration obtained from the standard non-parametric algorithm is exactly the same of the one obtained from the last iteration of the Kalman filter. This figure show only the estimation in a given iteration, so on the x-axis we do not have the time but the space, i.e. the 100 different input locations.

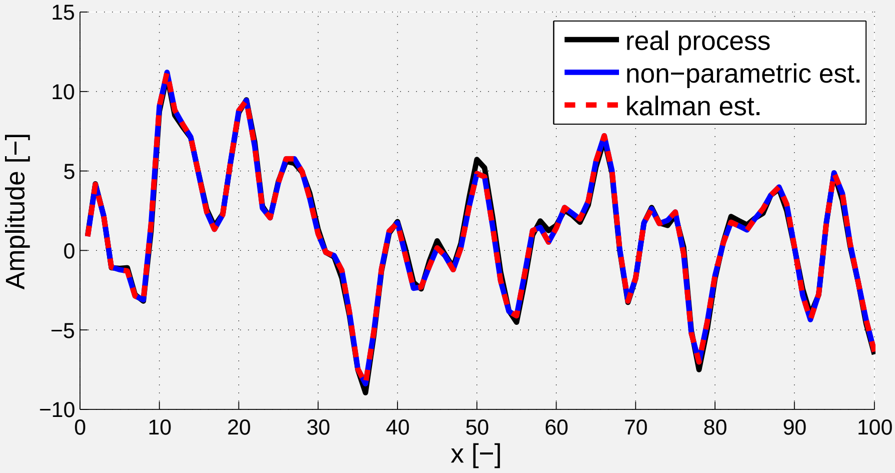


Figure 5.3: In solid black is shown the original process while in solid blue and dashed red are shown the estimation through the non-parametric estimator and the Kalman filter respectively both at the last iteration. The two estimate are exactly the same as stated in the previous section.

Figure 5.4 shows the complete behavior of the Kalman filter over time and space in a $3D$ plot. Obviously this is also the behavior of the non-parametric estimator being the estimate exactly the same.

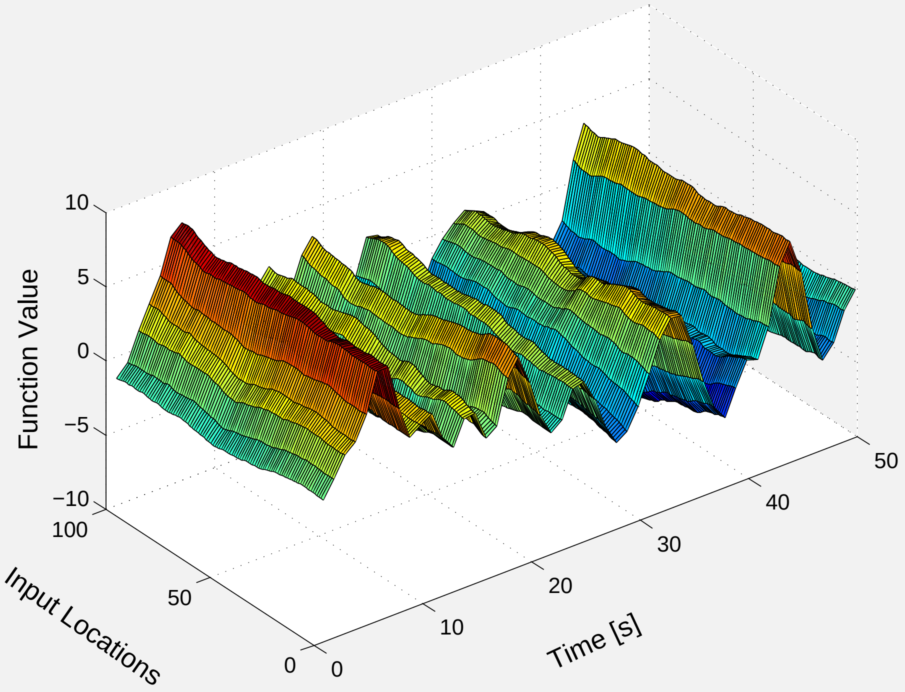


Figure 5.4: Reconstruction in a $3D$ plot of the Kalman estimate.

The second plot in figure 5.5 shows the fit of a non-parametric estimator which uses only a reduced set of measurements with respect to estimate obtained through the Kalman filter. In particular this estimator uses only the last MemoryStesps$*M$ that in our case is equal to MemoryStesps$*100$. The fit is computed as follow

$$\text{Fit}(\hat{f}_1, \hat{f}_2) = \frac{|\hat{f}_1 - \hat{f}_2|}{|\hat{f}_2|}100,$$

where $\hat{f}_1$ is the estimate obtained with the reduced set of measurements and $\hat{f}_2$ is the estimate obtained with the Kalman filter. As we can see when all the measurements are used the fit is, as expected, of 100%, which means that the estimate are the same.

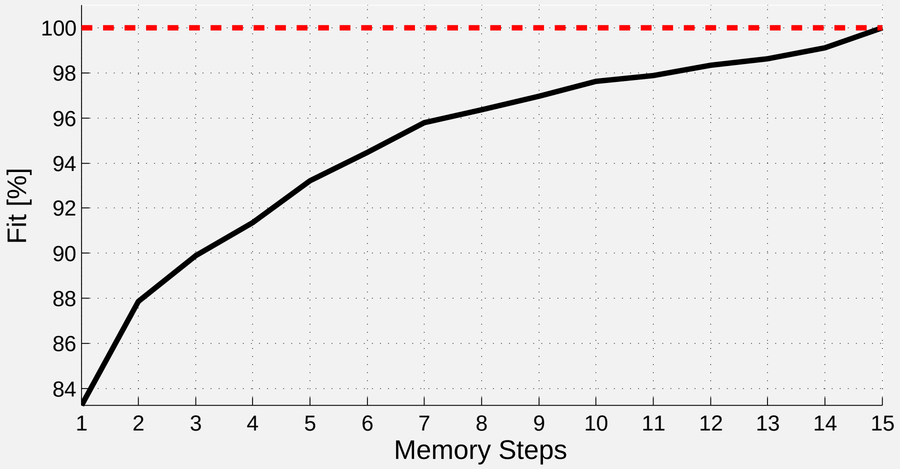


Figure 5.5: In black is shown the fit of the non-parametric estimator which uses a variable amount of memory, e.g. an increasing number of measurements. The red dashed line is the baseline represented by the Kalman filter estimate.

Example 5.10 In this second example we consider the estimation of a process generated by a kernel which not satisfies Assumption 5.2, e.g.

$$K(x_1, x_2, \tau) = \lambda e^{-\sigma_t \tau^2} e^{-(x_1 - x_2)^2 \sigma_x},$$

in particular this power spectral density is not a rational function. As stated in the previous section, a simple way to overcome this issue is to find a good rational approximation of the of the spectral density and a smart way to do that is to solve the following minimization problem

$$\min_{a_0,\ldots,b_m} \left(S(\omega^2) - W(i\omega)\overline{W(i\omega)} \right)^2$$

where

$$W(i\omega) = \frac{b_m(i\omega)^m + b_{m-1}(i\omega)^{m-1} + \cdots + b_0}{a_n(i\omega)^n + b_{a-1}(i\omega)^{n-1} + \cdots + a_0},$$

$S(\omega^2)$ is the p.s.d. associated to the kernel $K(x_1, x_2, \tau)$, $\lambda = 10$, $\sigma_t = 0.2$ and $\sigma_x = 0.5$. In figure 5.6 there is a comparison between the estimate obtained using the Kalman filter and the non-parametric estimator with variable memory. In particular on the x-axis we find for the Kalman filter the order of the approximation while for the non-parametric estimator the number of measurements used for the estimation divided by the number of input locations, e.g. $M = 100$. To be more precise for the Kalman filter we exploited 6 models, for the first order to the sixth order. For the non-parametric estimator the simulations are run starting from 100 measurements to 600. On the y-axis it is shown the fit to our baseline which is the estimate obtained using the non-parametric estimator that uses all the measurements, and it is computed as in the previous example. We can see from the plot that the Kalman filter with a fourth order model already has a fit close to 100%. Even if the two curves are not so straightforwardly comparable the Kalman filter exhibits a good behavior also with low order models while the non-parametric estimator need many data to guarantee good performances.

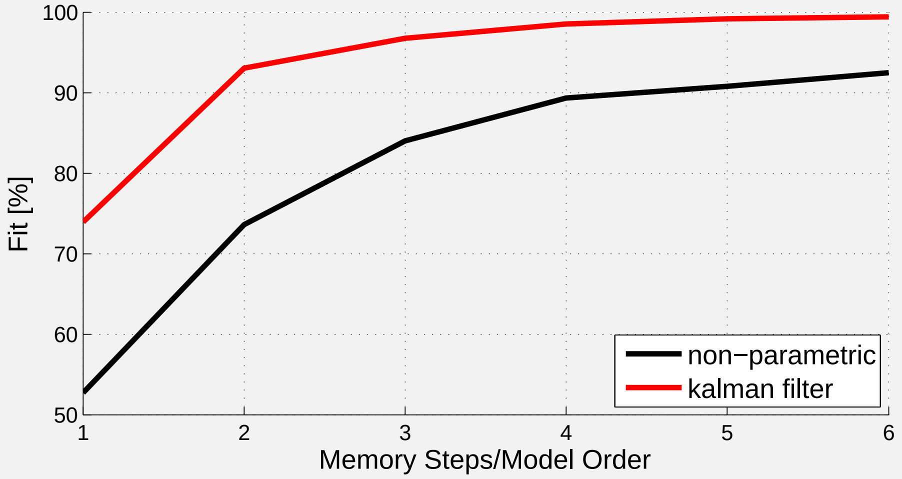


Figure 5.6: In red is shown the fit of the Kalman filter to the non-parametric estimator which uses all the measurements while in black the fit of the non-parametric estimator which uses a reduced set of measurements.

The plot is the result of a Monte Carlo simulation over 1000 runs.

Example 5.11 In this last example we show why is more convenient to use the algorithm that exploit the Kalman filter than the standard Gaussian regression. The reason is due to the fact that the Kalman filter, thanks to its iterative form, it is more efficient from a computational point of view. Consider a process generate by a kernel like the following

$$K(x_1, x_2, \tau) = \lambda(e^{-\sigma_1|\tau|} + e^{-\sigma_2|\tau|})e^{-(x_1-x_2)^2\sigma_x},$$

where $\lambda = 10$, $\sigma_1 = 0.01$, $\sigma_2 = 0.05$ and $\sigma_x = 0.2$. We measured the execution time for the algorithm that exploit the Kalman filter and for the one which relies on the standard Gaussian regression. What we can see is that the complexity of the latter is much higher with respect to the former. In particular from the theory we have the the Gaussian regression has a complexity which is cubic in the number of measurements, that in this case are 100 by the number of iterations. The Kalman filter also has cubic complexity, but it the state dimension which in this particular case is 2, and it has a complexity linear in time. In Figure 5.7 is reported the time needed to execute the estimate for a given number of iterations, e.g. and obviously a given number of measurements. As we can see the Kalman filter is much faster.

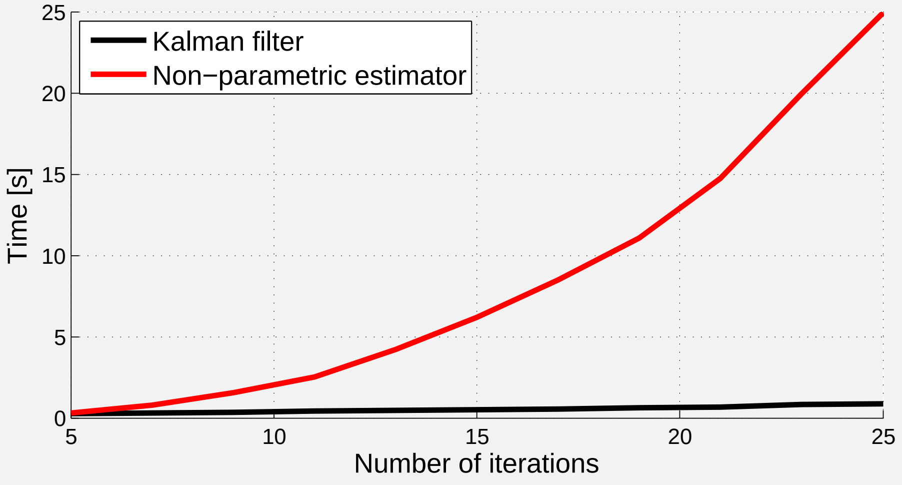


Figure 5.7

The plot is the result of a Monte Carlo simulation over 1000 runs.

6 Conclusions

In this dissertation we propose novel algorithms for the localization and mapping in robotics networks.

In Chapter 2 we propose a consensus based strategy to localize a group of agents using only local relative noisy measurements. This algorithm is thus efficient, scalable and even robust to packet losses and delays. The asynchronous implementation is shown to be exponentially convergent under simple communication protocols. In the randomized scenario, we performed a theoretical analysis of the rate of convergence in mean square, providing general lower and upper bounds.

We presented another algorithm for localization purposes in Chapter 3 which solves the problem of absolute position reconstruction of a multi-robots formation. In particular, it is assumed each agent to be endowed with standard noisy GPS and compass modules and finer relative range and bearing sensors. Combining the absolute and relative information, we showed how the absolute global formation can be reconstructed. Specifically, a fast distributed and asynchronous Linear Least-Squares algorithm which solves an approximation of the Maximum Likelihood estimation problem is presented. Moreover, the algorithm is shown to be robust to delays and packet losses in the communication channel. Exhaustive numerical simulations show how, for sufficiently small relative errors, the approximated solution behaves like the ML estimator. As future research directions, we will investigate the impact of the formation shape and of the communication graph on the relative formation reconstruction. Moreover, a solution which, filtering the absolute and relative angles measurements, could provide a better estimate of the robots absolute rotations will be analyzed.

In Chapter 4 we addressed the problem of simultaneously mapping and coverage with multiple robots. In particular we analyzed the problem of optimal coverage of a region using a formation of robots assuming the sensory field which approximate the sensory distribution of events is unknown. We proposed an algorithm in the context of a client-server architecture which is guaranteed to asymptotically exactly estimate the unknown sensory function and to achieve a partition which is arbitrarily close to a Lloyd partition even in the presence of lossy communication and noisy measurements. We also provided an approximated algorithm that, at the price of a (quantifiable) error in the sensory function estimation, has dramatic reduction in terms of memory and CPU requirements. We also suggested simple guidelines to tune the parameters of the algorithm and we tested the performance of the algorithms via extensive numerical simulations. Future research directions include the extension of this work to dynamic scenarios, i.e. scenarios in which the sensory function is not constant but time-varying, and to scenarios where also the positions of the robots are not perfectly known. Another very challenging avenue of research is the extension of non-parametric estimation tools similar to those adopted here in the context of distributed exploration and partitioning where there is no server involved and the robots have to collaborate only via local lossy communication.

Finally in Chapter 5 we found a class of kernels such that the standard non-parametric estimation can be rewritten as a Kalman filter. In particular we analyzed the problem of estimating functions which varies both in time and space. This is possible thanks to the exploitation of a grid which transform the infinite dimensional problem of the regression to a finite dimensional one. The main advantage of this approach is given by efficiency, e.g. reduced computational complexity, of the Kalman filter. For all the kernels which does not belong to the aforementioned class we also presented an approximation that allows to exploit the algorithm based on the Kalman filter. A future research directions include the extension of the estimation to input locations that does not belong to the grid and time instant that are not integer multiple of the sampling time.

$\mathcal{A}$ Consensus

Consider a graph $\mathcal{G} = (V, \mathcal{E})$, where the set of vertexes $V = \{1, \dots, N\}$ represents the agents and the set of edges $\mathcal{E} \subset V \times V$ represent the connections among the agents. The cardinality of the set $\mathcal{E}$, i.e. $|\mathcal{E}|$, measures how many connections are on the graph. We also define the transition matrix P, which satisfies

$$P_{ij} \geq 0, \qquad \sum_j P_{ij} = 1,$$

which is equivalent to

$$P\mathbf{1} = \mathbf{1},$$

where $\mathbf{1}$ is the vector of all ones. If we consider the discrete time system described by the equation

$$x_{k+1} = Px_k, \tag{A.1}$$

the consenus problem studies how and how fast the components on the state x_k find an agreement, i.e. $x_k \to \alpha\mathbf{1}$ where $\alpha \in \mathbb{R}$ is a constant, varying the matrix P and the underlying graph. The adjacency matrix E is defined as

$$E_{ij} = \begin{cases} 1 & if(i,j) \in \mathcal{E} \\ 0 & otherwise \end{cases}.$$

If we consider the case when $P > 0$, being the transition matrix stochastic, we have that $P\mathbf{1} = \mathbf{1}$, thus 1 is the eigenvalue associated to the right eigenvector $\mathbf{1}$. If we call with w the left eigenvector of P associated to the eigenvalue 1, we can factorize P as follow

$$P = \begin{bmatrix} \mathbf{1} & V \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \Lambda \end{bmatrix} \begin{bmatrix} w^T \\ W \end{bmatrix},$$

where V and W are part of the Jordan transformation matrices and Λ is a matrix containing the eigenvalues of P but one. Now, if we consider the initial problem A.1, we write it as follow

$$x_{k+1} = Px_k = P^k x_0 = \mathbf{1}w^T + V\Lambda^k W$$

Being the eigenvalues on Λ all less than one, when k goes to infinity we have

$$\lim_{k\to\infty} x_k = \lim_{k\to\infty} P^k x_0 = \mathbf{1}w^T x_0 = \alpha\mathbf{1}.$$

As we can see from the previous equation we reach consensus as a linear combination of the initial conditions, and the linear combinator is the left eigenvector of the transition matrix, which is related to coefficients of the matrix P and to the kind of graph that we are using. A similar result can be obtained even if the matrix P is not definite positive, but it is such that there exists an integer k such that $P^k > 0$. A sufficient condition to have convergence to consensus is that the graph $\mathcal{G}$ has to be strongly connected, namely for every pair of vertexes $i, j \in \mathcal{G}$ there exists a path

which connect them.
One important aspect in consensus problems is the speed needed to reach consensus, also called rate of convergence. This kind of analysis is related to the second largest eigenvalue of the matrix P, being the one which drives the dynamics of the transient of the system. One of the possible way to estimate the rate of convergence is to study the dynamics of

$$v_k \triangleq ||x_k - x_\infty|| = ||\left(\mathbf{1}w^t V\Lambda^t W\right)x_0 - \mathbf{1}w^t x_0|| = ||V\Lambda^t W x_0||,$$

when k goes to infinity v_k goes to zero, being the sum of exponential with argument strictly less than one.

B

Kalman Filter

Consider the following discrete time-invariant stochatic system:

$$\begin{cases} x_{k+1} = Ax_k + w_k \\ y_k = Cx_k + v_k \end{cases}$$

where $v_k \sim \mathcal{N}(0, R)$, $w_k \sim \mathcal{N}(0, Q)$ and $x_0 \sim \mathcal{N}(\hat{x}_0, P_0)$. These three variable are Gaussian, not correlated to each other. Moreover we assume

$$\begin{aligned} \mathbb{E}\left[v_k v_h\right] &= R\delta(h-k) \\ \mathbb{E}\left[w_k w_h\right] &= Q\delta(h-k), \end{aligned}$$

where $\delta(x)$ is the Kronecker delta. The Kalman filter is defined as

$$\hat{x}_{k+1|k+1} = \mathbb{E}\left[x_{k+1}|y_{k+1}, y_k, \ldots, y_0\right] = \mathbb{E}\left[x_{k+1}|y_{k+1}, Y^k\right], \tag{B.1}$$

where $Y^k = (y_k, y_{k-1}, \ldots, y_0)$ and its implementation is the recursive formulation of (B.1). Under the assumption aforementioned, it returns the optimal estimate of the state x. The filter has two steps, the first is a prediction of the state based on the model and the previous state

$$\begin{cases} \hat{x}_{k+1|k} = A\hat{x}_{k|k} \\ P_{k+1|k} = AP_{k|k}A^T + Q \end{cases} \tag{B.2}$$

where $\hat{x}_{k+1|k}$ is the estimate of the state and $P_{k+1|k}$ the estimated covariance in $\hat{x}_{k+1|k}$. From equation (B.2) we can see that the estimate depends only on the state and on the model, i.e. it is an open loop estimate, and its goodness will depend on the accuracy of A. The notaion $k+1|k$ means that at time $k+1$ is based on information up to time instant k. From the second equation in (B.2) we can see that the covariance can only increase being the sum of two positive definite matrices. As soon as a new noisy measurement y_{k+1} is available we can proceed with the second step of the Kalman estimation, which is called update. First we compute the so called innovation

$$e_{k+1} = y_{k+1} - C\hat{x}_{k+1|k},$$

which is the difference between what the sensors measure and what the sensors are predicted to measure. Second we compute the Kalman gain

$$K_{k+1} = P_{k+1|k}C^T\left(CP_{k+1|k}C^T + R\right)^{-1}$$

which maps the innovation into a correction for the predicted state. The update state is then given by

$$\begin{aligned} \hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + K_{k+1}e_{k+1} \\ P_{k+1|k+1} &= P_{k+1|k} - K_{k+1}HP_{k+1|k} \end{aligned} \tag{B.3}$$

where the Kalman gain optimally tweak the innvoation in order to improve the estimate. In (B.3) we can notice that, differently from (B.2), the uncertainty decreases since the second term is subtracted to the covariance obtained in the prediction step. To get the best performance from the Kalman filter is importanat to set the correct initial conditions, which are application dependent, i.e.

$$\begin{cases} \hat{x}_{0|-1} = \bar{x}_0 \\ P_{0|-1} = \bar{P}_0 \end{cases} .$$

We can make some useful consideration:

(i) The process can be non-stationary and the system can be time-variant,

(ii) the hypothesis of incorrelation between v_k and w_k can be relaxed,

(iii) the matrix R can be semidefinite positive, but it is necessary to substitute the inverse with the pseudoinverse,

(iv) all the random variables can have a mean different from zero,

(v) the update equation is affine in the measurements,

(vi) all the information up the instant $k-1$ is encapsulated in the state $x_{k-1|k-1}$,

(vii) the gain K_k is time-variant even if the system is time-invariant,

(viii) the signal v_k and w_k must be white and Gaussian.

An alternative formulation of the Kalman filter is given by the Information form which allows to reduce computational complexity when the number of collected measurements, say m, is bigger than the state dimension, say n. This can be computed applying the inversion lemma, and at the end we get two new equations for the update step:

$$\begin{aligned} P_{k+1|k+1} &= \left(P_{k+1|k}^{-1} + C^T R{-}1C\right)^{-1}, \\ \hat{x}_{k+1|k+1} &= P_{k+1|k+1}\left(P_{k+1|k}{-}1\hat{x}_{k+1|k} + C^T R^{-1} y_{k+1}\right). \end{aligned}$$

From the equations of the information filter we can see that now we need to invert the matrix $P_{k+1|k}^{-1} + C^T R{-}1C$ which is of dimension $m \times m$ instead of the inversion of $P_{k+1|k+1}$ which has the dimension of the state, i.e. $n \times n$. So it is convenient to choose the Kalman filter in information form when $m < n$.

C Tikhonov Regularization

Let $f : \mathcal{X} \to \mathbb{R}$ denote an unknown deterministic function defined on a compact $\mathcal{X} \subset \mathbb{R}^d$. Assume to have a measurement model as the following

$$y = f(x) + v,$$

where $x \in \mathcal{X}$ and x is a generic input location in the compact $\mathcal{X}$. Given the data set $\{x_i, y_i\}_{i=1}^N$, where N is the number of measurements collected, one of the most used approaches to estimate f is the so called Tikhonov Regularization, which relies on the Tikhonov regularization theory [64]. The hypothesis space is usually given by given by a Reproducing Kernel Hilbert Space (RKHS) defined by a Mercer Kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. In the following is reported the Representer theorem, which states that the optimal estimate of a function can be represented as a linear combination of basis functions.

Theorem C.1 (Representer theorem) Consider $\Phi : H \mapsto \mathbb{R}$ where H is a generic Hilbert space, defined as

$$\Phi(f) = F\left(L_1 f, \dots, L_m f, \|f\|_H^2\right) \tag{C.1}$$

where $F : \mathbb{R}^{m+1} \mapsto \mathbb{R}$ and where the L_i's are linear and bounded functionals. The map is thus the composition of three different maps: F, the norm in H, and the linear and bounded functionals L_i. The last assumption is that Φ is strictly monotonically increasing w.r.t. the last argument, i.e. it is strictly monotonically increasing w.r.t. $\|f\|_H^2$. Define

$$\widehat{f} = \arg\min_{f \in H} \Phi(f) \ . \tag{C.2}$$

Assume that there exists at least one solution of the previous problem (i.e. that the solution exists but may be not unique). Then it has the form

$$\widehat{f} = \sum_{i=1}^{m} c_i g_i \tag{C.3}$$

where the g_i's are the representers of the various L_i's, i.e. $L_i f = \langle f, g_i \rangle_H$ for all $f \in H$.

The Tikhonov regularization problem is comprised in this formulation, in fact the typical cost function is of the form

$$J(f) = \sum_{i=1}^{N} (y_i - f(x_i))^2 + \gamma ||f||_K^2$$

where $|| \cdot ||_K$ is the norm defined in the RKHS and γ is the regularization parameter that trades off empirical evidence and smoothness information on f. The estimate of the unknown function is

$$\hat{f} = \arg \min_{f \in \mathcal{H}_K} J(f),$$

where $\mathcal{H}_K$ is the associated RKHS. It is known form the literature that $\hat{f}$ admits the structure of a Regularization Network, see [69], being the sum of N basis functions with expansion coefficients obtainable by inverting a system of linear equations. More precisely, one has

$$\hat{f}(x) = \sum_{i=1}^{N} c_i K(x_i, x), \tag{C.4}$$

where the expansion coefficients c_i are obtained as

$$\begin{bmatrix} c_1 \\ \vdots \\ c_N \end{bmatrix} = (\bar{K} + \sigma^2 \mathbb{I})^{-1} \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix},$$

while the matrix $\bar{K}$ is obtained evaluating the kernel at the input-locations, i.e.

$$\bar{K} = \begin{bmatrix} K(x_1, x_1) & \cdots & K(x_1, x_N) \\ \vdots & \ddots & \vdots \\ K(x_N, x_1) & \cdots & K(x_N, x_N) \end{bmatrix}.$$

The estimate of f admits also a Bayesian interpretation in fact, if f is modeled as the realization of a zero-mean Gaussian random field with covariance K, the noise v is Gaussian, independent of the unknown function and with variance σ^2, setting $\gamma = \sigma^2$ one has

$$\hat{f}(x) = \mathbb{E}\left[f(x) | \{x_i, y_i\}_{i=1}^{N} \right]$$

and the associated posterior variance of the estimate at a generic input location $x \in \mathcal{X}$ is

$$V(x) = \text{Var}\left[f(a) | \{a_i, y_i\}_{i=1}^{N} \right] = K(x, x) - \begin{bmatrix} K(x_1, x) & \cdots & K(x_N, x) \end{bmatrix} (\bar{K} + \sigma^2 \mathbb{I})^{-1} \begin{bmatrix} K(x_1, x) \\ \vdots \\ K(x_N, x) \end{bmatrix}.$$

From a computational point of view one can see that to compute $\hat{f}$, $O(N^3)$ operations are required and this approach become unfeasible when the number of measurements increase.

Bibliography

[1] 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. 14, 31, 34, 37

[2] P. Barooah, “Estimation and control with relative measurements: Algorithms and scaling laws,” Ph.D. dissertation, University of California, Santa Barbara, 2007. 14, 21, 31, 43

[3] S. Bolognani, S. D. 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*, vol. 20, no. 2, pp. 176–193, 2010. 14, 36, 38

[4] R. Solis, V. Borkar, and P. R. Kumar, “A new distributed time synchronization protocol for multihop wireless networks,” in *45th IEEE Conference on Decision and Control (CDC’06)*, San Diego, December 2006, pp. 2734–2739. 14

[5] 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 2012. 14

[6] R. Tron and R. Vidal, “Distributed image-based 3-d localization of camera sensor networks,” in *Proceedings of the 49th IEEE Conference on Decision and Control CDC’09*, 2009, pp. 901–908. 14

[7] A. Giridhar and P. Kumar, “Distributed clock synchronization over wireless networks: Algorithms and analysis,” in *Decision and Control, 2006 45th IEEE Conference on*, 2006, pp. 4915–4920. 14, 34, 36, 37

[8] W. Rossi, P. Frasca, and F. Fagnani, “Transient and limit performance of distributed relative localization,” in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, 2012, pp. 2744–2748. 14, 23, 36

[9] C. Ravazzi, P. Frasca, H. Ishii, and R. Tempo, “A distributed randomized algorithm for relative localization in sensor networks,” in *Proceedings of European Conference on Control (ECC’13)*, 2013. 14

[10] A. Zouzias and N. Freris, “Randomized extended Kaczmarz for solving least-squares,” University of Toronto, Tech. Rep., 2013. 14, 34, 37

[11] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, “Coverage control for mobile sensing networks,” *Robotics and Automation, IEEE Transactions on*, vol. 20, no. 2, pp. 243–255, April 2004. 14

[12] J. Cortés and F. Bullo, “Coordination and geometric optimization via distributed dynamical systems,” *SIAM Journal on Control and Optimization*, vol. 44, no. 5, pp. 1543–1574, 2005. 14

[13] N. Leonard and A. Olshevsky, “Nonuniform coverage control on the line,” in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on.* IEEE, 2011, pp. 753–758. 14, 15

[14] P. Davison, M. Schwemmer, and N. Leonard, “Distributed nonuniform coverage with limited scalar measurements,” in *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on.* IEEE, 2012, pp. 1455–1460. 14, 15

[15] F. Lekien and N. Leonard, “Nonuniform coverage and cartograms,” in *Decision and Control (CDC), 2010 49th IEEE Conference on.* IEEE, 2010, pp. 5518–5523. 14

[16] J. Cortés, “Distributed Kriged Kalman filter for spatial estimation,” *Automatic Control, IEEE Transactions on*, vol. 54, no. 12, pp. 2816–2827, 2009. 14

[17] J. Choi, J. Lee, and S. Oh, “Swarm intelligence for achieving the global maximum using spatio-temporal Gaussian processes,” in *American Control Conference, 2008.* IEEE, 2008, pp. 135–140. 14, 16

[18] M. Schwager, D. Rus, and J.-J. Slotine, “Decentralized, adaptive coverage control for networked robots,” *The International Journal of Robotics Research*, vol. 28, no. 3, pp. 357–375, 2009. 14

[19] P. Davison, N. Leonard, M. Schwemmer, A. Olshevsky, *et al.*, “Nonuniform Line Coverage from Noisy Scalar Measurements,” *arXiv preprint arXiv:1310.4188*, 2013. 14

[20] J. L. Ny and G. Pappas, “Adaptive deployment of mobile robotic networks,” *Automatic Control, IEEE Transactions on*, vol. 58, no. 3, pp. 654–666, 2013. 15

[21] P. Van Overschee and B. De Moor, *Subspace identification for linear systems : theory, implementation, applications.* Boston: Kluwer Academic publ, 1996. 15

[22] L. Ljung, Ed., *System Identification (2Nd Ed.): Theory for the User.* Upper Saddle River, NJ, USA: Prentice Hall PTR, 1999. 15

[23] W. Larimore, “Canonical variate analysis in identification, filtering, and adaptive control,” in *Decision and Control, 1990., Proceedings of the 29th IEEE Conference on*, 1990. 15

[24] M. Verhaegen, “Identification of the deterministic part of mimo state space models given in innovations form from input-output data,” *Automatica*, 1994. 15

[25] P. V. Overschee and B. D. Moor, “N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems.” *Automatica*, 1994. 15

[26] ——, “A unifying theorem for three subspace system identification algorithms.” *Automatica*, vol. 31, 1995. 15

[27] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning).* The MIT Press, 2005. 15

[28] F. Lindgren, J. Lindström, and H. Rue, “An explicit link between gaussian fields and gaussian markov random fields; the spde approach,” Tech. Rep., 2010. 15

[29] K. Ni and G. Pottie, “Sensor network data fault detection with maximum a posteriori selection and bayesian modeling,” *ACM Trans. Sen. Netw.*, 2012. 15

[30] P. Hiltunen, S. Särkkä, I. Nissilä, A. Lajunen, and J. Lampinen, "State space regularization in the nonstationary inverse problem for diffuse optical tomography," *Inverse Problems*, vol. 27, 2011. 15

[31] P. Zhu, B. Chen, and J. C. PrÃncipe, "A novel extended kernel recursive least squares algorithm." *Neural Networks*, vol. 32, 2012. 15

[32] J. Quiñonero Candela and C. E. Rasmussen, "A unifying view of sparse approximate gaussian process regression," *J. Mach. Learn. Res.*, 2005. 15

[33] 15

[34] J. Cortes, "Distributed kriged kalman filter for spatial estimation," *Automatic Control, IEEE Transactions on*, Dec 2009. 15

[35] S. Särkkä, A. Solin, and J. Hartikainen, "Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing," *IEEE Signal Processing Magazine*, vol. 30, no. 4, pp. 51–61, 2013. 15

[36] J. Liu, B. Anderson, M. Cao, and A. S. Morse, "Analysis of accelerated gossip algorithms," in *Proceedings of the IEEE Conference on Decision and Control CDC'09*, 2009, pp. 871–876. 15, 28

[37] A. Carron, M. Todescato, R. Carli, and L. Schenato, "An asynchronous consensus-based algorithm for estimation from noisy relative measurements," *Control of Network Systems, IEEE Transactions on*, 2014. 17, 45

[38] ——, "Adaptive consensus-based algorithms for fast estimation from relative measurements," in *4th IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys'13)*, 2013. 17, 38

[39] A. Carron, M. Todescato, R. Carli, L. Schenato, and G. Pillonetto, "Multi-agents adaptive estimation and coverage control using gaussian regression," in *European Control Conference (ECC'15)*, 2015. 17

[40] R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri, "Communication constraints in the average consensus problem," *Automatica*, vol. 44, no. 3, pp. 671–684, 2008. 20, 23, 28, 29

[41] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–913, 2007. 21

[42] F. Garin and L. Schenato, *Networked Control Systems*, ser. Lecture Notes in Control and Information Sciences. Springer, 2011, vol. 406, ch. A survey on distributed estimation and control applications using linear consensus algorithms, pp. 75–107. 21

[43] P. Barooah and J. Hespanha, "Estimation on graphs from relative measurements: Distributed algorithms and fundamental limith," *IEEE Control Systems Magazine*, vol. 27, no. 4, 2007. 23

[44] F. Fagnani and S. Zampieri, "Randomized consensus algorithms over large scale networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 634–649, 2008. 26, 27, 30

[45] R. Carli, A. Carron, L. Schenato, and M. Todescato, "An exponential-rate consensus-based algorithms for estimation from relative measurements: implementation and performance analysis," University of Padova, Tech. Rep., 2013. 29, 31

[46] S. Boyd, A. Ghosh, D. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, 2006. 29

[47] M. Franceschetti and R. Meester, Eds., *Random networks for communication.* Cambridge University Press, Cambridge, 2007. 29

[48] F. Fagnani and P. Frasca, "Broadcast gossip averaging: interference and unbiasedness in large abelian cayley networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 866–875, 2011. 31

[49] M. R. Murthy, "Ramanujan graphs," *Journal of Ramanujan Mathematical society*, vol. 18, no. 6, pp. 1–20, 2003. 31

[50] A. Nedic and A. Ozdaglar, "Convergence rate for consensus with delays," *Journal of Global Optimization*, vol. 47, no. 3, pp. 437–456, 2008. 33, 34

[51] N. I. Fisher, T. Lewis, and B. J. J. Embleton, *Statistical analysis of spherical data.* Cambridge Univ Pr, 1993. 40

[52] P. Langevin, "Sur la théorie du magnétisme," *Journal de physique théorique et appliquée*, 1905. 40

[53] M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration.* Wiley-Interscience, 2007. 40, 47

[54] H. Abbott and D. Powell, "Land-vehicle navigation using gps," *Proceedings of the IEEE*, 1999. 40, 47

[55] R. Racz, C. Schott, and S. Huber, "Electronic compass sensor," in *Sensors, 2004. Proceedings of IEEE*, 2004. 40, 47

[56] J. Pfanzagl, *Parametric statistical theory.* Walter de Gruyter, 1994. 40

[57] H. Wang, G. Hu, S. Huang, and G. Dissanayake, "On the structure of nonlinearities in pose graph slam." in *Robotics: Science and Systems*, 2012. 43

[58] L. Carlone and A. Censi, "From angular manifolds to the integer lattice: Guaranteed orientation estimation with application to pose graph optimization," *IEEE Trans. on Robotics*, 2014. 43

[59] A. Carron, M. Todescato, R. Carli, and L. Schenato, "Distributed localization from relative noisy measurements: a packet losses and delays robust approach," 2014, [Online] Available at http://automatica.dei.unipd.it/people/carron/publications.html. 47

[60] A. Papoulis and S. Pillai, *Probability, random variables, and stochastic processes.* Tata McGraw-Hill Education, 2002. 53

[61] Q. Du, V. Faber, and M. Gunzburger, "Centroidal Voronoi tessellations: applications and algorithms," *SIAM review*, vol. 41, no. 4, pp. 637–676, 1999. 54

[62] S. Lloyd, "Least squares quantization in PCM," *Information Theory, IEEE Transactions on*, vol. 28, no. 2, pp. 129–137, Mar 1982. 54

[63] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *Automatica*, vol. 20, no. 2, pp. 243–255, 2004. 54

[64] A. Tikhonov and V. Arsenin, *Solutions of Ill-Posed Problems.* Washington, D.C.: Winston/Wiley, 1977. 56, 87

[65] F. Cucker and S. Smale, "On the mathematical foundations of learning," *Bulletin of the American mathematical society*, vol. 39, pp. 1–49, 2001. 56

[66] F. Bullo, R. Carli, and P. Frasca, "Gossip Coverage Control for Robotic Networks: Dynamical Systems on the Space of Partitions," *SIAM Journal on Control and Optimization*, vol. 50, no. 1, pp. 419–447, 2012. 60

[67] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, ser. (Adaptive Computation and Machine Learning). MIT Press, 2001. 69

[68] P. S. Maybeck, *Stochastic models, estimation and control. Volume I.*, A. Press, Ed., 1979. 72

[69] F. Girosi, M. Jones, and T. Poggio, "Regularization theory and neural networks architectures," *Neural Computation*, vol. 7, pp. 219–269, 1995. 88

[70] D. Gu and H. Hu, "Spatial Gaussian process regression with mobile sensor networks," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 23, no. 8, pp. 1279–1290, 2012.

[71] B. Lu, J. Oyekan, D. Gu, H. Hu, and H. Nia, "Mobile sensor networks for modelling environmental pollutant distribution," *International Journal of Systems Science*, vol. 42, no. 9, pp. 1491–1505, 2011.

[72] P. Ogren, E. Fiorelli, and N. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *Automatic Control, IEEE Transactions on*, vol. 49, no. 8, pp. 1292–1302, 2004.

[73] K. Lynch, I. Schwartz, P. Yang, and R. Freeman, "Decentralized environmental modeling by mobile sensor networks," *Robotics, IEEE Transactions on*, vol. 24, no. 3, pp. 710–724, 2008.

[74] T. Söderström and P. Stoica, Ed., *System Identification.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988.

[75] T. Chen, H. Ohlsson, and L. Ljung, "On the estimation of transfer functions, regularizations and Gaussian processes," *Automatica*, vol. 48, no. 8, pp. 1525 – 1535, 2012.

[76] G. Pillonetto, A. Chiuso, and G. D. Nicolao, "Prediction error identification of linear systems: A nonparametric Gaussian regression approach," *Automatica*, vol. 47, no. 2, pp. 291 – 305, 2011.

[77] G. Pillonetto, F. Dinuzzo, T. Chen, G. D. Nicolao, and L. Ljung, "Kernel methods in system identification, machine learning and function estimation: A survey," *Automatica*, vol. 50, no. 3, pp. 657 – 682, 2014.

[78] T. Poggio and F. Girosi, "Networks for approximation and learning," in *Proceedings of the IEEE*, vol. 78, 1990, pp. 1481–1497.

[79] S. Smale and D. Zhou, "," *Constructive Approximation*, vol. 26, pp. 153–172, 2007.

[80] ——, "Online learning with Markov sampling," *Analysis and Applications*, vol. 07, no. 01, pp. 87–113, 2009.

[81] F. Bach and M. Jordan, "Predictive low-rank decomposition for kernel methods," *ICML*, pp. 33–40, 2005.

[82] F. Cucker and S. Smale, "On the mathematical foundations of learning," *Bulletin of the American Mathematical Society*, vol. 39, pp. 1–49, 2002.

[83] F. Dinuzzo, "Analysis of Fixed-Point and Coordinate Descent Algorithms for Regularized Kernel Methods," *Trans. Neur. Netw.*, vol. 22, no. 10, pp. 1576–1587, Oct. 2011.

[84] N. List and H. Simon, *A General Convergence Theorem for the Decomposition Method*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2004, vol. 3120.

[85] B. Kulis, M. Sustik, and I. Dhillon, "Learning low-rank kernel matrices," in *In ICML.* Morgan Kaufmann, 2006, pp. 505–512.

[86] N. Leonard, D. Paley, F. Lekien, R. Sepulchre, D. Fratantoni, and R. Davis, “Collective Motion, Sensor Networks, and Ocean Sampling,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 48–74, Jan 2007.

[87] A. Carron, M. Todescato, R. Carli, L. Schenato, and G. Pillonetto, “Multi-agents adaptive estimation and coverage control using Gaussian regression,” *ArXiv*, 2014.

[88] Q. Shi, C. He, H. Chen, and L. Jiang, “Distributed wireless sensor network localization via sequential greedy optimization algorithm,” *IEEE Transactions on Signal Processing*, 2010.

[89] R. Aragues, L. Carlone, G. Calafiore, and C. Sagues, “Distributed centroid estimation from noisy relative measurements,” *Systems & Control Letters*, 2012.

[90] G. Calafiore, L. Carlone, and M. Wei, “A distributed technique for localization of agent a distributed technique for localization of agent formations from relative range measurements,” *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, 2012.

[91] T. Eren, O. Goldenberg, W. Whiteley, Y. R. Yang, A. Morse, B. D. O. Anderson, and P. Belhumeur, “Rigidity, computation, and randomization in network localization,” in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, 2004.

[92] E. Olson, J. Leonard, and S. Teller, “Robust range-only beacon localization,” *IEEE Journal of Oceanic Engineering*, 2006.

[93] K. Dogancay, “Bearings-only target localization using total least squares,” *Signal Processing*, 2005.

[94] G. Mao, B. Fidan, , and B. D. O. Anderson, “Wireless sensor network localization techniques,” *Computer Networks*, 2007.

[95] S. Lupashin, A. Schollig, M. Hehn, and R. D’Andrea, “The flying machine arena as of 2010,” in *Proceedings of IEEE International Conference on Robotics and Automation*, 2011.

[96] R. Ritz, Muller, M.W., M. Hehn, and R. D’Andrea, “Cooperative quadrocopter ball throwing and catching,” in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[97] K. Sreenath and V. Kumar, “Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots,” in *Proceedings of Robotics: Science and Systems*, 2013.

[98] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, “Towards a swarm of agile micro quadrotors,” *Autonomous Robots*, 2013.

[99] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. Mueller, J. Willmann, F. Gramazio, M. Kohler, and R. D’Andrea, “The flight assembled architecture installation: Cooperative construction with flying machines,” *IEEE Control Systems Magazine*, 2014.

[100] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[101] G. Calafiore, L. Carlone, and M. Wei, “Position estimation from relative distance measurements in multi-agents formations,” in *Control Automation (MED), 2010 18th Mediterranean Conference on*, 2010.

[102] G. Picci, *Filtraggio statistico (Wiener, Levinson, Kalman) e applicazioni.* Progetto Libreria, 2007.

[103] K. Yildirim, R. Carli, and L. Schenato, “Proportional-integral synchronization in wireless sensor networks,” *ACM Transactions on Sensor Networks (submitted).*

[104] R. B. Millar, “Maximum likelihood estimation and inference: With examples in r, sas, and admb by russell b. millar,” *International Statistical Review*, 2012.

[105] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: part i,” *Robotics Automation Magazine, IEEE*, 2006.

[106] A. Franchi, G. Oriolo, and P. Stegagno, “Mutual localization in multi-robot systems using anonymous relative measurements,” *The Int. Journal of Robotics Research*, 2013.

[107] S. Lee, Y. Diaz-Mercado, and M. Egerstedt, “Multirobot Control Using Time-Varying Density Functions,” *Robotics, IEEE Transactions on*, vol. 31, no. 2, pp. 489–493, April 2015.

[108] R. Shah, S. Roy, S. Jain, and W. Brunette, “Data MULEs: modeling a three-tier architecture for sparse sensor networks,” in *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, May 2003, pp. 30–41.

[109] A. A. de Menezes Pereira, H. K. Heidarsson, C. Oberg, D. A. Caron, B. H. Jones, and G. S. Sukhatme, “A Communication Framework for Cost-effective Operation of AUVs in Coastal Regions,” in *The 7th International Conference on Field and Service Robots*, Cambridge, Massachusetts, 2009.

[110] R. Patel, P. Frasca, J. W. Durham, R. Carli, and F. Bullo, “Dynamic Partitioning and Coverage Control with Asynchronous One-To-Base-Station Communication,” *Control of Network Systems, IEEE Transaction on*, 2015, to appear.

[111] J. W. Durham and R. Carli and P. Frasca and F. Bullo, “Discrete partitioning and coverage control for gossiping robots,” *Robotics, IEEE Transactions on*, vol. 28, no. 2, pp. 364–378, 2012.

[112] S. Bhattacharya, R. Ghrist, and V. Kumar, “Multi-robot Coverage and Exploration on Riemannian Manifolds with Boundary,” *International Journal of Robotics Research*, vol. 33, no. 1, pp. 113–137, January 2014.

[113] ——, “Multi-robot coverage and exploration in non-euclidean metric spaces,” in *Proceedings of The Tenth International Workshop on the Algorithmic Foundations of Robotics*, 2012.

[114] M. Neve, G. D. Nicolao, and L. Marchesi, “Nonparametric identification of population models via gaussian processes,” *Automatica*, vol. 43, no. 7, pp. 1134 – 1144, 2007.

Acknowledgments

This thesis would not have been possible without the support of several people who, in one way or another, contributed to its completion. I would like to express my gratitude to all those people to give me the possibility to complete it.

First of all I would like to thank Luca Schenato for being a devoted and very competent advisor. I appreciate his dedication and his great capabilities and I hope to have learned a piece of his great knowledge and experience. I would also like to express my gratitude to my co-advisor Ruggero Carli, he is a brilliant and admirable researcher and always able to motivate his students. I wish to acknowledgments also Gianluigi Pillonetto, one of the finest researcher I ever met, with outstanding mathematical capabilities.

I am deeply indebted to my supervisors Antonio Franchi, Elisa Franco and Francesco Bullo for give me the chance to spend more than one year in their laboratories, providing me also guide and advice during all my periods abroad.

A special thanks to Rush Patel for his unbelievable patience, for answering to all my questions and doubts with the fastest and clearest answers.

Thanks to Marco Todescato, a colleague but first a friend. I shared with him most of the time during my Ph.D. He is a very skilled guy always willing to listen. Without him I would never reached these results.

I would like to thank Diego, a mate, a friend for more than 8 years. There are no words to describe how much I am grateful for him to accept, stand and bear me.

A special thanks to Chiara, Giulia, Giulia (Gipsy), Irene, Nicoletta, Giacomo, Michele and Chen. Their arrival brought a breath of fresh air in our office. All the memories of the last year with you are awesome and I hope to have the chance to spend other time with you all.

Thanks also to Fabio, Gian Antonio, Guido and Mattia the experienced post-doc. I have to be grateful also with you if I had the opportunity to start my new and exciting job.

Another special thank to Giulio and Francesco. They are great friends, I will never forget the time spent talking and enjoying together.

It is mandatory mention Simone. My buddy always willing to listen my never-ending problems, to support me with patience and understanding. I have great memories with him and I hope to share much more in the future.

Last but not least, many thanks to my family: without their support, I should have never been where I am now; thank you, for having always struggled to give me the best. In particular I want to say thank you to my parents, Massimo and Giuliana, for all the advices, to my sister Valentina, to my uncles Sergio, Silvia, Marco, Silvia, Mario and Franca and to my grandparents Gina and Fulvio.