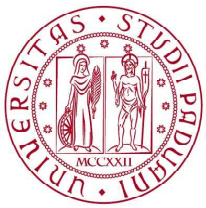

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Sede amministrativa: UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Ingegneria dell'Informazione

Scuola di Dottorato di Ricerca in Ingegneria dell'Informazione
Indirizzo: Scienza e Tecnologia dell'Informazione

Ciclo XXIV

Tesi di Dottorato di Ricerca

Clock Synchronization in Wireless Sensor Networks: Statistical and Algorithmic Analysis

Direttore della Scuola: *Ch.mo Prof. Matteo Bertocco*

Supervisore: *Ch.mo Prof. Lorenzo Vangelista*

Dottorando: *Davide Zennaro*

A Mamma e Papà
- Davide Zennaro -

Table of Contents

List of Figures

List of Tables

Acknowledgements

Padova, Gennaio 2012

Giunto al termine del percorso triennale di Dottorato, desidero ringraziare le persone che mi sono state vicine, certo di cadere nella disgrazia di dimenticare qualcuno.

Devo la maggior parte di quello che ho ottenuto al supporto costante ed incondizionato di Mamma Palmira e Papà Armando, che fin dalle prime decisioni riguardo al mio futuro si sono sempre comportati da genitori modello, lasciandomi piena libertà nelle mie scelte personali, scolastiche e professionali, ma garantendomi allo stesso tempo tutto il supporto di cui avevo bisogno.

Esempi, forse inconsapevoli, per la mia vita sono stati (e sono tutt'ora) i miei fratelli, Tiziana e Umberto e rispettive famiglie, che con la loro perseveranza e coraggio mi permettono di godere di due riferimenti e pietre di paragone del tutto invidiabili.

Ringrazio Prof. Lorenzo Vangelista, per avermi offerto l'opportunità di intraprendere il percorso di Dottorato e per aver sempre creduto nelle mie qualità e capacità, professionali e personali.

Meritevoli di ringraziamenti sono sicuramente tutti i docenti e i colleghi del Dipartimento di Ingegneria dell'Informazione dell'Università di Padova con cui ho avuto il piacere di lavorare assieme. Uno speciale ringraziamento va a tutti i compagni di ufficio che si sono alternati durante questi tre anni.

Un infinito *Grazie* ad un grande Amico, Marco Massari, per esserci sempre stato, in ogni momento. Un affettuoso ringraziamento anche a Tommaso e Martina, per avermi sopportato in questi tre anni, e a Chiara per la costante vicinanza.

Un *Grazie* agli amici di Pellestrina, a tutti quanti, ma in particolare modo ad Anna per essere sempre stata fonte di saggi consigli. Un enorme *Grazie* a Silvia, perché le discussioni via chat ad orari per lei improponibili non rimangano solamente un vago ricordo.

Un forte ringraziamento a Giò, Ermy, Elisa, Sbalky, Matteo e a tutti gli altri amici della combriccola padovana per avermi fatto sentire a tempo di record parte di un gruppo di amici eccezionale. Un sincero ed umile *Grazie*

va in particolare ad Elisa, Ermy e Giò per i fondamentali confronti personali, a volte accesi ma sempre rispettosi.

I would like to thank Prof. Erchin Serpedin for having given me the chance to come to Texas A&M University in 2011 to do research work together with him and Aitzaz Ahmad. A special *Thank You* also to Sabit Ekin and Ali Riza Ekti, for their fundamental help before and during my first days in College Station.

A huge *Thank You* to Patricia ("oh yeah!"), Alberto ("eja"), Darko ("yeah yeah"), Pilar and Cesar for all the gorgeous moments we spent together in College Station and in Northgate in particular. A special *Thank You* to Tasha and Sarah for having tried desperately to teach me how to dance country music. Texas time will definitely hold a special place in my heart.

Infine, vorrei ringraziare tutte quelle persone (colleghi e non) che, in un modo o nell'altro, hanno condiviso con me esperienze fondamentali della mia vita, permettendomi di diventare quello che sono oggi.

Davide

Abstract

In the past few years, the impressive growth of applications performing tasks in a distributed fashion has been enabled by the availability of tiny, inexpensive devices which, in turn, has been made possible by the recent micro-electromechanical technology advancements. Sparsely disposing small intelligent appliances throughout a specific area is something that the community has become used to. Low cost and low power sensing devices equipped with telecommunication hardware are attractive for use in an infrastructure-less network in which the absence of a central node stands out, making robustness be one of the strengths of this kind of networks. Environmental monitoring and military surveillance are just a few examples of the number of applications suitable for sensor networks; in fact, home automation and several health services also can be implemented given that a distributed network of sensors exists.

Sensors need to keep track of a common time scale. This is fundamental for prolonging the network lifetime, making channel access schemes work properly, for example, or for allowing precise duty cycling among the nodes. Clock synchronization is also basic if the goal of a running application is to track moving objects in the battlefield or, more generally, to perform distributed processing of the sensed data. Since the local notion of time in a sensor is based on a low quality local oscillator, it turns out that even small changes in the environmental conditions, like temperature and pressure, lead to modification in the oscillation frequency of the quartz crystals, thus producing time discrepancies among different sensor nodes as time goes by.

This thesis tackles the problem of clock synchronization in sensor networks, both from a perspective of clock parameters estimation and from an algorithmic point of view, to pursue the final goal of making nodes agree on a common time scale, all across the network.

In the first part of the thesis, the so-called *two-way message exchange* between two nodes is thoroughly analyzed. After recalling existing results on clock parameters estimation exploiting data collected via this message exchange process on the wireless channel, an innovative mathematical frame-

work is introduced, which encompasses several common assumptions for the random delays present in the collected data, in a more general treatment. Based on this new framework, a factor graph-based clock offset estimator for wireless sensor networks is proposed and evaluated. Comparison of the variance of the estimation error with classical bounds available in the literature shows that the new estimator has extremely good performance, therefore it can be considered outstanding among Bayesian clock offset estimators.

The focus of the second part of the thesis is on the design of distributed consensus algorithms in wireless sensor networks, especially for observations averaging purposes. In fact, an innovative fast consensus algorithm is proposed and evaluated, based on the *alternating direction multipliers method*, which is a distributed method used to solve minimization problems in an iterative fashion. The new consensus algorithm is compared with the state-of-the-art of fast consensus, showing an excellent convergence rate and an outstanding noise resilience. The proposed algorithm is then applied to solve a network-wide clock synchronization issue, assuming both clock skew and offset for the nodes in the network, showing a relevant performance improvement with respect to previously proposed consensus-based synchronization schemes.

Finally, the Appendix contains a work whose topic falls out of the main stream of this thesis: in uplink cellular networks, based on the knowledge of channel statistics, surrounding base stations are carefully and iteratively chosen in order to provide the mobile terminal a certain quality of service in terms of the maximum allowed outage probability, with the aim of minimizing the overall backhaul network usage.

Sommario

Negli ultimi anni abbiamo assistito alla continua comparsa di applicazioni distribuite, la cui implementabilità risulta consentita dalla possibilità di avere a disposizione sensori piccoli ed economici. I recenti progressi tecnologici nel settore micro-elettronico-meccanico hanno infatti consentito una miniaturizzazione dei nodi sensore. La comunità scientifica si è oramai abituata alla possibilità, con una spesa minima, di collocare piccoli dispositivi intelligenti lungo un'area specifica. Sensori economici e a basso consumo, una volta muniti dell'hardware necessario per le telecomunicazioni, risultano ideali per l'utilizzo in reti senza infrastruttura, uno scenario in cui spicca l'assenza di un nodo centrale e la robustezza diviene quindi una proprietà fondamentale. Monitoraggio ambientale e sorveglianza militare sono solamente un paio di esempi di applicazioni adatte a reti di sensori, così come la domotica e l'ambito sanitario risultano scenari in cui l'uso di una rete distribuita di sensori può rivelarsi, in effetti, utile e vantaggiosa.

I sensori necessitano di una base temporale comune. Questo bisogno risulta fondamentale al fine di prolungare il tempo di vita di una rete, ottimizzando schemi di accesso deterministico al mezzo, ad esempio, oppure schedulando i periodi di attività dei nodi in maniera precisa. La sincronizzazione risulta fondamentale anche in applicazioni legate alla localizzazione, o più genericamente, per permettere l'elaborazione distribuita di dati raccolti dai sensori stessi. Dal momento che la nozione di tempo locale in un sensore è fornita da un oscillatore di bassa qualità, anche minime perturbazioni delle condizioni ambientali (come temperatura e pressione) si riflettono in modifiche nella frequenza di oscillazione del cristallo di quarzo, producendo discrepanze nel comportamento tra oscillatori in diversi sensori, che diventano non trascurabili man mano che il tempo scorre.

Questa tesi affronta il problema della sincronizzazione di clock in reti di sensori, sia da una prospettiva di stima dei parametri di clock, sia da un punto di vista algoritmico lungo tutta la rete, con l'obiettivo finale di permettere ai nodi interessati di trovare una concordanza su una scala temporale comune.

Nella prima parte di questa tesi viene analizzato il processo di scambio di

informazioni tra due nodi chiamato *two-way message exchange*. Dopo aver richiamato la letteratura esistente sulla stima dei parametri del clock utilizzando questo protocollo di scambio dati attravero il canale wireless, viene introdotto un nuovo framework matematico per permettere un'assunzione più generale riguardo i ritardi casuali presenti nei dati raccolti. Basandosi su questo framework, viene proposto e studiato un nuovo stimatore del clock offset basato sulla teoria dei factor graphs. Dal confronto della varianza dell'errore di stima con classici limiti inferiori presenti in letteratura risulta che il nuovo stimatore proposto permette degli ottimi risultati, per cui può a pieno titolo essere considerato meritevole di menzione nella teoria della stima Bayesiana applicata al clock offset.

La seconda parte della tesi riguarda invece la progettazione di algoritmi di consensus distribuiti per reti di sensori wireless, in special modo per operazioni di averaging svolte in maniera distribuita. Viene proposto e valutato un nuovo algoritmo di consensus velocizzato basato su *alternating direction multipliers method*, un metodo distribuito per risolvere problemi di minimizzazione in modo iterativo. Il nuovo algoritmo di consensus viene confrontato con lo stato dell'arte del consensus velocizzato, mostrando un'eccellente velocità di convergenza e una resistenza al rumore migliore rispetto agli altri algoritmi presenti in letteratura. Lo schema proposto viene poi applicato al problema della sincronizzazione di clock in reti di sensori wireless, assumendo presenza di clock skew e clock offset tra i vari oscillatori della rete. L'algoritmo di sincronizzazione risultante consente un rilevante miglioramento delle prestazioni rispetto a schemi di sincronizzazione basati su consensus proposti in precedenza.

Infine, nell'Appendice viene descritto un lavoro il cui argomento si discosta da quello principale della tesi: in reti cellulari in uplink, in base alla statistica del canale le stazioni base cooperanti vengono selezionate tramite l'utilizzo di tecniche iterative con l'obiettivo di garantire al terminale mobile una certa qualità del servizio in termini di probabiltà di disservizio massima permessa e allo stesso tempo di minimizzare l'utilizzo della rete di backhaul.

List of Acronyms and Abbreviations

3GPP: Third Generation Partnership Project

AC: Average Consensus
ACK: Acknowledgement
ADMM: Alternating Direction Multipliers Method
ARQ: Automatic Repeat Request
AS: Analytical Solution
AWGN: Additive White Gaussian Noise

BCHRB: Bayesian Chapman-Robbins Bound
BCRB: Bayesian Cramér-Rao lower Bound
BLUE-OS: Best Linear Unbiased Estimator using Order Statistics
BO: Boyd optimum solution
BS: Base Station
BWC: Broadband Wireless Communication

cdf: Cumulative Distribution Function
CHRB: Chapman-Robbins Bound
CRB: Cramér-Rao lower Bound
CSMA: Carrier Sense Multiple Access

DLL: Delay-Locked Loop
DPLL: Digital Phase-Locked Loop

ES: Exhaustive Search

FC: Flexible Cooperation
FDD: Frequency Division Duplexing
FGOE: Factor Graph-based clock offset Estimator
FTSP: Flooding Time Synchronization Protocol

Ga: Gaussian approximation
GMKPF: Gaussian Mixture Kalman Particle Filter

HARQ: Hybrid Automatic Repeat Request
HARQ-CC: Hybrid Automatic Repeat Request with Chase Combining
HARQ-IR: Hybrid Automatic Repeat Request with Incremental Redundancy

IGMKPF: Iterative Gaussian Mixture Kalman Particle Filter

JD: Joint Decoding

LA: Laplacian
LD: Local Decoding
LTE: Long Term Evolution

MAC: Medium Access Control
MCP: Multi-Cell Processing
MGF: Moment Generating Function
MH: Metropolis-Hastings
MIMO: Multiple-Input Multiple-Output
ML: Maximum Likelihood
MLE: Maximum Likelihood Estimate
MMSE: Minimum Mean Squared Error
MnLD: Minimum Link Delay Algorithm
MSE: Mean Squared Error
MT: Mobile Terminal
MVUE: Minimum Variance Unbiased Estimator

NACK: Not Acknowledgement
NC: No Cooperation
NIC: Network Interface Card
NTP: Network Time Protocol

OSI: Open Systems Interconnection

PBS: Pairwise Broadcast Synchronization
pdf: Probability Distribution Function
PLL: Phase-Locked Loop
ppm: Parts Per Million

QoS: Quality of Service

RBS: Reference Broadcast Synchronization
RF: Radio Frequency
RGG: Random Geometric Graph
ROS: Receiver-Only Synchronization
RRS: Receiver-Receiver Synchronization
RS: Recursive Search

SC: Static Cooperation
SINR: Signal to Interference plus Noise Ratio
SNR: Signal to Noise Ratio
SRS: Sender-Receiver Synchronization

TDMA: Time Division Multiple Access
TOA: Time Of Arrival
TPSN: Timing-sync Protocol for Sensor Networks

UMVUE: Uniformly Minimum Variance Unbiased Estimator
UWB: Ultra-Wide Band

VCO: Voltage Controlled Oscillator

WSN: Wireless Sensor Network

Introduction

In the last few years, the evolution in micro-electronics and micro-electro-mechanical systems technology has lead to the design of tiny devices able to perform tasks in an independent manner and capable of reliable communication with other agents [1]. Thanks to the miniaturization achieved in designing electronic and mechanical components, these devices can *sense* the surrounding environment through on board sensors and can actuate some tasks since they can perform signal processing as they are equipped with micro-controllers. The key feature that makes them interesting from a telecommunications viewpoint is that a group of them placed relatively close to each other can constitute a network, using the available on-board network interface card which allows them to access the wireless channel. These intelligent tiny devices therefore form a wireless sensor network (WSN) through which environmental data is distributively processed and potentially transmitted elsewhere for further uses and for taking decisions.

Recently, many researchers have been focusing on distributed computing thanks to the new focus on distributed networking. Scenarios like WSNs as well as ad-hoc networks [2] and vehicular networks [3] are strongly investigated in these years. In these networks, distributed computing is becoming more and more important, since the single agents are generally entities with poor computational potential and low energy. All nodes are supposed to be equally poor from the computation capabilities viewpoint, therefore there is a strong need to avoid centralized approaches in which a node has to elaborate data coming from other nodes, thus encouraging distributed algorithms. In other words, in these scenarios the assumption of a central sink node is often unrealistic.

The advent of the WSNs makes possible to implement specific distributed applications which could not be performed in an automated fashion before. In fact, having sensors connected through a wireless channel permits a non-invasive but efficient form of environmental monitoring or surveillance, if nodes are equipped with the required sensors or video cameras. Temperature, pressure and soil make-up are just a few examples of monitorable quantities,

as well as vehicles and persons are actual objects whose movements may deserve surveillance. Seismic movements are also often controlled by WSNs [4].

In the last decades, cellular communications have monopolized the attention of the researchers, and synchronization in such networks is generally achieved via a master-slave approach between the mobile terminal and the base station. On the other hand, in the Internet the Network Time Protocol (NTP) [5], which makes use of a network of time servers, is used for clock synchronization among users. Clock synchronization between agents in a WSN is instead a tougher issue, mainly due to the fact that such a network is generally infrastructure-less, therefore neither NTP is well suited nor hierarchical synchronization can always be effectively implemented.

The communication capabilities of sensors allow the whole network to distributively compute functions of the gathered data as well as cost functions useful for future uses and decisions. Assuming that sensors are able to communicate with the surrounding neighbors means that all sensors should have a common notion of time, otherwise channel access schemes (such as time division multiple access, TDMA, for example) cannot be used. The lack of synchronization in this case leads also to a side effect with respect to channel access failure, which is an unwanted power consumption due to unsuccessful packet transmissions and collisions. Algorithms for distributed processing also assume the network nodes to be timely-synchronized, otherwise there would be no agreement on whether deciding if an event happened before or after another. In other words, a common time scale is fundamental also for distributed processing in order to match each event with the time instant it has happened. Another application of relevant interest which bases its functioning on clock synchronization is localization: several methods have been taken into account in order to make sensor nodes in a network able to localize themselves as well as objects on the basis of the knowledge on the position of specific reference nodes. Among the various proposed techniques for localization, the time-based approaches can be exploited to perform node localization under a fundamental assumption: nodes must have the same notion of time. To stress this point, the reader should notice that the localization accuracy in an ultra-wideband (UWB) system using a time-of-arrival (TOA) approach is extremely high [6], therefore fine clock synchronization in such scenarios becomes an extremely important issue. Channel access, distributed processing and localization are not the only applications that assume clock synchronization between sensor nodes: other examples are cooperative communications, coordinated actuation, spectrum sensing and power control, among many others.

The problem of clock synchronization in WSNs has been thorough inves-

tigated in the past. Several surveys [7–10] have been written about this issue, and for a detailed explanation of the problem the reader should also read Elson's Ph.D. dissertation [11]. Many synchronization algorithms (also called *synchronization protocols*) have been proposed in the literature, both for sensor networks that make use of radio frequency (RF) communications [12–18] and for sensor networks based on underwater acoustic transmissions [19–22]. These algorithms either estimate clock parameters with respect to a reference (leading to "uncoupled clocks") [12–15], or actively modify the clock behaviour via suitable control circuits (leading to "coupled clocks") [16–18]. There is a vast literature as well about possible techniques to better estimate the clock parameters among two nodes with specific assumptions on the nature of the network delays [23–31].

This thesis is divided into two parts, after the first chapter which is intended to provide the reader the necessary notions in order to fully appreciate the analytical and simulative results presented after. Part I of this thesis describes clock estimation and synchronization algorithms for WSNs based on the *two-way message exchange*, which allows to estimate the clock parameters and subsequently correct the local clock behaviour, thus leading to uncoupled clocks among the network. Based on data gathered thanks to this exchange process, a general framework is first introduced in order to allow clock offset estimation when the network delays have probability distributions belonging to the exponential family of distributions, thus generalizing existing results in the literature as well as adding new findings. Based on this, estimation algorithms are proposed both for the classical (ML) and the Bayesian framework, using the theory of factor graphs.

The second part of this thesis relies on the application of consensus algorithms for reaching synchronization in WSNs. First, an innovative fast consensus algorithm is presented which allows top performance in terms of convergence speed and outstanding results in terms of noise resilience compared to what is available in the literature. After recasting the clock synchronization issue as a double consensus problem with both clock offset and skew among nodes, the new fast consensus approach is finally applied for reaching clock synchronization, showing that it brings substantial improvements with respect to similar algorithms based on consensus.

Finally, in the Appendix a work whose topic falls out of the main stream of the thesis is reported. In fact, an algorithm for reducing the backhaul network usage in cellular networks is presented, when multi-cell processing is allowed for uplink communications from a mobile terminal, making use of error control policies and several cooperation techniques between the base stations, under a constraint on the quality of service in terms of the maximum allowed outage probability.

Chapter 1

General Concepts on Clock Synchronization

The purpose of this chapter is to give the reader the basics of clock synchronization so that most the fundamental concepts presented and discussed in this thesis can actually find a reference in the present thesis itself.

The chapter deals with the basics of an oscillator in a sensor node, which is the component from which the node gets its own notion of time. After having explained how an oscillator is basically composed, hardware techniques for oscillation frequency correction and control are illustrated. The presentation of a generally accepted clock model follows, which is the foundation for all the results presented in this thesis. Finally, the main characteristics of message-based algorithms are presented as well as the distinction between estimation-based and clock coupling-based algorithms

1.1 The Notion of Time in a Node

When people need to know the current time they consult devices whose specific purpose is to provide an estimate of the time of the day. Looking at the wall clock or at the watch are common ways to answer the question "what time is it?". In sensors, and more generally in every electronic device, this answer is provided through an electric circuit by means of an oscillating signal generated inside. The name of such a circuit is precisely *oscillator*, or *clock*.

There exist natural materials which have a particular oscillating property, namely *piezoelectricity*: in a crystal of a piezoelectric material, a mechanical stress produces an accumulation of charges on the opposite sides of the crystal. In this sense, the material is actually behaving like a capacitor: by

connecting the two sides with an electric circuit, the flow of electric current is induced. In more general terms, the crystal behaves like a RLC circuit (resistor, capacitor and inductor) with a certain frequency of resonance, typical of the material. If the mechanical compression is done periodically at such a frequency, then an output periodic electric signal at the frequency of resonance is obtained. This is precisely the oscillating signal provided by a clock in the electronic device.

A piezoelectric material largely used in oscillators is quartz. A common oscillation frequency for quartz oscillators is 32768 Hz, as it happens, for example, in the microprocessor Texas Instrument MSP430 F1611 [32] embedded in Tmote Sky sensors. A clock oscillating at 4 MHz can be found instead in MICA motes, which are employed in testbeds at the University of California, Berkeley [33].

Although they are claimed to run at the same nominal frequency, even nominally identical crystal oscillators have slightly different running behaviours, mainly due to fabrication issues and climate conditions. Indeed, it happens that their instantaneous phases could be different and generally time-varying. Given that an estimate of the phase difference is possible, what is needed right after is a way to implement a phase correction to the oscillating circuit. The application of a voltage correcting signal to an oscillator for modifying its running behaviour is exactly what happens in a *Voltage Controlled Oscillator* (VCO). The feedback-based circuits that generate such correcting signals are generally referred as *(digital) phase-locked loops* ((D)PLLs) or *delay-locked loops* (DLLs). For a detailed description of these circuits see [34].

In an actual application of clock synchronization, by means of a PLL the node i is able to synchronize its own clock to an external signal produced through the interaction with other nodes' oscillators. Designing the nature of such an interaction between nodes means providing a clock synchronization algorithm, which is precisely what is being described in the present thesis.

1.2 Clock Models

The main building block in a synchronizing network of agents is the clock, which can be defined as follows.

Definition 1.1 *A* clock *is an electronic device that counts oscillations in an accurately-machined quartz crystal, at a particular frequency [7].*

Therefore, a clock can be considered a timer which increases of a unit at every clock tick, at a certain frequency.

The observable value at sensor node i is the clock counter at absolute time t, i.e.,

$$T_i(t) \tag{1.1}$$

which is often simply denoted by "clock" of node i at time t.

Each oscillator runs at frequency $f_i(t)$, generally different from the oscillator nominal frequency f_0. The clock oscillation period (the inverse of the oscillation frequency) is defined as

$$\Upsilon_i(t) = \frac{1}{f_i(t)} \, . \tag{1.2}$$

As far as the clock accuracy is concerned, it is generally known [8] that oscillators running at higher frequencies are more stable, while oscillators at lower frequencies consume less power, so that it is common practice to have sensors equipped with slow running oscillators for energy consumption purposes. This comes at the price of lower clock stability and less achievable accuracy, since the time resolution is coarser. Indeed, changes in environmental factors like temperature, pressure and humidity make the oscillation frequency of clock of node i, $f_i(t)$, vary with time. Also, factory imperfections make different clocks oscillate inherently at different frequencies. Therefore, clock discrepancies between nodes are due both to different values of the clock counters and to slightly different oscillation frequencies.

1.2.1 Clock Parameters

Terms like clock *offset*, *skew* and *drift* are commonly used to describe the differences in running behavior between clocks of any two nodes i and j. These concepts are defined in the following, coherently with the definitions provided in [7, 35].

The clock offset of node i relative to node j is the difference between the clock counters of the two nodes, i.e.,

$$\theta_{ij}^{(o)}(t) = T_i(t) - T_j(t) \, .$$

In many cases in the present thesis, the key parameter is the *initial clock offset* between i and j, which is defined as the clock offset between the two nodes at time 0, i.e.,

$$\theta_{ij}^{(o)} = \theta_{ij}^{(o)}(0) = T_i(0) - T_j(0) \, . \tag{1.3}$$

If node j is the reference time, it turns out that the clock offset of node i with respect to the reference time is simply

$$\theta_i^{(o)}(t) = T_i(t) - t \ ,$$

while the initial offset of node i is

$$\theta_i^{(o)} = \theta_i^{(o)}(0) = T_i(0) \ , \qquad \theta_i^{(o)} \in \mathbb{R} \ . \tag{1.4}$$

The clock skew of node i relative to node j is the difference of the first derivative of the clock counters of nodes i and j, i.e.,

$$\theta_{ij}^{(s)} = \frac{\partial T_i(t)}{\partial t} - \frac{\partial T_j(t)}{\partial t} \ . \tag{1.5}$$

If node j is the reference time, it turns out that the clock skew of node i with respect to the reference time is simply

$$\theta_i^{(s)} = \frac{\partial T_i(t)}{\partial t} - 1 \ , \qquad \theta_i^{(s)} \in [-1, +\infty) \ . \tag{1.6}$$

The clock drift of node i relative to node j is the difference of the second derivative of the clocks of nodes i and j, i.e.,

$$\theta_{ij}^{(d)} = \frac{\partial^2 T_i(t)}{\partial t^2} - \frac{\partial^2 T_j(t)}{\partial t^2} \ . \tag{1.7}$$

If node j is the reference time, it turns out that the clock drift of node i with respect to the reference time is simply

$$\theta_i^{(d)} = \frac{\partial^2 T_i(t)}{\partial t^2} \ , \qquad \theta_i^{(d)} \in \mathbb{R} \ . \tag{1.8}$$

1.2.2 The Adopted Clock Model

In order to describe the clock dynamics, several models have been proposed in the literature. The most complete modeling of a clock consists of a Taylor expansion of the clock counter truncated at the second degree, i.e., [7, 35, 36]

$$T_i(t) = \theta_i^{(o)} + \left(1 + \theta_i^{(s)}\right) t + \theta_i^{(d)} \frac{t^2}{2} \ .$$

In this thesis, clock drift $\theta_i^{(d)}$ is not considered so that the considered reference model is the following [7–10, 12–16, 23–31, 35, 37–43] (either with or without the skew presence)

$$T_i(t) = \theta_i^{(o)} + \left(1 + \theta_i^{(s)}\right) t \ , \tag{1.9}$$

in which $\theta_i^{(o)}$ and $\theta_i^{(s)}$ are unknown parameters to be estimated. Often, an a-priori description for these parameters is also available. In each chapter of this thesis it is specified if clock skew $\theta_i^{(s)}$ is also considered, after clock offset $\theta_i^{(o)}$, which is always taken into account. Fig. 1.1 shows the running behaviours of different clocks adhering to the model (1.9).

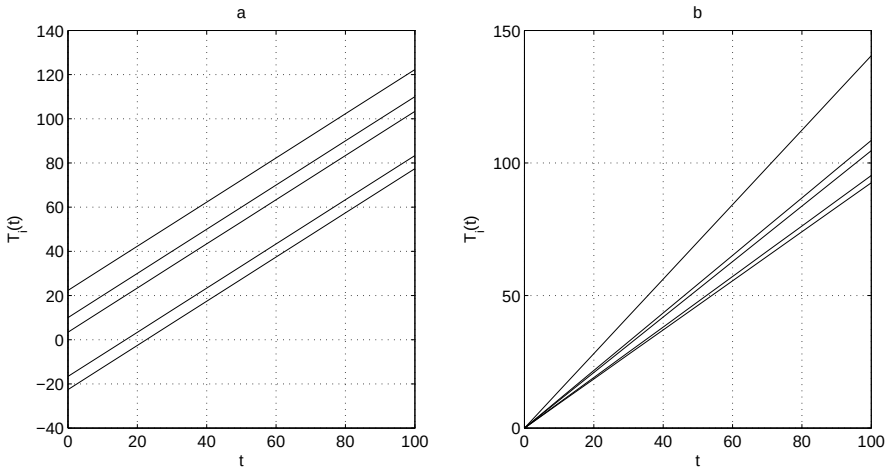

Figure 1.1: Examples of clocks with different offsets and no skews (a) and with different skews but no offsets (b).

Clock offsets and skews are real values, describing the level of discrepancy between two clocks. However, they have strongly different meanings and relevance, since a relatively small clock skew can lead to a huge clock offset after a long period of time in which no synchronizing correction is made. As a result, if having an unbounded initial clock offset could not heavily affect synchronization, even a small clock skew requires frequent re-synchronizations if it is not taken into account during the design of the algorithm.

In real sensor networks, values of clock skews are generally bounded and quite small. The community is used to express the skew in terms of parts per million (ppm): if a clock is said to deviate c parts per million, it means that after a million of clock ticks it diverges of the equivalent of c clock ticks. To be concrete, the clock embedded in the microprocessor Texas Instrument MSP430 F1611 [44] has a clock skew on the order of 40-50 ppm, which means that for each second the clock deviates of 40-50 μs. Also other datasheets [45] report skews of this order of magnitude. If deviating of a few μs every second could appear as a negligible effect, just realize that in a day ($24 \times 60 \times 60 = 86400$ seconds) the oscillator deviates of few seconds, which is a huge time from a sensor network application viewpoint.

Absolute or Relative Synchronization?

In clock synchronization, identifying the known and unknown quantities is a matter of fundamental importance. In the model (1.9), $T_i(t)$ represents the observation at node i at time t; on the other hand, the time scale at node i is $T_i(t)$ itself, so that the sensor node is totally unconscious of the absolute time reference t. In fact, it can only access the counter $T_i(t)$. As a result, by itself the node is not able to estimate its own clock parameters $\theta_i^{(o)}$ and $\theta_i^{(s)}$, unless by using time-related observations coming from other nodes. Moreover, since timestamps recorded by other nodes are related to these nodes' timescales, an estimate of the absolute values $\theta_i^{(o)}$ and $\theta_i^{(s)}$ is generally not possible, except for the case in which node i uses readings from a special node whose timescale is assumed as the reference, i.e., $\theta_i^{(o)} = \theta_i^{(s)} = 0$.

1.3 Approaches to Clock Synchronization

Clock synchronization between agents requires information exchange between nodes in order to pursue a common time basis. The main question now is *what kind of data should nodes exchange in order to reach synchronization?* Existing clock synchronization algorithms either make use of messages (OSI level 2 data exchange) or electromagnetic radio frequency pulses for exchanging clock related data (OSI level 1 data exchange). The two approaches lead respectively to *message-based* and *pulse-coupled* algorithms. The comparison between them is done in the next section.

1.3.1 Message-Based vs Pulse-Coupled Algorithms

Depending on the available hardware, the target level of accuracy and the channel resources, nodes can either exchange message-based data or physical signaling [9]:

Message-Based Algorithms : in message-based algorithms, the local time of a node is written (e.g. *timestamped*) in a message, which is broadcast to the neighboring nodes via the wireless channel. Since they involve data exchange at level 2 of the OSI stack, such algorithms suffer of software delays, such as interrupts or operating system calls. However, it is proved that most of such delays can be effectively neglected through kernel [12] or MAC-layer timestamping [13, 14], i.e. reading the timestamp and adding it to the packet just before the transmission on the wireless medium. A fundamental advantage of such algorithms is that

they don't require the network to allocate specific resources (such as bandwidth) for synchronization purposes.

Pulse-Coupled Algorithms : pulse-coupled algorithms are based on physical layer signaling instead of message transmissions; in fact, time information is encoded in the transmission instants of the sent waveforms. To this purpose, generally either a specific frequency band is dedicated or an overlay system (such as UWB) is employed. The received signal at neighboring nodes is then processed in order to achieve synchronization with the sending node. Such processing techniques possibly include the use of PLLs [16, 37].

Although pulse-coupled approaches are interesting from the signal processing point of view, this thesis entirely focuses on message-based clock synchronization algorithms, since it is assumed that neither specific channel resources are reserved for synchronization purposes nor sensor nodes are equipped with UWB transceivers. Basic wireless resources and simple communication hardware are well suited constraints to low-cost low-power sensing devices.

1.3.2 Coupled Clocks vs Uncoupled Clocks

In the realm of message-based algorithms, synchronization methods can be divided in two categories[1]. Depending on the type of data processing and clock correction, message-based algorithms can lead either to *coupled clocks* or to *uncoupled clocks* [9]. Indeed, the algorithms can either estimate the clock parameters with respect to the other nodes and then perform a correction through a VCO (uncoupled clocks), or act directly to the oscillator circuit to correct it step by step (coupled clocks). Regarding this aspect, the difference between uncoupled and coupled clocks reflects also the distinction between Part I and Part II of this thesis: in fact, in Part I clock synchronization algorithms are designed with the goal of estimating clock parameters, so that clocks in the network result to be uncoupled, since no correction signal is applied to the oscillator circuit before the estimation algorithm stops running; on the other hand, the synchronization algorithm derived in Chapter 7 in Part II relies on the application of a correction signal to the oscillator on the fly at each step, therefore leading to coupled clocks along the network. It can also be observed that, while in Part I the focus is on the statistical detailed description of the message exchange process between two nodes, Part

[1] The distinction is actually transversal with respect to the categorization message-based vs pulse-coupled.

II provides a broader picture of the problem, better analysing the overall network performance while lacking of local detail.

1.3.3 Synchronization Via Parameter Estimation

Clock synchronization through parameter estimation can be performed via time-related data exchange. The following definition holds.

Definition 1.2 *The process by which nodes exchange time-related information in message-based algorithms is called a synchronization* paradigm.

To the best of the author's knowledge, three synchronization paradigms can be identified [10, 15]:

Sender-Receiver Synchronization - SRS : in this classic paradigm two nodes S (the sender) and R (the receiver) are involved; they exchange timestamped messages in a two-way handshake fashion (started by node S) in order to make node S able to estimate its clock relative parameters with respect to node R.

Receiver-Receiver Synchronization - RRS : at least three nodes are involved in this paradigm, the parent node P that sends broadcast messages, and two other nodes, R_1 and R_2. After that, R_1 and R_2 exchange data about the local time of reception of the messages and estimate their clock relative parameters.

Receiver-Only Synchronization - ROS : a node L is able to synchronize to node S by listening to the messages exchanged by node S and node R that are synchronizing using the SRS paradigm.

Part I of this thesis is dedicated to the analysis of clock synchronization algorithms based on clock parameters estimation which make use of the SRS paradigm.

1.3.4 Synchronization Via Clock Coupling

Some network-wide clock synchronization algorithms that perform a step by step clock correction assume a low level of detail in the communication model but allow for a more comprehensive analysis and provide a broader picture of the algorithm behaviour. Synchronization algorithms that fall in this category and are taken into account in this thesis are based on the consensus approach. Basics of consensus algorithms as well as the application of consensus to the problem of clock synchronization are described in Part II.

Part I

Clock Synchronization in WSNs via the Two-Way Message Exchange

Clock synchronization can be reached through the estimation of the parameters that identify the disagreement between node's clocks. This is the approach considered in Part I of the present thesis.

Chapter 2

The Two-Way Message Exchange

In Chapter 1, message-based protocols have been divided in two categories, based on the fact that they lead either to coupled or to uncoupled clocks. The current chapter describes the process of data exchange exploited in estimation-based clock synchronization algorithms, that lead to uncoupled clocks. This communication protocol provides the rules for getting the time-related information needed for clock parameters estimation. The obtained observation model is carefully analysed and its performance studied.

2.1 The Two-Way Data Exchange

One of the preferred ways to exchange time related data in a message-based algorithm is the so called *two-way message exchange*, which identifies a two-way communication handshake protocol between two nodes sharing their timestamps in order to achieve synchronization.

The two-way message exchange is a data communication process between two nodes in which there is a double message exchange between the involved nodes, one for each link direction. The messages carry timestamped information according to the local clock time scale, so that at the end of the information exchange there is sufficient data to infer the clock disagreement parameters of the two nodes, making clock synchronization possible between them.

The mechanism is not perfectly symmetric, since there is a node which initiates the process and the other which responds. On the other hand, with little overhead (just one more short message, which can also exploit piggybacking of following messages) at the end of the data exchange both the nodes get all the raw information available, thus they can perform the same data processing to clock parameters estimation.

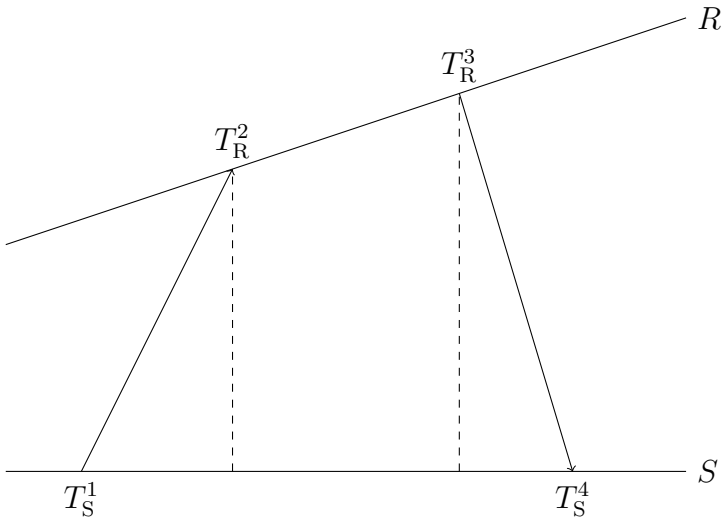

Figure 2.1: Two-way message exchange between node S and R.

2.2 System Model

The two-way message exchange can be better described if one looks at Fig. 2.1. The lowest horizontal line represents the time scale of node S, which is the node that begins the data exchange process. The upper oblique line instead represents the time scale of node R. The non-zero angle between the two lines represents the clock skew between the time basis of node S and node R.

The data exchange [23, 25–31, 38–42] starts with a message sent by node S, containing the timestamp T_{S}^1 of the time at which such a message has been sent[1], measured according to the time scale of node S. Such a message is received by node R at time T_{R}^2, according to the time scale of node R. After a predefined period of time, node R replies with a message containing the timestamps T_{S}^1, T_{R}^2 and the timestamp T_{R}^3 relative to the sending time of this message, measured according to the time scale of node R. Finally, such a message arrives to node S at time T_{S}^4, according to the time scale of node S.

By keeping into account of both clock offset $\theta^{(\mathrm{o})}$ and skew $\theta^{(\mathrm{s})}$ between the two nodes using the model (1.9), the following relations between the

[1]The dependence on the absolute time of the counter values $T(t)$ is dropped in order to make the notation more readable.

timestamps hold [26, 27, 31, 38]:

$$T_{\mathrm{R}}^2 = \left(1+\theta^{(\mathrm{s})}\right)\left(T_{\mathrm{S}}^1 + d + X\right) + \theta^{(\mathrm{o})}$$
$$T_{\mathrm{R}}^3 = \left(1+\theta^{(\mathrm{s})}\right)\left(T_{\mathrm{S}}^4 - d - Y\right) + \theta^{(\mathrm{o})} \ . \tag{2.1}$$

The fixed portion of the link delay is assumed symmetric and denoted by d, while the random portions of the same delay in uplink and downlink respectively are denoted by X and Y. A full discussion about these delays as well as their possible modeling is performed in the following.

2.2.1 Fixed and Random Portions of Link Delay

The major impairments in the two-way message exchange come from the message transmission delays. There is no need to stop and try to find a possible cause of delay that could affect a message transmission through a shared medium. The wireless channel is indeed used by a potentially large number of users, and there is an extensive bibliography on the available channel access schemes. For sure, the channel access is neither deterministic nor instantaneous. In other words, there is a stochastic delay to be modelled.

The delay encountered by a message transmission through a wireless channel has been decomposed into four categories [10, 12–14]:

- *Send Time*: the time required by the transmitter to construct the message and to forward it to the NIC[2]. This time includes delays coming from interrupts and operating system calls.
- *Access Time*: the time required to get a successful message transmission in the wireless medium. It depends on the channel access scheme.
- *Propagation Time*: the time needed for the electromagnetic wave to travel on the wireless medium from the transmitter to the receiver. It encompasses latencies and forwarding delays at intermediate hops in large networks.
- *Receive Time*: the time needed to the receiver for the message reception and decoding as well as for the notification of the message arrival to the operating system.

These delays are typical of a large class of wireless networks and protocols, from the TDMA one-hop networks to the large multi-hop CSMA networks. In

[2]In the following, the NIC (Network Interface Card) is intended as the part of the device which is actually sending the packet over the air.

this thesis, for the delay analysis in the two-way message exchange, only one-hop communications will be considered and CSMA is assumed as the wireless channel access scheme [33]. With this assumption, the just mentioned four delays can be divided in two categories. Assuming nodes are not mobile, the propagation delay can be considered *fixed* (or *deterministic*), since it just depends on the distance between the two sensors, no random components are present. However, the send, access and receive times encompass random impairments among their causes (operating system calls, channel access, ...), therefore they can be considered *random* (or *stochastic*). In this thesis, the following definitions hold (see [46]):

Definition 2.1 *The* deterministic (fixed) *part of the link delay in the two-way message exchange process is the minimum of the link delays over the message exchange rounds.*

Definition 2.2 *The* stochastic (random) *part of the link delay in the two-way message exchange process is the positive quantity to be added to the deterministic part of the link delay in order to get the overall delay.*

As a result, both the deterministic and the stochastic delays are positive quantities.

In the past, a measurement campaign has been performed [46] to see if the end-to-end delay could be somehow modelled. What turned out from such work is that the stochastic end-to-end delay can be better described by a Gamma probability distribution, which comprises a number of distributions. What plays a key role in the stochastic part of the delay is the queueing delay at the transmitters; this is the reason why many works in the literature assumed exponential distributed random portions of delays [23–27]. However, the Gaussian, log-normal and Weibull distributions are other good candidates adopted in the literature for modelling the stochastic delay. Given the substantial diversity between these distributions, it is hard to find a common framework which encompasses the entire range of possible distributions. On the other hand, in Chapter 3 a new framework is presented in order to allow for the study of different random delay distributions at the same time in a more general fashion, through the use of tools borrowed from the theory of the exponential family of distributions.

Message delays can be reduced through smart tricks. The most used techniques involve reading the timestamps just before forwarding the message to the NIC for the actual transmission (kernel [12] or MAC layer timestamping [13,14]) to eliminate the send and the access times. Moreover, if also the receiver timestamps the message arrival time at a low enough level in the host's operating system kernel, the receive time is strongly reduced since it

does not include the overhead of system calls, context switches, or even the transfer of the message from the network interface to the host. Finally, the propagation time cannot be eliminated because of its physical reason, but it can be estimated from the round trip time through a handshake paradigm.

2.3 State of The Art

Several clock synchronization algorithms exploit the two-way message exchange in order to reach clock agreement. Indeed, as an example, both Timing-sync Protocol for Sensor Networks (TPSN [14]) and Reference Broadcast Synchronization (RBS [12]) use data coming from this exchange process in order to compensate for clock offset and skew.

The two-way message exchange well applies to a statistical analysis of timestamped data, both in presence of clock offset and skew, as it can be seen from (2.1). A major statistical analysis of the two-way message exchange mechanism has been done first in [23], where a thorough discussion on the best use of the timestamped data is also presented. After this first work, a deep study on the statistical performance of this message exchange process for clock parameter estimation has been done in the literature.

If clock skew is not considered, the issue is to correctly estimate the clock offset between nodes S and R. Assuming the random portion of delays being identically distributed exponential random variables both in uplink and downlink, Ghaffar in [23] states that an ML estimate (MLE) of the clock offset does not uniquely exist in the case in which both the mean of the random portion of the delays and the fixed portion of delay are known. On the other hand, Jeske in [24] proved that such an MLE actually exists when the fixed portion of delay is unknown, with the mean of the exponentially distributed delays being either known or unknown; an expression for the MLE is therein derived. However, several *ad-hoc* clock offset estimators were proposed in [23]; one of them (the *Minimum Link Delay Algorithm*, MnLD) coincides with the actual MLE derived in [24]. Still assuming that the random delay distribution is exponential but, more generally, asymmetric between uplink and downlink, in [25] the best linear unbiased estimator using order statistics (BLUE-OS) [47] is derived as well as the minimum variance unbiased estimator (MVUE) [47] via the application of the Rao-Blackwell-Lehmann-Sheffé theorem. Making the assumption that the stochastic part of the delay is the result of a diverse serie of causes, [26] considers it as following a Gaussian distribution. The applicability of such an hypothesis could be questionable, because if so, there is a non-zero probability that the overall delay (deterministic plus stochastic) is negative and this is physically

not possible. On the other hand, the Gaussian assumption may hold in the case the standard deviation is chosen small enough, by recalling that about the 99.7% of the samples are inside the 6σ interval symmetric around the distribution mean. The authors in [26] provide expressions for the MLE for the clock offset and the relative Cramér-Rao lower bound (CRB) under this Gaussian assumption. The case of a Weibull distributed random delay is faced instead in [28], where it is shown that the problem of finding the MLE of the clock offset is generally convex, therefore it is solvable by numeric optimization A uniform MVUE (UMVUE) is also derived for the exponential distribution case in which the fixed portion of delay is known. When a more general assumption is made about the noise distribution, it becomes hard to find an exhaustive solution to the clock offset estimation problem. In [29,30] such a difficulty has been faced by the usage of (Iterative) Gaussian Mixture Kalman Particle Filter, (I)GMKPF, in order to find an estimate of the clock offset in the presence of non-Gaussian and non-exponential random delay distribution or in the presence of a mixture of several distributions.

If the model assumes the presence of both clock skew and offset among node's oscillators, frequent re-synchronization is reduced since a skew correction naturally leads to a longer clock synchronization. A Gaussian distribution for the random portion of delay is assumed in [31] where the joint MLEs for clock offset, skew and fixed part of the delay are derived as long as performance bounds. Moreover, a lower complexity suboptimal estimator is also proposed. In [26] the joint MLE for both clock skew and offset is derived in the case of Gaussian distributed random delays with the assumption of knowing the fixed part of delay. Since in this case the estimator expression involves non-trivial computation (and an MLE for the exponential case does not even exist if d is known), an ML-like estimator is proposed for both the Gaussian and the exponential cases, which exploits just the first and the last time stamps. This estimator assumes d to be unknown, which is a more realistic assumption in real WSNs. Performance is obviously worse than the MLE but a simpler expression is provided for the estimators so that they are easily implementable. Later in [27] a graphical algorithm to find the ML clock offset and skew estimate is derived in the case of exponential symmetric random delays. Due to its complexity, a suboptimal alternative empirical algorithm is proposed.

2.3.1 Relevant Synchronization Algorithms

In this section, the most famous clock synchronization algorithms in radio WSNs based on the two-way message exchange are described.

Reference Broadcast Synchronization (RBS)

To the best of the author's knowledge, RBS [11, 12] was the first relevant algorithm proposed for clock synchronization in WSNs; it was presented in 2002 [12] by Elson *et al.* for synchronizing a set of Berkeley motes [33]. The problem of reducing the uncertainty in the message delivery time has been solved by using the RRS paradigm in a smart way: in order to eliminate the Send and the Access times, the authors let a set of nodes synchronize themselves by recording the arrival time (according to their local clocks) of a number of broadcast packets received from an elected reference node; the key assumption is that the absolute time of arrival of the packet is the same at all the nodes: in other words it is assumed that the packets arrive to all the reference node's neighbors at the same instant. After having received these packets, such nodes exchange their local readings. Each node is therefore able to compute clock offset and skew with respect to any other node (except the reference node) by solving a least square linear regression. The receive time is limited by allowing system clock reading directly inside of the network driver's interrupt handler. A multi-hop extension of RBS consists on exploiting nodes that are in the coverage range of more than one node: such nodes work as clock bridges between two or more broadcast regions, therefore they are able to make the time conversion from a broadcast domain to another.

This algorithm has the following pros:

- The reference broadcast packet doesn't need to be a dedicated packet, therefore it is possible to exploit any broadcast transmission.
- The algorithm is robust against node deaths and packet losses since the linear regression works even if some points are missing.
- Local time scales are built up, making the algorithm well suited to those applications that don't need a network global time scale (like, for example, a localization application).
- A large number of synchronizing broadcasts enable tighter synchronization because residuals errors tend to follow well-behaved distributions.

There are also some cons:

- For a single-hop network of N nodes there is a huge ($O(N^2)$) number of exchanged packets for synchronization.
- Low convergence speed because of the high number of message exchanges.

- Need for the election of the beacon sender.
- The reference node is left unsynchronized.

Timing-sync Protocol for Sensor Networks (TPSN)

Ganeriwal et al. in [14] propose an algorithm that works upon a different paradigm with respect to RBS: in fact, while the latter is based on synchronizing a set of receiver nodes, TPSN bases its fundamentals on the more conventional approach, SRS. In order to synchronize the network, a hierarchical structure is constructed: this is the reason for the presence of a "discovery phase" in which a root node is chosen by running a leader election algorithm in the network so that a tree is built up. The "synchronization phase" follows, in which every node synchronizes itself to its parent by a two-way message exchange mechanism. The implementation results on Berkeley Motes are quite encouraging since TPSN provides twice better synchronization performance with respect to RBS; moreover, the synchronization error doesn't blow up with the hop distance. However, a strong limitation in this protocol is that it doesn't provide any skew correction, therefore a periodic resynchronization phase of the entire network is needed. MAC layer time-stamping is also implemented in this algorithm in order to reduce the send and the access time.

TPSN has the following pros:

- It is robust to single node deaths and births, with effective special provisions.
- It is easily scalable, since the synchronization error doesn't blow up with hop distance.
- Performance is good also with respect to RBS.

There are also some cons:

- It is not suitable in the case nodes have some mobility since a hierarchical structure has to be built up and kept running.
- Need for the election of the root node.
- No clock skew estimation.

Flooding Time Synchronization Protocol (FTSP)

In FTSP [13], the first step is the election of a root which is intended to represent the global reference time; it periodically broadcasts its clock to its neighbors which collect this data and build up a table with a set of entries *global time-local time*, using linear regression to estimate clock skew and offset with respect to the root node. After they get synchronized to the root, they themselves become roots for their neighborhood by broadcasting their synchronized time. Differently from RBS, the broadcast packet must contain the sender timestamp, but similar to TPSN the paradigm is SRS. Also here, MAC layer time-stamping is used in order to reduce the send and the access time.

The pros of this algorithm are the following:

- It supports topology changes since no fixed hierarchical structure is required.
- Less number of message exchanges with respect to RBS and TPSN.

On the other hand, the main cons are:

- A root election and renewing process is needed.
- The actual number of exchanged messages is redundant, due to coverage overlapping.

Pairwise Broadcast Synchronization (PBS)

The authors of PBS [15], are the inventors of a new synchronization approach, the ROS. As explained in Chapter 1, a node L exploits data embedded in messages exchanged by node S and R, synchronizing through the SRS paradigm, to estimate its clock skew and offset with respect to the clock of node S. Results of PBS in terms of synchronization error and in terms of the number of exchanged packets are encouraging since the clock precision is comparable to that of RRS approach while the number of exchanged messages is proportional to the number of clusters in the network. Here some advantages of this algorithm are listed:

- Low number of exchanged messages.
- Fine results in term of synchronization error.

However, the main disadvantages are:

- PBS needs to work over another SRS synchronization protocol.

- There is the need to divide the network in clusters and synchronize them by additional pairwise synchronization among super nodes in different clusters.

Several clock offset and skew estimators have been proposed [39–42] for the listening nodes, assuming this framework. On the other hand, a thorough analysis of this problem is out of the scope of this thesis.

Chapter 3

Clock Offset Estimation: A Unified Framework

The results presented in this chapter refer to a collaboration of Prof. Vangelista and me with Prof. Erchin Serpedin and Aitzaz Ahmad, both of them with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX (USA). The research work has been performed while I was on leave as a "Visiting Scholar" at Texas A&M University, from January to July 2011, partially supported by an "A. Gini" fellowship. Results have been submitted to the journal *IEEE Transactions on Information Theory* [48].

3.1 Introduction

The concepts introduced in this chapter are based on the two-way message exchange introduced in Chapter 2. Here, a new framework is defined for treating several assumptions on the random part of the link delays in a unified manner. In the considered scenario, only clock offset is assumed, so that the model depicted in Chapter 2 is substantially simplified. This allows to make a key quantity substitution in the observation model which leads to parameters decoupling.

Tools borrowed from the exponential family of distributions prove useful to tackle the problem of finding the MLE of the clock offset under the general framework, both recovering existing results and extending them to new findings using the convex optimization theory.

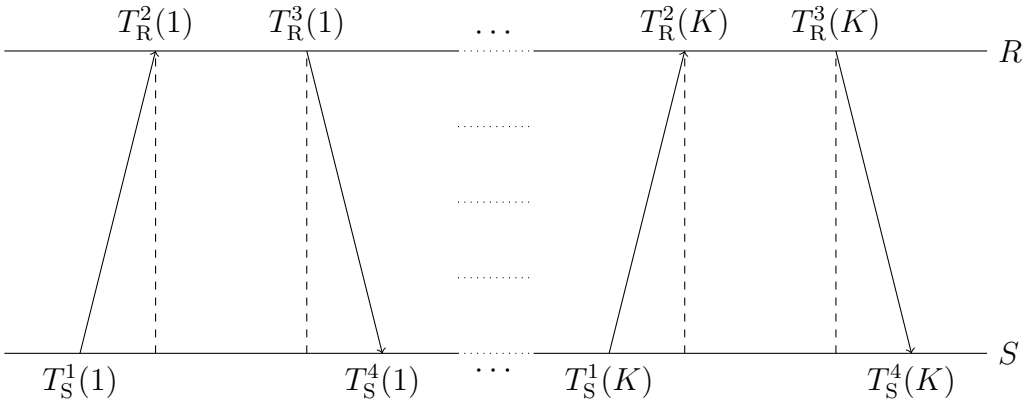

Figure 3.1: K rounds two-way message exchange between node S and R with clock offset only between them.

3.2 System Model

The data exchange mechanism on which the new common framework bases its fundamentals is the two-way message exchange, as described in Chapter 2. As an extension of what introduced therein, here it is assumed that the data exchange is iterated for K rounds, with the aim of providing a more accurate estimate of the clock parameters. In order to keep into account of the exchange repetition, the notation is extended as follows. The four timestamps $\{T_{\mathrm{S}}^1, T_{\mathrm{R}}^2, T_{\mathrm{R}}^3, T_{\mathrm{S}}^4\}$ gathered at round k are here denoted as

$$\{T_{\mathrm{S}}^1(k), T_{\mathrm{R}}^2(k), T_{\mathrm{R}}^3(k), T_{\mathrm{S}}^4(k)\} \ , \quad k = 1, \ldots, K \ .$$

In other words, brackets are added to remark the dependence of the timestamps on the round index k.

No clock skew is taken into account: Fig. 3.1 represents the considered scenario: a K rounds two-way message exchange mechanism with offset presence only. This assumption leads to a simplified relation between timestamps with respect to (2.1). In fact, setting $\theta^{(\mathrm{s})} = 0$ leads to

$$\begin{aligned} T_{\mathrm{R}}^2(k) &= T_{\mathrm{S}}^1(k) + d + X(k) + \theta^{(\mathrm{o})} \\ T_{\mathrm{R}}^3(k) &= T_{\mathrm{S}}^4(k) - d - Y(k) + \theta^{(\mathrm{o})} \ , \end{aligned} \tag{3.1}$$

where, as before, d represents the fixed (among all K rounds) portion of the delay while $X(k)$ and $Y(k)$ model the i.i.d. random portions of the link delay in uplink and downlink at round k, respectively.

Following the approach of [23, 25, 28–30], the readings are then processed to obtain

$$U(k) = T_{\mathrm{R}}^{2}(k) - T_{\mathrm{S}}^{1}(k) = d + \theta^{(\mathrm{o})} + X(k) \tag{3.2a}$$
$$V(k) = T_{\mathrm{S}}^{4}(k) - T_{\mathrm{R}}^{3}(k) = d - \theta^{(\mathrm{o})} + Y(k)\ . \tag{3.2b}$$

At this point, it results possible to define new quantities such that the two relations in (3.2) results to be uncoupled. In fact, by defining

$$\xi = d + \theta^{(\mathrm{o})} \tag{3.3a}$$
$$\psi = d - \theta^{(\mathrm{o})} \tag{3.3b}$$

the model (3.2) can be rewritten as

$$U(k) = \xi + X(k)$$
$$V(k) = \psi + Y(k)$$

for $k = 1, \ldots, K$.

3.3 A New General Framework

As pointed out in Chapter 2, the random portions of the delays $X(k)$ and $Y(k)$ can obey to a number of possible probability distribution functions. The aim of this chapter is to propose a new *general* framework for these delays so that three important distributions can be considered at the same time in a general fashion.

In order to make the notation shorter, the readings $U(k)$ and $V(k)$ are organized in vector form, i.e.,

$$\boldsymbol{U} = [U(1), \ldots, U(K)]^{'}$$
$$\boldsymbol{V} = [V(1), \ldots, V(K)]^{'}\ ,$$

so that it comes easier to express the likelihood functions of the observations. Here it is assumed that the likelihoods follow either the Gaussian or log-normal distribution, in a general framework. This can be done by borrowing notation and properties from the theory of the *exponential family of distributions* [49, 50].

Unconstrained Likelihood:

$$f_U(\boldsymbol{U}|\xi) \propto \exp\left(\xi \sum_{k=1}^{K} \eta_\xi(U(k)) - K\phi_\xi(\xi)\right) \tag{3.4a}$$

$$f_V(\boldsymbol{V}|\psi) \propto \exp\left(\psi \sum_{k=1}^{K} \eta_\psi(V(k)) - K\phi_\psi(\psi)\right) \tag{3.4b}$$

where $\eta_\xi(\cdot)$ and $\eta_\psi(\cdot)$ are functions of the observations $\boldsymbol{U}$ and $\boldsymbol{V}$, respectively and they are assumed to be sufficient statistics for estimating ξ and ψ. The $\phi_\xi(\cdot)$ and $\phi_\psi(\cdot)$ are called *log-partition functions* and they serve as normalization factors so that both $f_U(\boldsymbol{U}|\xi)$ and $f_V(\boldsymbol{V}|\psi)$ are valid probability distribution functions. Given the normalization purposes of the log-partition functions, they are naturally defined except for an additive constant. The proportional signs in (3.4a) and (3.4b) refer to the fact that a multiplicative factor only dependent on the observations $\boldsymbol{U}$ and $\boldsymbol{V}$, respectively, is present in front of the exponential term. In this case, the likelihood functions are called "unconstrained" since the domains are independent on the parameters ξ and ψ, as opposite to the case when the likelihood functions are assumed adhering to the exponential distribution, as shown below.

Constrained Likelihood:

$$f_U(\boldsymbol{U}|\xi) \propto \exp\left(\xi \sum_{k=1}^{K} \eta_\xi(U(k)) - K\phi_\xi(\xi)\right) \prod_{k=1}^{K} \mathbb{I}(U(k) - \xi) \tag{3.5a}$$

$$f_V(\boldsymbol{V}|\psi) \propto \exp\left(\psi \sum_{k=1}^{K} \eta_\psi(V(k)) - K\phi_\psi(\psi)\right) \prod_{k=1}^{K} \mathbb{I}(V(k) - \psi) \tag{3.5b}$$

where $\mathbb{I}(\cdot)$ is the indicator function, defined as

$$\mathbb{I}(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} .$$

In this case, the likelihood function is called "constrained" since its domain is dependent on the parameters ξ and ψ.

The exponential family of distributions represents a precious tool to analyse the likelihood functions in a sufficiently general fashion to encompass together potentially very different distributions like Gaussian, log-normal and exponential. The three of them are all valid candidates for best describing the random behaviour of $X(k)$ and $Y(k)$, thus it follows that the choice of adopting the exponential family framework is an effective way to study the

problem of clock offset estimation in a more general way with respect to what done in the past.

Although original, the approach of adopting uses and costumes of the exponential family of distribution can be specialized at the specific need, recovering the results of the literature when the assumption is to have random portions of delays obeying specifically either to Gaussian, log-normal or exponential distribution. This is of fundamental importance, since if on one side the framework permits a general analysis of the problem, simultaneously it leads to a continuity with the existing results.

It is useful to dig further into the theory of the exponential family of distributions: the following properties [49] will prove useful in the rest of the upcoming discussion. They are listed referring to the parameter ξ and the observations $\boldsymbol{U}$: similar properties hold for ψ and $\boldsymbol{V}$, respectively.

1. The mean value and the variance of the sufficient statistics $\eta_\xi(U(k))$ can be written as

$$\mathbb{E}\left[\eta_\xi(U(k))\right] = \frac{\partial \phi_\xi(\xi)}{\partial \xi} \tag{3.6}$$

$$\sigma^2_{\eta_\xi} = \mathrm{Var}\left[\eta_\xi(U(k))\right] = \frac{\partial^2 \phi_\xi(\xi)}{\partial \xi^2} \ . \tag{3.7}$$

2. The moment generating function (MGF) of the random variable $\eta_\xi(U(k))$ is given by

$$M_{\eta_\xi}(h) = \exp\left(\phi_\xi(\xi + h) - \phi_\xi(\xi)\right) \ . \tag{3.8}$$

The following proposition holds from the positiveness of the variance in (3.7).

Proposition 3.1 *The log-partition function $\phi_\xi(\cdot)$ is convex.*

The following theorem specializes the common framework to the three considered specific distributions.

Theorem 3.2 *The sufficient statistics and the log-partition function for the Gaussian, log-normal and exponential likelihood functions are respectively*

1. $\mathcal{N}\left(\xi, \sigma^2_\xi\right)$*:*

$$\eta_\xi\left(U(k)\right) = \frac{U(k)}{\sigma^2_\xi} \ , \qquad \phi_\xi(\xi) = \frac{\xi^2}{2\sigma^2_\xi} \ , \tag{3.9}$$

Distribution	$f_U(\boldsymbol{U}\|\xi)$	$\eta_\xi(U(k))$	$\phi_\xi(\xi)$	$\sigma^2_{\eta_\xi}$
$\mathcal{N}(\xi,\sigma^2_\xi)$	$\propto \exp\left[-\frac{\sum_{k=1}^K (U(k)-\xi)^2}{2\sigma^2_\xi}\right]$	$\frac{U(k)}{2\sigma^2_\xi}$	$\frac{\xi^2}{2\sigma^2_\xi}$	$\frac{1}{\sigma^2_\xi}$
$\log\mathcal{N}(\xi,\sigma^2_\xi)$	$\propto \exp\left[-\frac{\sum_{k=1}^K (\log U(k)-\xi)^2}{2\sigma^2_\xi}\right]$	$\frac{\log U(k)}{2\sigma^2_\xi}$	$\frac{\xi^2}{2\sigma^2_\xi}$	$\frac{1}{\sigma^2_\xi}$
$\mathscr{E}(\lambda_\xi)$	$\lambda_\xi \exp\left[-\lambda_\xi \sum_{k=1}^K (U(k)-\xi)\right] \cdot$ $\cdot \prod_{k=1}^K \mathbb{I}(U(k)-\xi)$	λ_ξ	0	0

Table 3.1: Specialization of the proposed new general framework.

2. $\log\mathcal{N}\left(\xi,\sigma^2_\xi\right)$*:*

$$\eta_\xi(U(k)) = \frac{\log U(k)}{\sigma^2_\xi}\,, \qquad \phi_\xi(\xi) = \frac{\xi^2}{2\sigma^2_\xi}\,, \tag{3.10}$$

3. $\mathscr{E}(\lambda_\xi)$*:*

$$\eta_\xi(U(k)) = \lambda_\xi\,, \qquad \phi_\xi(\xi) = 0\,. \tag{3.11}$$

Proof: See Appendix 3.A.1. ∎

The results stated by Theorem 3.2 are summarized in Tab. 3.1.

The following corollary holds.

Corollary 3.3 *In the Gaussian, log-normal as well as in the exponential case the log-partition function can be expressed as a second-degree incomplete polynomial, i.e.,*

$$\phi_\xi(\xi) = a_\xi \xi^2 = \frac{\sigma^2_{\eta_\xi}}{2}\xi^2\,. \tag{3.12}$$

In a real scenario, if the variance of the sufficient statistics is not known, it can be substituted by its empirical version, since the weak law of large numbers assures that

$$\frac{\sum_{k=1}^K \eta_\xi(U(k))}{K} \xrightarrow{p} \mathbb{E}\left[\eta_\xi(U)\right], \quad K\to\infty$$
$$\frac{\sum_{k=1}^K \eta^2_\xi(U(k))}{K} \xrightarrow{p} \mathbb{E}\left[\eta^2_\xi(U)\right], \quad K\to\infty\,.$$

3.4 Maximum Likelihood Estimation

This section focuses on the ML estimation of the clock offset, thus recovering the results existing in the literature. In order to perform the estimation, the problem is recast as an instance of convex optimization, differently from the graphical approach adopted in [24]. The case of unconstrained and constrained likelihoods are treated separately.

3.4.1 Unconstrained Likelihood

Using (3.4a) and (3.12), the unconstrained likelihood functions are given by

$$f_U(\boldsymbol{U}|\xi) \propto \exp\left(\xi \sum_{k=1}^{K} \eta_\xi(U(k)) - K\frac{\sigma_{\eta_\xi}^2}{2}\xi^2\right) \tag{3.13a}$$

$$f_V(\boldsymbol{V}|\psi) \propto \exp\left(\psi \sum_{k=1}^{K} \eta_\psi(V(k)) - K\frac{\sigma_{\eta_\psi}^2}{2}\psi^2\right) . \tag{3.13b}$$

The MLE of ξ and ψ can be expressed as

$$\hat{\xi}_{\mathrm{ML}} = \arg \max_{\xi} \exp\left(\xi \sum_{k=1}^{K} \eta_\xi(U(k)) - K\frac{\sigma_{\eta_\xi}^2}{2}\xi^2\right) \tag{3.14a}$$

$$\hat{\psi}_{\mathrm{ML}} = \arg \max_{\psi} \exp\left(\psi \sum_{k=1}^{K} \eta_\psi(V(k)) - K\frac{\sigma_{\eta_\psi}^2}{2}\psi^2\right) . \tag{3.14b}$$

Theorem 3.4 *The likelihood maximization problems* (3.14a) *and* (3.14b) *are strictly concave and the MLEs are given by*

$$\hat{\xi}_{ML} = \frac{\sum_{k=1}^{K} \eta_\xi(U(k))}{K\sigma_{\eta_\xi}^2} \tag{3.15a}$$

$$\hat{\psi}_{ML} = \frac{\sum_{k=1}^{K} \eta_\psi(V(k))}{K\sigma_{\eta_\psi}^2} . \tag{3.15b}$$

Hence, the MLE $\hat{\theta}_{ML}^{(\mathrm{o})}$ for the clock offset can be written as

$$\hat{\theta}_{ML}^{(\mathrm{o})} = \frac{\hat{\xi}_{ML} - \hat{\psi}_{ML}}{2} . \tag{3.16}$$

Proof: See Appendix 3.A.2 ∎

Gaussian Distribution

Theorem 3.4 finds an application in the case the likelihood functions f_U and f_V follow a Gaussian distribution, i.e., $f_U(U(k)|\xi) \sim \mathcal{N}(\xi, \sigma_\xi^2)$ and $f_V(V(k)|\psi) \sim \mathcal{N}(\psi, \sigma_\psi^2)$ [26] (see also Tab. 3.1). Using (3.15a) with (3.9) and (3.12), it follows that the MLE is given by

$$\hat{\xi}_{\mathrm{ML}} = \frac{\sum_{k=1}^{K} U(k)}{K} \,. \tag{3.17}$$

By a similar reasoning, the MLE for ψ (3.15b) is given by

$$\hat{\psi}_{\mathrm{ML}} = \frac{\sum_{k=1}^{K} V(k)}{K} \,. \tag{3.18}$$

Using (3.16), the MLE for the offset $\theta^{(\mathrm{o})}$ can be expressed as

$$\hat{\theta}^{(\mathrm{o})}_{\mathrm{ML}} = \frac{\sum_{k=1}^{K} (U(k) - V(k))}{2N} \,. \tag{3.19}$$

The above estimate coincides exactly with the one reported in [26].

Log-Normal Distribution

When the likelihood functions of the readings $\boldsymbol{U}$ and $\boldsymbol{V}$ are log-normal (see Tab. 3.1), Theorem 3.4 can still be applied, making use also of (3.10) and (3.12), for finding the MLE of ξ, i.e.,

$$\hat{\xi}_{\mathrm{ML}} = \frac{\sum_{k=1}^{K} \log U(k)}{K} \,.$$

With a similar reasoning, the MLE for ψ results to be

$$\hat{\psi}_{\mathrm{ML}} = \frac{\sum_{k=1}^{K} \log V(k)}{K} \,.$$

Finally, the ML estimator for $\theta^{(\mathrm{o})}$ can be written as

$$\hat{\theta}^{(\mathrm{o})}_{\mathrm{ML}} = \frac{\sum_{k=1}^{K} (\log U(k) - \log V(k))}{K} \,. \tag{3.20}$$

3.4.2 Constrained Likelihood

Using (3.5a), (3.5b) and (3.12), the constrained likelihood functions can be written as

$$f_U(\boldsymbol{U}|\xi) \propto \exp\left(\xi \sum_{k=1}^{K} \eta_\xi(U(k)) - K\frac{\sigma_{\eta_\xi}^2}{2}\xi^2\right) \prod_{k=1}^{K} \mathbb{I}(U(k)-\xi) \tag{3.21a}$$

$$f_V(\boldsymbol{V}|\psi) \propto \exp\left(\psi \sum_{k=1}^{K} \eta_\psi(V(k)) - K\frac{\sigma_{\eta_\psi}^2}{2}\psi^2\right) \prod_{k=1}^{K} \mathbb{I}(V(k)-\psi)\ . \tag{3.21b}$$

The MLEs of ξ and ψ can be expressed as

$$\hat{\xi}_{\mathrm{ML}} = \arg\ \max_{\xi} \exp\left(\xi \sum_{k=1}^{K} \eta_\xi(U(k)) - K\frac{\sigma_{\eta_\xi}^2}{2}\xi^2\right)$$
$$\text{such that} \quad U(k) \geq \xi \tag{3.22a}$$

$$\hat{\psi}_{\mathrm{ML}} = \arg\ \max_{\psi} \exp\left(\psi \sum_{k=1}^{K} \eta_\psi(V(k)) - K\frac{\sigma_{\eta_\psi}^2}{2}\psi^2\right)$$
$$\text{such that} \quad V(k) \geq \psi\ . \tag{3.22b}$$

It proves useful to define the order statistics of the samples $U(k)$ and $V(k)$ as

$$U_{(1)} = \min\left(U(1), \ldots, U(K)\right)$$
$$V_{(1)} = \min\left(V(1), \ldots, V(K)\right)\ .$$

Theorem 3.5 *The likelihood maximization problems* (3.22a) *and* (3.22b) *are strictly concave and the MLEs can be expressed as*

$$\hat{\xi}_{ML} = \min\left(\frac{\sum_{k=1}^{K} \eta_\xi(U(k))}{K\sigma_{\eta_\xi}^2}, U_{(1)}\right) \tag{3.23a}$$

$$\hat{\psi}_{ML} = \min\left(\frac{\sum_{k=1}^{K} \eta_\psi(V(k))}{K\sigma_{\eta_\psi}^2}, V_{(1)}\right)\ . \tag{3.23b}$$

The MLE $\hat{\theta}_{ML}^{(\mathrm{o})}$ *for the clock offset is given by*

$$\hat{\theta}_{ML}^{(\mathrm{o})} = \frac{\hat{\xi}_{ML} - \hat{\psi}_{ML}}{2}\ . \tag{3.24}$$

Proof: See Appendix 3.A.3. ∎

Exponential Distribution

An application of Theorem 3.5 can be found if the likelihood functions f_U and f_V represent exponential distributions, with parameters λ_ξ and λ_ψ, respectively, i.e., $f_U \sim \mathscr{E}(\lambda_\xi)$ and $f_V \sim \mathscr{E}(\lambda_\psi)$ (see also Tab. 3.1). Using (3.11) yields

$$\hat{\xi}_{\mathrm{ML}} = U_{(1)} \ . \tag{3.25}$$

With a similar reasoning,

$$\hat{\psi}_{\mathrm{ML}} = V_{(1)} \ .$$

Using (3.24), the MLE of $\theta^{(\mathrm{o})}$ is given by

$$\hat{\theta}_{\mathrm{ML}} = \frac{U_{(1)} - V_{(1)}}{2} \ , \tag{3.26}$$

thus recovering the result of [24], therein derived through graphical arguments.

3.5 Statistical Bounds

The purpose of this section is to provide statistical bounds to clock offset estimation in the classical framework in order to evaluate the performance of the MLEs derived in the previous sections in terms of the variance of the estimation error.

The upcoming analysis assumes that the likelihood functions f_U and f_V belongs to the exponential family of distributions without specifying any shape for the log-partition function, therefore the provided analytical results are indeed generally valid for a wide variety of distributions and almost all the distribution of interest. Moreover, the upcoming results can be used in a context different from the clock offset estimation, since they are self standing in classical estimation theory.

Both the case of unconstrained and constrained likelihood are taken into account for providing the necessary generality to the discussion of the problem. The general expressions for the likelihoods of the observations $\mathbf{Z} = [Z(1), \ldots, Z(K)]^T$ are given by

Unconstrained Likelihood:

$$f_Z(\mathbf{Z}; \rho) \propto \exp\left(\rho \sum_{k=1}^{K} \eta(Z(k)) - K\phi(\rho)\right) \tag{3.27}$$

Constrained Likelihood:

$$f_Z(\mathbf{Z};\rho) \propto \exp\left(\rho\sum_{k=1}^{K}\eta(Z(k)) - K\phi(\rho)\right)\prod_{k=1}^{K}\mathbb{I}(Z(k)-\rho) \tag{3.28}$$

where ρ is the scalar parameter to be estimated. In the following, the Cramer-Rao and Chapman-Robbins bounds for the estimation error variance of the deterministic parameter ρ are derived.

3.5.1 Cramer-Rao Lower Bound

The Cramer-Rao lower bound (CRB) [47] provides a limit on the variance of the estimation error for an unbiased estimator of a deterministic parameter. Although it is a simple bound to compute, it relies on some "regularity conditions" which are not always satisfied. In particular, in the case of constrained likelihoods (cf. (3.28)) the CRB cannot be computed, so that the variance of the estimation error has to be described by a different bound.

The CRB states that, given the condition

$$\mathbb{E}\left[\frac{\partial \log f_Z(\mathbf{Z};\rho)}{\partial\rho}\right] = 0 \quad \text{for all } \rho\,, \tag{3.29}$$

the variance of an unbiased estimator $\hat{\rho}$ of ρ is lower bounded by

$$\mathrm{Var}(\hat{\rho}) \geq \frac{-1}{\mathbb{E}\left[\frac{\partial^2 \log f_Z(\mathbf{Z};\rho)}{\partial\rho^2}\right]}\,, \tag{3.30}$$

Theorem 3.6 *The CRB for the estimation error of ρ in the unconstrained likelihood function in* (3.27) *is given by*

$$\mathrm{Var}(\hat{\rho}) \geq \frac{1}{K\sigma_\eta^2} \tag{3.31}$$

where

$$\sigma_\eta^2 = \frac{\partial^2\phi(\rho)}{\partial\rho^2}\,.$$

Proof: The Fisher information is given by

$$\begin{aligned} I(\rho) &= \mathbb{E}\left[\frac{\partial^2 \log f_Z(\mathbf{Z};\rho)}{\partial\rho^2}\right] \\ &= -K\frac{\partial^2\phi(\rho)}{\partial\rho^2} = -K\sigma_\eta^2 \end{aligned}$$

so that the proof readily follows. ∎

3.5.2 Chapman-Robbins Bound

The Chapman-Robbins bound (CHRB) [51] provides a lower bound on the variance of an estimator of a deterministic parameter. Differently from the CRB, the CHRB does not make any assumptions on the regularity condition (3.29) that often prevents from the use of the CRB. Moreover, the CHRB is substantially tighter than the CRB in many cases. Hence, CHRB is here used for determining a lower bound on the variance of an unbiased estimator of ρ for constrained likelihood functions.

Generally speaking, the CHRB for estimating a parameter ρ is given by

$$\mathrm{Var}(\hat{\rho}) \geq \left[\inf_{h} \frac{1}{h^2} \left\{ \mathbb{E}\left(\frac{f_Z(\mathbf{Z}; \rho + h)}{f_Z(\mathbf{Z}; \rho)} \right)^2 - 1 \right\} \right]^{-1} . \tag{3.32}$$

Theorem 3.7 *The CHRB for the parameter ρ given the likelihood function* (3.28) *can be expressed as*

$$\mathrm{Var}(\hat{\rho}) \geq \left[\inf_{h} \frac{\left\{ \left(M_\eta(h)\right)^{-2K} \cdot \delta^K(h) - 1 \right\}}{h^2} \right]^{-1} \tag{3.33}$$

where $M_\eta(h)$ is the MGF of the statistic $\eta(Z(k))$ and

$$\delta(h) = \mathbb{E}\left[\exp\left(2h\eta(Z(k))\right) \mathbb{I}\left(Z(k) - \rho - h\right) \right] \tag{3.34}$$

with the expectation taken with respect to any $Z(k)$.

Proof: The details of the proof are relegated to Appendix 3.A.4. ■

3.5.3 MSE and Variance of the Estimation Error

The performance bounds computed above provide lower limits to the variance of the estimation error of ξ and ψ. Such results can be used to lower bound the mean squared error (MSE) of the clock offset $\theta^{(\mathrm{o})}$,

$$\mathrm{MSE}\left(\hat{\theta}^{(\mathrm{o})}\right) = \mathbb{E}\left[\left(\hat{\theta}^{(\mathrm{o})} - \theta^{(\mathrm{o})} \right)^2 \right] .$$

In fact, by using (3.3), the following result is immediate.

Proposition 3.8 *The MSE of any estimator of $\theta^{(\mathrm{o})}$ can be expressed as*

$$\mathrm{MSE}\left(\hat{\theta}^{(\mathrm{o})}\right) = \frac{1}{4}\left(Var\left(\hat{\xi}\right) + \mathrm{Var}\left(\hat{\psi}\right) \right) + \frac{1}{4}\left(b_\xi - b_\psi\right)^2$$

where b_ξ and b_ψ [47] are the biases of the estimators $\hat{\xi}$ and $\hat{\psi}$, respectively.

Gaussian Distribution - CRB

If the likelihood function for ξ is Gaussian distributed (see Tab. 3.1), then using (3.31), it results that the CRB for an unbiased estimator $\hat{\xi}$ is given by

$$\mathrm{Var}\left(\hat{\xi}\right) \geq \frac{\sigma_\xi^2}{K} ,$$

with a similar expression for the CRB in estimating ψ as well. Using Proposition 3.8, it follows that

$$\mathrm{MSE}\left(\hat{\theta}^{(\mathrm{o})}\right) \geq \frac{\sigma_\xi^2 + \sigma_\psi^2}{4K} . \tag{3.35}$$

As a remark, $\hat{\theta}_{\mathrm{ML}}^{(\mathrm{o})}$ (3.19) is an efficient estimator since its MSE achieves (3.35) with equality (cf. Appendix 3.A.5).

Exponential Distribution - CHRB

If the likelihood for ξ is exponentially distributed (see Tab. 3.1), using (3.8) it can be easily shown that

$$M_{\eta_\xi(U)}(h) = 1$$

and (3.34) becomes

$$\delta(h) = \exp\left(\lambda_\xi h\right) ,$$

so that the CHRB (3.33) can be rewritten as

$$\mathrm{Var}\left(\hat{\xi}\right) \geq \left[\inf_h \frac{\exp\left(\lambda_\xi h K\right) - 1}{h^2}\right]^{-1} = \frac{0.6476}{\lambda_\xi^2 K^2}$$

and similarly for $\hat{\psi}$. Using Proposition 3.8, it holds that

$$\begin{aligned} \mathrm{MSE}\left(\hat{\theta}^{(\mathrm{o})}\right) &= \frac{1}{4}\left(\mathrm{Var}\left(\hat{\xi}\right) + \mathrm{Var}\left(\hat{\psi}\right)\right) + \frac{1}{4}\left(b_\xi - b_\psi\right)^2 \\ &\geq \frac{0.162}{K^2}\left(\frac{1}{\lambda_\xi^2} + \frac{1}{\lambda_\psi^2}\right) + \frac{1}{4}\left(b_\xi - b_\psi\right)^2 . \end{aligned} \tag{3.36}$$

3.6 Numerical Results

In this section the theoretical findings of this chapter are corroborated by numerical simulations. In particular, performance of the estimator in case of

log-normal likelihood is analysed, as well as the MSE in all cases is compared with the statistical bounds computed in Section 3.5. The setting parameters are given by $\sigma_\xi = \sigma_\psi = 0.1$ for both Gaussian and log-normally distributed likelihoods, while $\lambda_\xi = \lambda_\psi = 10$ for exponentially distributed likelihood functions.

3.6.1 MSE Performance and Bounds

Comparing the performance of the proposed estimator with the statistical bounds is useful to provide a measure of correctness and efficiency of MLEs derived in Section 3.4 in the case of Gaussian and exponentially distributed likelihoods. Fig. 3.2 reports the MSE in estimating the clock offset using the MLEs for Gaussian and exponential likelihoods, (3.19) and (3.26), respectively (the log-normal case is analogous to the Gaussian case, therefore it is not reported here). As benchmarks, the CRB and CHRB are also shown in the plot. First, it is evident that the MLE $\hat{\theta}^{(\mathrm{o})}_{\mathrm{ML}}$ in the case of Gaussian likelihood is efficient since the MSE reaches the CRB and the CHRB, the two being coincident. On the other hand, in the exponential case the CRB cannot be computed because the regularity assumption (3.29) is not satisfied, therefore only the CHRB is shown in figure. The MSE of the MLEs in this case is close, but not coincident, to the CHRB. A further fair comparison which can be made is among the Gaussian and the exponential case, since the parameter choice is such that the variance of the observations is the same: from the shown curves it is evident that the MSE in the exponential case is lower than that in the Gaussian case as the number of observations K increases. This behaviour can be verified by the MSE expressions (3.37) and (3.38), reported in Appendix 3.A.5: in the Gaussian case, the MSE decreases proportionally to $1/K$, while in the exponential distribution case it decreases proportional to $1/K^2$.

3.6.2 MSE Robustness with Log-Normally Distributed Likelihood

So far, no clock offset estimator has been proposed in the literature in case the likelihood function of the readings is log-normal. The proposed estimator $\hat{\theta}^{(\mathrm{o})}_{\mathrm{ML}}$ (3.20) can then be used to determine an estimate of the clock offset in the maximum likelihood sense. Until now, if the likelihood of the readings was log-normal, the available estimators were just those assuming the likelihoods to be either Gaussian or exponentially distributed, therefore they are expected to perform worse than the estimator (3.20) matched to the case of

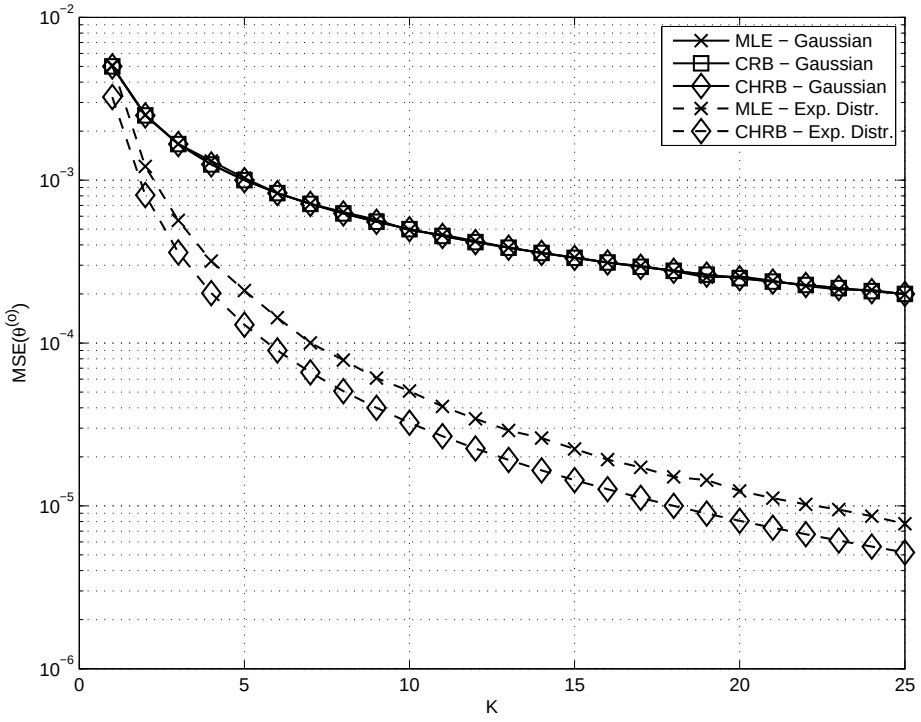

Figure 3.2: MSE and bounds for estimating $\theta^{(\mathrm{o})}$ by using the MLE with Gaussian and exponentially distributed likelihood.

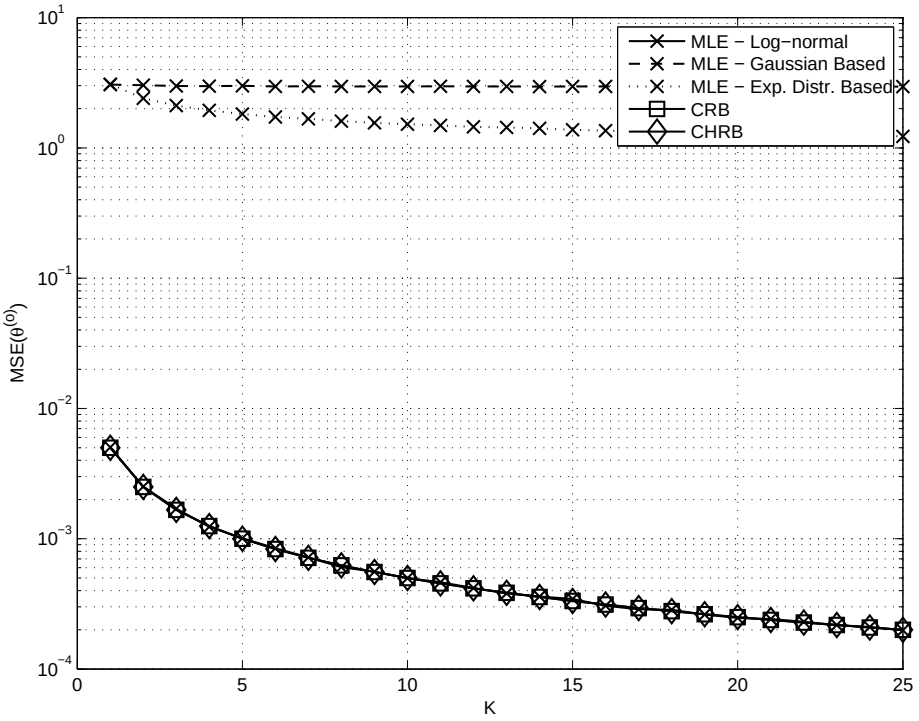

Figure 3.3: MSE robustness (in terms of the likelihood functions) and bounds for the estimation of $\theta^{(o)}$ in case of log-normally distributed likelihood.

log-normal likelihood.

Fig. 3.3 compares the MSE of the new proposed estimator (3.20) with that of the MLEs that (wrongly) assume the likelihood to be either Gaussian or exponentially distributed, when the actual likelihood of the readings is log-normally distributed. The figure shows how the two latter approaches are not robust with respect to the likelihood distribution, since their performance is extremely poor. Moreover, it can be observed how the proposed MLE (3.20) is efficient since it achieves the CRB (as well as the CHRB).

3.7 Conclusions

The new proposed framework allows to uniformly treat different likelihood functions, by exploiting the common properties among them, and using notation and fundamental properties of the exponential family of distributions. This generalization process allows to tackle the ML estimation problem in a more general fashion, which can be specialized at the specific need depending on the behaviour of the likelihood functions. This particular approach is

naturally based on convex optimization, therefore in the exponential distribution case it represents an alternative view to the problem of clock offset ML estimation with respect to what has been done up to now, under this specific assumption [24]. Statistical bounds are provided for all the cases of interests, while numerical results confirm the performance of the derived MLEs.

Moreover, it is worth to highlight that the provided framework allows to derive the expression of $\theta_{\mathrm{ML}}^{(\mathrm{o})}$ in case the likelihood functions of the readings are log-normal. This result is new in the literature, although the log-normal distribution is among the candidates for modelling the distribution of the random portions of the delay. Simulation results show that the new MLE performs better than the state-of-the-art in clock offset estimation.

The ML estimation method outlined above differs from the previous work [24] in that it is based on convex optimization, while [24] maximized the likelihood graphically. Hence, this approach presents an alternative view of the ML estimation of the clock offset. It also allows us the determine the ML estimator of $\theta^{(\mathrm{o})}$ when the likelihood function is log-normally distributed. In addition, Theorem 3.5 also provides a stepping stone for the findings described in Chapter 4, where estimation of $\theta^{(\mathrm{o})}$ in the Bayesian regime is discussed.

3.A Appendix

3.A.1 Proof of Theorem 3.2

1. $\mathcal{N}\left(\xi, \sigma_\xi^2\right)$:

$$\begin{aligned} f_U(\boldsymbol{U}|\xi) &\propto \exp\left[-\frac{\sum_{k=1}^K \left(U(k)-\xi\right)^2}{2\sigma_\xi^2}\right] \\ &\propto \exp\left[-\frac{\sum_{k=1}^K \left(U^2(k)-2\xi U(k)+\xi^2\right)}{2\sigma_\xi^2}\right] \\ &\propto \exp\left[\xi\sum_{k=1}^K \frac{U(k)}{\sigma_\xi^2} - K\frac{\xi^2}{2\sigma_\xi^2}\right] , \end{aligned}$$

   so that (3.9) follows from comparison with (3.4a).

2. $\log\mathcal{N}\left(\xi, \sigma_\xi^2\right)$:

$$\begin{aligned} f_U(\boldsymbol{U}|\xi) &\propto \exp\left[-\frac{\sum_{k=1}^K \left(\log U(k)-\xi\right)^2}{2\sigma_\xi^2}\right] \\ &\propto \exp\left[-\frac{\sum_{k=1}^K \left(\log^2 U(k)-2\xi\log U(k)+\xi^2\right)}{2\sigma_\xi^2}\right] \\ &\propto \exp\left[\xi\sum_{k=1}^K \frac{\log U(k)}{\sigma_\xi^2} - K\frac{\xi^2}{2\sigma_\xi^2}\right] , \end{aligned}$$

   therefore (3.10) follows, again from comparison with (3.4a).

3. $\mathscr{E}\left(\lambda_\xi\right)$:

$$\begin{aligned} f_U(\boldsymbol{U}|\xi) &\propto \exp\left[-\lambda_\xi\sum_{k=1}^K \left(U(k)-\xi\right)\right]\prod_{k=1}^K \mathbb{I}\left(U(k)-\xi\right) \\ &\propto \exp\left[-\lambda_\xi\sum_{k=1}^K U(k)+\xi\sum_{k=1}^K \lambda_\xi\right]\prod_{k=1}^K \mathbb{I}\left(U(k)-\xi\right) \\ &\propto \exp\left[\xi\sum_{k=1}^K \lambda_\xi\right]\prod_{k=1}^K \mathbb{I}\left(U(k)-\xi\right) , \end{aligned}$$

   which leads to (3.11) through comparison with (3.5a).

3.A.2 Proof of Theorem 3.4

The MLE of ξ (3.15a) can be determined by maximizing the exponent of (3.14a), since it can be easily proved as strictly concave [52]. Setting to zero the first derivative of the exponent of (3.14a) leads to (3.15a). A similar reasoning for the MLE (3.14b) of ψ brings to (3.15b). Finally, the MLE of $\theta^{(\mathrm{o})}$ (3.16) is derived using (3.3) and the invariance principle [47].

3.A.3 Proof of Theorem 3.5

The strict concavity of the maximization problems (3.22a) and (3.22b) follows from noting that the constraints $U(k) \geq \xi$ (resp. $V(k) \geq \psi$) are linear functions of ξ (resp. ψ) and using similar arguments to those used in the proof of Theorem 3.4. Moreover, the K constraints can be reduced to a single constraint in each problem, by using the order statistics of the observations, i.e.,

$$\begin{aligned}
\hat{\xi}_{\mathrm{ML}} &= \arg \max_{\xi} \exp\left(\xi \sum_{k=1}^{K} \eta_{\xi}(U(k)) - K\frac{\sigma_{\eta_\xi}^2}{2}\xi^2\right) \\
&\quad \text{such that} \quad U_{(1)} \geq \xi \\
\hat{\psi}_{\mathrm{ML}} &= \arg \max_{\psi} \exp\left(\psi \sum_{k=1}^{K} \eta_{\psi}(V(k)) - K\frac{\sigma_{\eta_\psi}^2}{2}\psi^2\right) \\
&\quad \text{such that} \quad V_{(1)} \geq \psi \, .
\end{aligned}$$

The solution of the unconstrained problem in (3.22a), $\bar{\xi}$, is given by (3.15a). If $\bar{\xi} \leq U_{(1)}$, then the solution of (3.22a) is given by $\hat{\xi}_{\mathrm{ML}} = \bar{\xi}$. On the other hand, being the objective function concave, if $U_{(1)} \leq \bar{\xi}$, then the ML estimate is given by $\hat{\xi}_{\mathrm{ML}} = U_{(1)}$. The combination of the two cases leads to (3.23a). A similar reasoning for ψ yields (3.23b). Finally, the MLE of $\theta^{(\mathrm{o})}$ (3.24) is derived using (3.3) and the invariance principle [47].

3.A.4 Proof of Theorem 3.7

The ratio of the likelihood functions can be expressed as

$$\begin{aligned}
&\frac{f_Z(\mathbf{Z};\rho+h)}{f_Z(\mathbf{Z};\rho)}\\
=&\frac{e^{\left((\rho+h)\sum_{k=1}^{K}\eta(Z(k))-K\phi(\rho+h)\right)}\prod_{k=1}^{K}\mathbb{I}(Z(k)-\rho-h)}{e^{\left(\rho\sum_{k=1}^{K}\eta(Z(k))-K\phi(\rho)\right)}\prod_{k=1}^{K}\mathbb{I}(Z(k)-\rho)}\\
=&e^{\left(h\sum_{k=1}^{K}\eta(Z(k))-K\phi(\rho+h)+K\phi(\rho)\right)}\prod_{k=1}^{K}\mathbb{I}(Z(k)-\rho-h)\\
=&e^{\left(h\sum_{k=1}^{K}\eta(Z(k))\right)}e^{(-K(\phi(\rho+h)+\phi(\rho)))}\prod_{k=1}^{K}\mathbb{I}(Z(k)-\rho-h)\ .
\end{aligned}$$

The expectation of the ratio of the likelihood functions can now be calculated as

$$\begin{aligned}
&\mathbb{E}\left(\frac{f_Z(\mathbf{Z};\rho+h)}{f_Z(\mathbf{Z};\rho)}\right)^2\\
&=\mathbb{E}\left[e^{\left(2h\sum_{k=1}^{K}\eta(Z(k))\right)}e^{(-2K(\phi(\rho+h)+\phi(\rho)))}\prod_{k=1}^{K}\mathbb{I}(Z(k)-\rho-h)\right]\\
&=e^{(-2K(\phi(\rho+h)+\phi(\rho)))}\mathbb{E}\left[e^{\left(2h\sum_{k=1}^{K}\eta(Z(k))\right)}\prod_{k=1}^{K}\mathbb{I}(Z(k)-\rho-h)\right]\\
&=\left(M_{\eta(Z)}(h)\right)^{-2K}\mathbb{E}\left[e^{\left(2h\sum_{k=1}^{K}\eta(Z(k))\right)}\prod_{k=1}^{K}\mathbb{I}(Z(k)-\rho-h)\right]
\end{aligned}$$

where it follows from (3.8) that

$$\left(M_{\eta(Z)}(h)\right)^{-2K}=e^{-2K(\phi(\rho+h)-\phi(\rho))}\ .$$

Since the samples $Z(k)$ are *i.i.d*,

$$\begin{aligned}
&\mathbb{E}\left[e^{\left(2h\sum_{k=1}^{K}\eta(Z(k))\right)}\prod_{k=1}^{K}\mathbb{I}(Z(k)-\rho-h)\right]\\
&\quad=\left(\mathbb{E}\left[e^{(2h\eta(Z(k)))}\mathbb{I}(Z(k)-\rho-h)\right]\right)^K\ .
\end{aligned}$$

With $\delta(\cdot)$ defined in the theorem, the proof is complete.

3.A.5 MSE Expressions for ML Estimators

Gaussian Distribution

If the likelihood for ξ is Gaussian distributed (see Tab. 4.1), the MLE is given by (3.17). Since the variance of the readings U_j is σ_ξ^2 and the MLE (3.17) is unbiased, it is straightforward to see that

$$\mathrm{MSE}\left(\hat{\xi}_{\mathrm{ML}}\right) = \frac{\sigma_\xi^2}{K} ,$$

and similarly for $\hat{\psi}_{\mathrm{ML}}$. Given (3.16) and Proposition 3.8, it can be concluded that

$$\mathrm{MSE}\left(\hat{\theta}_{\mathrm{ML}}^{(\mathrm{o})}\right) = \frac{\sigma_\xi^2 + \sigma_\psi^2}{4K} . \tag{3.37}$$

Exponential Distribution

If the likelihood for ξ is exponential distributed (see Tab. 4.1), the MLE is given by (3.25). Through simple algebra it can be seen that $U_{(1)}$ is exponentially distributed with parameter $\lambda_\xi^{'} = \lambda_\xi K$, so that $\mathrm{Var}\left(\hat{\xi}_{\mathrm{ML}}\right) = \frac{1}{\lambda_\xi^2 K^2}$. It can be noticed that $\hat{\xi}_{\mathrm{ML}}$ is a biased estimator for ξ, with bias $b_{\xi,\mathrm{ML}} = \frac{1}{\lambda_\xi K}$. Similarly, $\mathrm{Var}\left(\hat{\psi}_{\mathrm{ML}}\right) = \frac{1}{\lambda_\psi^2 K^2}$ and $b_{\psi,\mathrm{ML}} = \frac{1}{\lambda_\psi K}$. Therefore, given (3.16) and Proposition 3.8, it can be concluded that

$$\mathrm{MSE}\left(\hat{\theta}_{\mathrm{ML}}^{(\mathrm{o})}\right) = \frac{0.25}{K^2}\left(\frac{1}{\lambda_\xi^2} + \frac{1}{\lambda_\psi^2}\right) + \frac{0.25}{K^2}\left(\frac{1}{\lambda_\xi} - \frac{1}{\lambda_\psi}\right)^2 . \tag{3.38}$$

Chapter 4

A Factor Graph-Based Clock Offset Estimator

The results presented in this chapter refer to a collaboration of Prof. Vangelista and me with Prof. Erchin Serpedin and Aitzaz Ahmad, both of them with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX (USA). The research work has been performed while I was on leave as a "Visiting Scholar" at Texas A&M University, from January to July 2011, partially supported by an "A. Gini" fellowship. Results have been summarized in two papers, one submitted to the journal *IEEE Transactions on Information Theory* [48] and the other accepted for publication at the conference *ICASSP 2012* [53].

4.1 Introduction

This chapter presents a new clock offset estimator based on a Bayesian version of the new framework introduced in Chapter 3. The estimation procedure assumes a time-varying clock offset and it makes use of the factor graph theory together with message passing for deriving a closed-form expression of the estimator. Moreover, in order to measure the performance of the estimator, Bayesian estimation bounds are derived for the MSE and existing estimators are implemented, then showing that the new proposed estimator performs better than the state-of-the-art in clock offset estimators and it provides performance really close to the computed lower bounds.

The assumptions made in Chapter 3 refer to the absence of clock skew between node pairs in a WSN and time-invariant clock offsets. Although this model permits to derive a common framework and closed-form expressions for clock offset estimators, it is generally not well suited to real world scenar-

ios, where climate conditions change over time, but they are not kept into consideration. In fact, with fixed clock offsets, only fabrication imperfections are modelled. Considering the fixed portion of the delay to be time-invariant is also quite restrictive because it does encompass neither slow mobility of the wireless nodes nor shadowing effects in urban environments.

To deal with offset evolution over time, here it is assumed that clock offset evolves obeying to a Gauss-Markov model [29, 30, 42]. The Bayesian estimation procedure is performed by the application of the factor graph theory with the message passing algorithm. The basics of factor graphs are explained in the next section.

4.2 Factor Graphs and Message Passing

Factor graphs are *bipartite* graphical models used to better represent the relations among variables in a global function. The global function is decomposed in a set of local sub-functions, called *factors*, each of them dependent on a subset of variables. Generally, each factor is represented by a factor node and each variable by an edge. The fundamental rule is that an edge is connected to a particular factor node if and only if the variable associated to that edge is an argument of the factor associated to that factor node [54].

Message passing algorithms are used to perform inference in factor graphs by exchanging messages (*beliefs*) among nodes in the graph. The *sum-product* and the *max-product* algorithm are examples of message passing algorithms in which summation and maximization are performed respectively at each factor node. Here, the focus is on the *max-product* algorithm, and the exchanged messages are as follows

variable to factor node :

$$m_{x\to f}(x) = \prod_{h\in n(x)\setminus f} m_{h\to x}(x) \tag{4.1}$$

factor node to variable :

$$m_{f\to x}(x) = \max_{\setminus\{x\}} \left(f(Z) \prod_{z\in n(f)\setminus\{x\}} m_{z\to f}(z) \right) \tag{4.2}$$

where $Z = n(f)$ is the subset of variables from which the local function f is dependent. The marginal distribution, or the belief, associated with a variable is obtained by performing the product of all the incoming messages to that variable node.

4.3 A Factor Graph-based Estimator

The time-varying behaviour of the clock offset and the fixed portion of the delay is kept into account by assuming a Gauss-Markov process for both the parameters ξ and ψ, i.e.,

$$\xi(k) = \xi(k-1) + w(k) \tag{4.3}$$
$$\psi(k) = \psi(k-1) + v(k) \quad \text{for } k = 1, \ldots, K \tag{4.4}$$

with $w(k)$ and $v(k)$ i.i.d and $w(k), v(k) \sim \mathcal{N}(0, \sigma^2)$. The Gauss-Markov model for the clock offset has already found application in previous works on clock synchronization [29, 30, 42].

Under the Bayesian assumption, the a-posteriori pdf can be expressed as

$$\begin{aligned} f_{\mathrm{a}}(\boldsymbol{\xi}, \boldsymbol{\psi} | \boldsymbol{U}, \boldsymbol{V}) &\propto f_{\mathrm{p}}(\boldsymbol{\xi}, \boldsymbol{\psi}) f_{\ell}(\boldsymbol{U}, \boldsymbol{V} | \boldsymbol{\xi}, \boldsymbol{\psi}) \\ &= f_{\mathrm{p},\xi}(\xi(0)) \prod_{k=1}^{K} f_{\xi}(\xi(k)|\xi(k-1)) f_{\mathrm{p},\psi}(\psi(0)) \prod_{k=1}^{K} f_{\psi}(\psi(k)|\psi(k-1)) \\ &\cdot \prod_{k=1}^{K} f_U(U(k)|\xi(k)) f_V(V(k)|\psi(k)) \end{aligned} \tag{4.5}$$

where the a-priori distributions $f_{\mathrm{p},\xi}(\xi(0))$ and $f_{\mathrm{p},\psi}(\psi(0))$ are assumed uniform, i.e., they can be considered constant. To make the notation more readable, the following definitions hold:

$$\begin{aligned} \zeta_{k-1}^{k} &= f_{\xi}(\xi(k)|\xi(k-1)) \sim \mathcal{N}(\xi(k-1), \sigma^2) \,, \\ \nu_{k-1}^{k} &= f_{\psi}(\psi(k)|\psi(k-1)) \sim \mathcal{N}(\psi(k-1), \sigma^2) \,, \\ \varphi_k &= f_U(U(k)|\xi(k)) \,, \\ \chi_k &= f_V(V(k)|\psi(k)) \,. \end{aligned}$$

In the following treatment, only the case of constrained likelihood is analysed, since the unconstrained likelihood is a subset of the constrained case, as it will be clarified shortly.

The common framework introduced in Chapter 3 is adapted to the Bayesian framework, therefore the likelihood functions f_U and f_V are assumed obeying to the exponential family of distributions (3.21), so that they can be

expressed as

$$f_U(U(k)|\xi(k)) \propto \exp\left(\xi(k)\eta_\xi(U(k)) - \frac{\sigma^2_{\eta_{\xi,k}}}{2}\xi^2(k)\right)\mathbb{I}(U(k)-\xi(k))$$
$$f_V(V(k)|\psi(k)) \propto \exp\left(\psi(k)\eta_\psi(V(k)) - \frac{\sigma^2_{\eta_{\psi,k}}}{2}\psi^2(k)\right)\mathbb{I}(V(k)-\psi(k))\ . \tag{4.6}$$

In Fig. 4.1, the a-posteriori pdf (4.5) is represented via factor graphs. As an example of factor graph representation of Gauss-Markov models, graphical phase estimation has been performed in [55]. It is fundamental to notice that the structure of the pdf naturally leads to a separation between terms dependant on $\xi(k)$ only and terms dependant on $\psi(k)$ only. This means that the resulting factor graph is actually composed by two sub-graphs, not communicating to each other. This is the consequence of the decoupling between the two parameters derived in Chapter 3 and still valid. Moreover, through inspection of (4.5) it can be inferred that the two sub-graphs have a similar structure, therefore in the following the message passing is described for ξ only, being the case for ψ completely equivalent. Finally, the sub-graphs in Fig. 4.1 are cycle-free, therefore inference through message passing in these graphs is optimal [54].

To make the notation easier, it is convenient to define the following quantities

$$\alpha_{\xi,k} = -\frac{\sigma^2_{\eta_{\xi,k}}}{2}\ , \qquad \beta_{\xi,k} = \eta_\xi(U(k))$$

so that the constrained likelihood function φ_k can be expressed as

$$\varphi_k \propto \exp\left(\alpha_{\xi,k}\xi^2(k) + \beta_{\xi,k}\xi(k)\right)\mathbb{I}(U(k)-\xi(k))\ . \tag{4.7}$$

The next step is to apply the message passing mechanism to derive an estimate for $\theta^{(\mathrm{o})}(K)$.

4.3.1 Message Computation

The application of the max-product algorithm leads to message passing between variable nodes and factor nodes. The procedure begins at the factor node φ_K, from which a message is sent to the variable $\xi(K)$ (see Fig. 4.2), i.e.,

$$m_{\varphi_K \to \xi(K)} = \varphi_K\ .$$

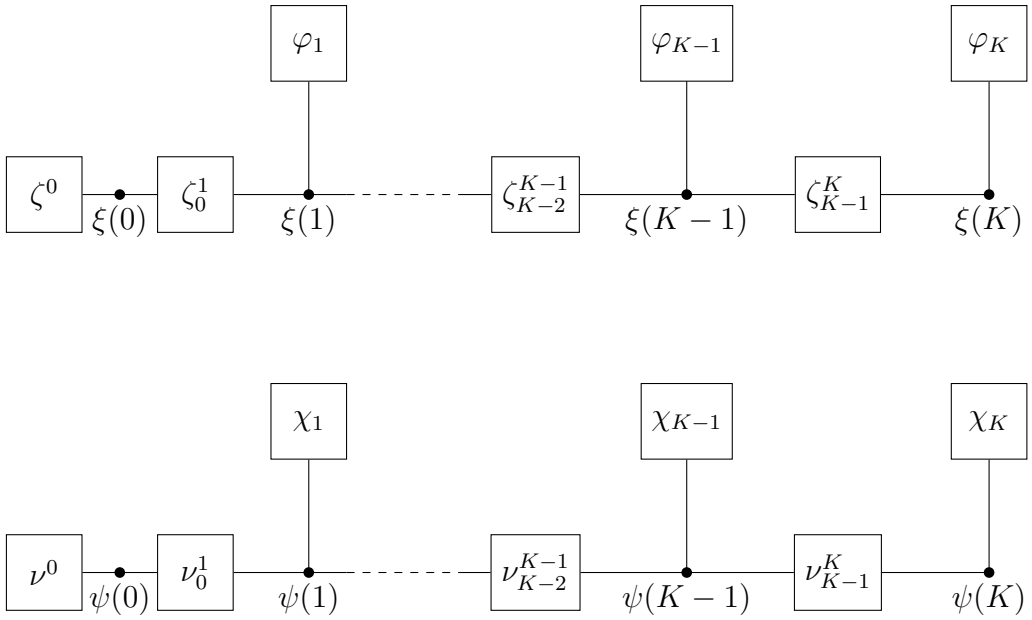

Figure 4.1: Factor graph representation of the density (4.5).

Distribution	$f_U(U(k)\|\xi)$	$\eta_\xi(U(k))$	$\phi_\xi(\xi(k))$	$\sigma^2_{\eta_\xi}$
$\mathcal{N}(\xi(k),\sigma_\xi^2)$	$\propto \exp\left[-\frac{(U(k)-\xi(k))^2}{2\sigma_\xi^2}\right]$	$\frac{U(k)}{2\sigma_\xi^2}$	$\frac{\xi^2(k)}{2\sigma_\xi^2}$	$\frac{1}{\sigma_\xi^2}$
$\log\mathcal{N}(\xi(k),\sigma_\xi^2)$	$\propto \exp\left[-\frac{(\log U(k)-\xi(k))^2}{2\sigma_\xi^2}\right]$	$\frac{\log U(k)}{2\sigma_\xi^2}$	$\frac{\xi^2(k)}{2\sigma_\xi^2}$	$\frac{1}{\sigma_\xi^2}$
$\mathscr{E}(\lambda_\xi)$	$\lambda_\xi \exp\left[-\lambda_\xi\left(U(k)-\xi(k)\right)\right]\cdot$ $\cdot\mathbb{I}\left(U(k)-\xi(k)\right)$	λ_ξ	0	0

Table 4.1: Specialization of the proposed new general framework for time-varying parameters.

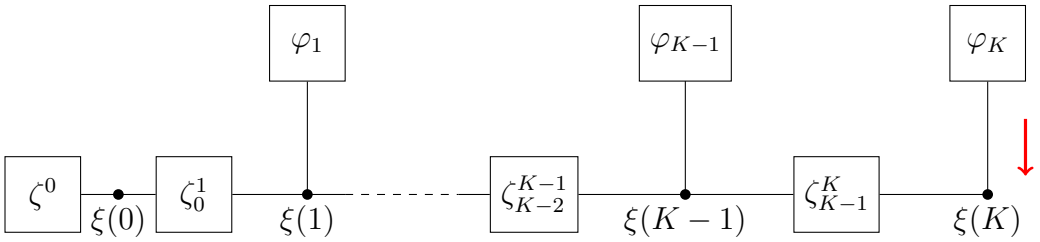

Figure 4.2: The factor φ_K sends the message $m_{\varphi_K \to \xi(K)}$ to the variable $\xi(K)$.

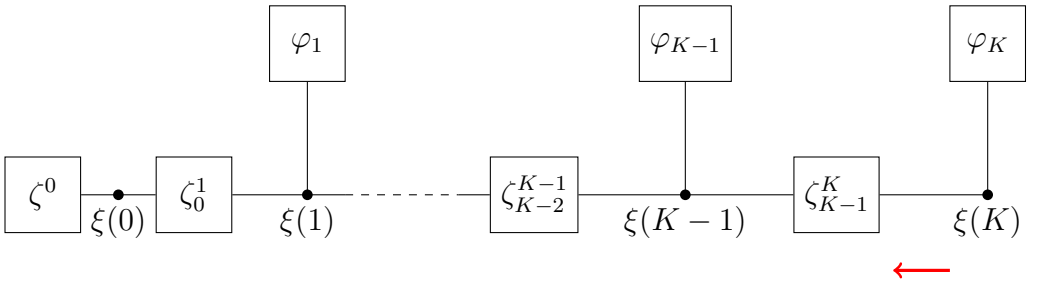

Figure 4.3: The variable $\xi(K)$ sends the message $m_{\xi(K) \to \zeta_{K-1}^K}$ to the factor ζ_{K-1}^K.

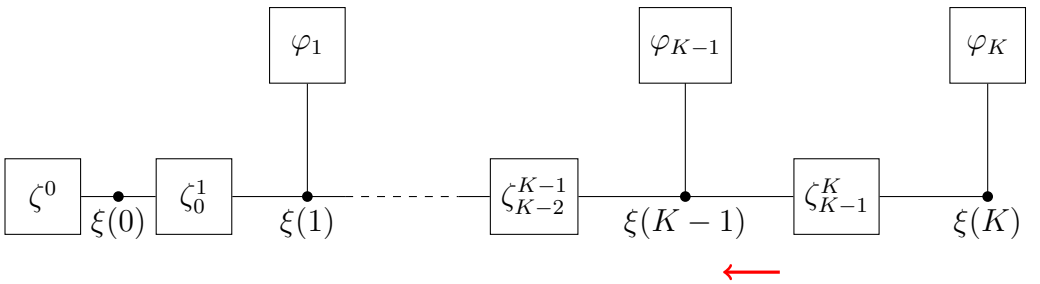

Figure 4.4: The factor ζ_{K-1}^K sends the message $m_{\zeta_{K-1}^K \to \xi(K-1)}$ to the variable $\xi(K-1)$.

Such message is then relayed to the factor node ζ_{K-1}^{K} (see Fig. 4.3), i.e.,

$$m_{\xi(K)\to\zeta_{K-1}^{K}} = \varphi_K \ .$$

which computes the product of such message with the factor ζ_{K-1}^{K}. The resulting message is then sent to the variable $\xi(K-1)$ after the *maximization* (cf. (4.2)) process, performed over the variable $\xi(K)$ (see Fig. 4.4), i.e.,

$$\begin{aligned} m_{\zeta_{K-1}^{K}\to\xi(K-1)} &\propto \max_{\xi(K)} \zeta_{K-1}^{K} \cdot m_{\xi(K)\to\zeta_{K-1}^{K}} \\ &= \max_{\xi(K)} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(\xi(K)-\xi(K-1))^2}{2\sigma^2}\right) \\ &\cdot \exp\left(\alpha_{\xi,K}\xi^2(K) + \beta_{\xi,K}\xi(K)\right) \mathbb{I}(U(K)-\xi(K)) \end{aligned}$$

that can be rewritten in a more general form as

$$\begin{aligned} m_{\zeta_{K-1}^{K}\to\xi(K-1)} \propto \max_{\xi(K)\le U(K)} \exp\big(&A_{\xi,K}\xi^2(K) + B_{\xi,K}\xi^2(K-1) + \\ &+ C_{\xi,K}\xi(K)\xi(K-1) + D_{\xi,K}\xi(K)\big) \end{aligned} \tag{4.8}$$

with

$$\begin{aligned} A_{\xi,K} &= -\frac{1}{2\sigma^2} + \alpha_{\xi,K} \ , & B_{\xi,K} &= -\frac{1}{2\sigma^2} \\ C_{\xi,K} &= \frac{1}{\sigma^2} \ , & D_{\xi,K} &= \beta_{\xi,K} \ . \end{aligned} \tag{4.9}$$

To determine the message (4.8) the procedure is the same as that followed in the proof of Theorem 3.5 with a single available reading. Indeed, by indicating with $\bar{\xi}(K)$ the unconstrained maximizer of the exponent in the objective function, i.e.,

$$\bar{\xi}(K) = \arg\max_{\xi(K)} \left(A_{\xi,K}\xi^2(K) + B_{\xi,K}\xi^2(K-1) + C_{\xi,K}\xi(K)\xi(K-1) + D_{\xi,K}\xi(K)\right) \ ,$$

it follows that

$$\bar{\xi}(K) = -\frac{C_{\xi,K}\xi(K-1) + D_{\xi,K}}{2A_{\xi,K}} \ , \tag{4.10}$$

therefore the solution of the maximization problem in (4.8) is given by

$$\hat{\xi}(K) = \min\left(\bar{\xi}(K), U(K)\right) \ .$$

If $\bar{\xi}(K) \ge U(K)$, the estimation problem is solved. If instead $\bar{\xi}(K) < U(K)$, then the chain has to be traversed backward. On the other hand, the expression of $\bar{\xi}(K)$ depends on $\xi(K-1)$ which is not determined at this stage. To

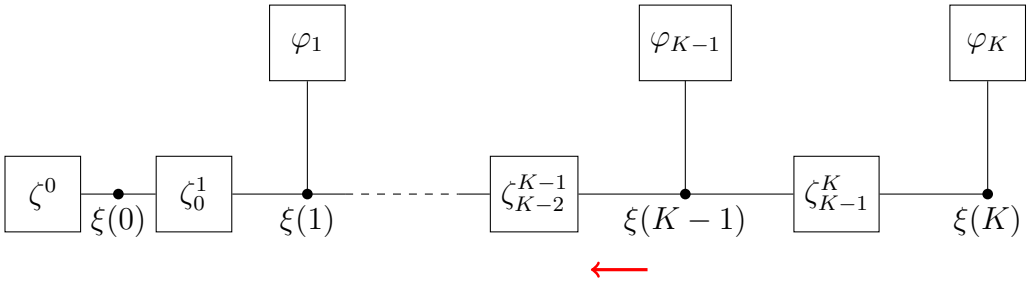

Figure 4.5: The variable $\xi(K-1)$ sends the message $m_{\xi(K-1)\to\zeta_{K-2}^{K-1}}$ to the factor ζ_{K-2}^{K-1}.

complete the computation, further messages need to be propagated backward in the graph from the variable node $\xi(K-1)$.

It is now assumed that $\bar{\xi}(K) \leq U(K)$, therefore $\bar{\xi}(K)$ from (4.10) is plugged back in (4.8) to obtain a modified version of (4.8), that is

$$m_{\zeta_{K-1}^{K}\to\xi(K-1)} \propto \exp\left\{\left(B_{\xi,K} - \frac{C_{\xi,K}^2}{4A_{\xi,K}}\right)\xi^2(K-1) - \frac{C_{\xi,K}D_{\xi,K}}{2A_{\xi,K}}\xi(K-1)\right\}. \tag{4.11}$$

The next step is the message sent to the factor node ζ_{K-2}^{K-1} from the variable node $\xi(K-1)$, which is the product between the message $m_{\zeta_{K-1}^{K}\to\xi(K-1)}$ and the message received from the factor node φ_{K-1} (see Fig. 4.5), i.e.,

$$m_{\xi(K-1)\to\zeta_{K-2}^{K-1}} = m_{\zeta_{K-1}^{K}\to\xi(K-1)} \cdot m_{\varphi_{K-1}\to\xi(K-1)} \,.$$

At this stage, the factor node ζ_{K-2}^{K-1} takes the maximization over $\xi(K-1)$ of the product between the incoming message and the factor ζ_{K-2}^{K-1}, in order to deliver the following message to the variable node $\xi(K-2)$:

$$\begin{aligned}
m_{\zeta_{K-2}^{K-1}\to\xi(K-2)} &\propto \max_{\xi(K-1)\leq U(K-1)} \zeta_{K-2}^{K-1} \cdot m_{\xi(K-1)\to\zeta_{K-2}^{K-1}} \\
&= \max_{\xi(K-1)} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\xi(K-1)-\xi(K-2))^2}{2\sigma^2}\right) \\
&\quad \cdot \exp\left\{\left(B_{\xi,K} - \frac{C_{\xi,K}^2}{4A_{\xi,K}}\right)\xi^2(K-1) - \frac{C_{\xi,K}D_{\xi,K}}{2A_{\xi,K}}\xi(K-1)\right\} \\
&\quad \cdot \exp\left(\alpha_{\xi,K-1}\xi^2(K-1) + \beta_{\xi,K-1}\xi(K-1)\right)\mathbb{I}(U(K-1)-\xi(K-1)) \,.
\end{aligned}$$

As before, the message above can be rewritten in terms of A, B, C and D as

$$\begin{aligned}
m_{\zeta_{K-2}^{K-1}\to\xi(K-2)} \propto \max_{\xi(K-1)\leq U(K-1)} &\exp(A_{\xi,K-1}\xi^2(K-1) + \\
&B_{\xi,K-1}\xi(K-2)^2 + C_{\xi,K-1}\xi(K-1)\xi(K-2) + D_{\xi,K-1}\xi(K-1))
\end{aligned} \tag{4.12}$$

with

$$A_{\xi,K-1} = -\frac{1}{2\sigma^2} + \alpha_{\xi,K-1} + B_{\xi,K} - \frac{C_{\xi,K}^2}{4A_{\xi,K}},$$
$$B_{\xi,K-1} = -\frac{1}{2\sigma^2}, \quad C_{\xi,K-1} = \frac{1}{\sigma^2}$$
$$D_{\xi,K-1} = \beta_{\xi,K-1} - \frac{C_{\xi,K}D_{\xi,K}}{2A_{\xi,K}} .$$

As before, one can compute the unconstrained maximizer $\bar{\xi}(K-1)$ of the objective function above, which turns out to be given by

$$\bar{\xi}(K-1) = -\frac{C_{\xi,K-1}\xi(K-2) + D_{\xi,K-1}}{2A_{\xi,K-1}}$$

and the solution to the maximization problem (4.12) is again expressed as

$$\hat{\xi}(K-1) = \min\left(\bar{\xi}(K-1), U(K-1)\right) .$$

In the same way as previously seen, $\bar{\xi}(K-1)$ depends on $\xi(K-2)$ therefore at least another set of messages must be passed backward on the factor graph (see Fig. 4.6). Once $\bar{\xi}(K-1)$ is plugged back in (4.12), it holds that

$$m_{\zeta_{K-2}^{K-1}\to\xi(K-2)} \propto$$
$$\exp\left\{\left(B_{\xi,K-1} - \frac{C_{\xi,K-1}^2}{4A_{\xi,K-1}}\right)\xi^2(K-2) - \frac{C_{\xi,K-1}D_{\xi,K-1}}{2A_{\xi,K-1}}\xi(K-2)\right\} . \quad (4.13)$$

Note that the above has a form similar to (4.11). By keeping traversing the graph, messages similar to (4.11) and (4.13) are obtained, with opportune definitions for the quantities A, B, C and D.

A generalization of what has just been seen leads to, for $i = 1, \ldots, K-1$

$$A_{\xi,K-i} = -\frac{1}{2\sigma^2} + \alpha_{\xi,K-i} + B_{\xi,K-i+1} - \frac{C_{\xi,K-i+1}^2}{4A_{\xi,K-i+1}}$$
$$B_{\xi,K-i} = -\frac{1}{2\sigma^2}, \quad C_{\xi,K-i} = \frac{1}{\sigma^2} \quad (4.14)$$
$$D_{\xi,K-i} = \beta_{\xi,K-i} - \frac{C_{\xi,K-i+1}D_{\xi,K-i+1}}{2A_{\xi,K-i+1}}$$

and

$$\bar{\xi}(K-i) = -\frac{C_{\xi,K-i}\xi(K-i-1) + D_{\xi,K-i}}{2A_{\xi,K-i}} \quad (4.15)$$
$$\hat{\xi}(K-i) = \min\left(\bar{\xi}(K-i), U(K-i)\right) . \quad (4.16)$$

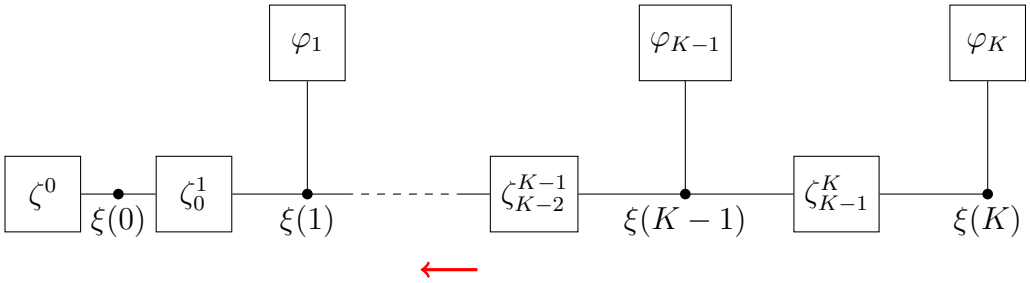

Figure 4.6: The factor ζ_{K-2}^{K-1} sends the message $m_{\zeta_{K-2}^{K-1}\to\xi(K-2)}$ to the variable $\xi(K-2)$.

The message passing keeps going until the end of the graph is reached, which means that, for $i = K-1$ and using (4.15) and (4.16) one obtains

$$\bar{\xi}(1) = -\frac{C_{\xi,1}\xi(0) + D_{\xi,1}}{2A_{\xi,1}} \tag{4.17}$$

$$\hat{\xi}(1) = \min\left(\bar{\xi}(1), U(1)\right) \ . \tag{4.18}$$

Similarly, it follows that

$$m_{\zeta_0^1\to\xi_0} \propto \exp\left\{\left(B_{\xi,1} - \frac{C_{\xi,1}^2}{4A_{\xi,1}}\right)\xi^2(0) - \frac{C_{\xi,1}D_{\xi,1}}{2A_{\xi,1}}\xi(0)\right\} \ . \tag{4.19}$$

The estimate $\hat{\xi}(0)$ can be obtained by maximizing the above message, which naturally leads to an unconstrained maximization issue, thus yielding

$$\begin{aligned}\hat{\xi}(0) = \bar{\xi}(0) &= \max_{\xi(0)} m_{\zeta_0^1\to\xi(0)} \\ \Rightarrow \hat{\xi}(0) &= \frac{C_{\xi,1}D_{\xi,1}}{4A_{\xi,1}B_{\xi,1} - C_{\xi,1}^2} \ .\end{aligned} \tag{4.20}$$

Now that the estimate in (4.20) is fully determinable from the observations: its value can be used to solve for $\hat{\xi}(1)$ in (4.18) and it can then be plugged back in the chain using the recursions (4.15) and (4.16).

4.3.2 Closed-Form Expressions For $\hat{\theta}_{\mathrm{ML}}^{(\mathrm{o})}(K)$

The aim of this section is to provide a concise expression for the estimator $\hat{\theta}^{\mathrm{ML}}(K)$, by defining new quantities: given the linear function of x

$$g_{\xi,k}(x) = -\frac{C_{\xi,k}x + D_{\xi,k}}{2A_{\xi,k}} \ , \tag{4.21}$$

which has a key property, summarized in the following lemma.

Lemma 4.1 *For real numbers x_1 and x_2, the function $g_{\xi,k}(\cdot)$, defined in* (4.21)*, satisfies*

$$g_{\xi,k}\left(\min(x_1, x_2)\right) = \min\left(g_{\xi,k}(x_1), g_{\xi,k}(x_2)\right) \ .$$

Proof: The constants $A_{\xi,k}$, $C_{\xi,k}$ and $D_{\xi,k}$ are defined in (4.9) and (4.14). The proof follows by observing that $\frac{-C_{\xi,k}}{2A_{\xi,k}} > 0$, which implies that $g_{\xi,k}$ is a monotonically increasing function. ■

Using the just defined function $g_{\xi,k}$, the following chain of equalities can be stated as

$$\begin{aligned}
\bar{\xi}(1) &= g_{\xi,1}\left(\hat{\xi}(0)\right) \\
\hat{\xi}(1) &= \min\left(U(1), g_{\xi,1}\left(\hat{\xi}(0)\right)\right) \\
\bar{\xi}(2) &= g_{\xi,2}\left(\hat{\xi}(1)\right) \\
\hat{\xi}(2) &= \min\left(U(2), g_{\xi,2}\left(\hat{\xi}(1)\right)\right)
\end{aligned}$$

with

$$\begin{aligned}
g_{\xi,2}\left(\hat{\xi}(1)\right) &= g_{\xi,2}\left(\min\left(U(1), g_{\xi,1}\left(\hat{\xi}(0)\right)\right)\right) \\
&= \min\left(g_{\xi,2}\left(U(1)\right), g_{\xi,2}\left(g_{\xi,1}\left(\hat{\xi}(0)\right)\right)\right) \qquad (4.22)
\end{aligned}$$

where (4.22) holds thanks to Lemma 4.1. In turn, the estimate $\hat{\xi}(2)$ can be rewritten as

$$\begin{aligned}
\hat{\xi}(2) &= \min\left(U(2), \min\left(g_{\xi,2}\left(U(1)\right), g_{\xi,2}\left(g_{\xi,1}\left(\hat{\xi}(0)\right)\right)\right)\right) \\
&= \min\left(U(2), g_{\xi,2}\left(U(1)\right), g_{\xi,2}\left(g_{\xi,1}\left(\hat{\xi}(0)\right)\right)\right) \ .
\end{aligned}$$

With the same procedure, it holds

$$\hat{\xi}_3 = \min\left(U(3), g_{\xi,3}\left(U(2)\right), g_{\xi,3}\left(g_{\xi,2}\left(U(1)\right)\right), g_{\xi,3}\left(g_{\xi,2}\left(g_{\xi,1}\left(\hat{\xi}(0)\right)\right)\right)\right) .$$

For $m \geq j$, it is convenient to define the function

$$G^m_{\xi,j}(.) = g_{\xi,m}\left(g_{\xi,m-1} \ldots g_{\xi,j}\left(.\right)\right) \ , \qquad (4.23)$$

so that the estimate $\hat{\xi}_3$ can be compactly rewritten as

$$\hat{\xi}_3 = \min\left(U(3), G^3_{\xi,3}\left(U(2)\right), G^3_{\xi,2}\left(U(1)\right), G^3_{\xi,1}\left(\hat{\xi}(0)\right)\right) .$$

It must be noted that the just presented estimators only depends on functions of data, therefore they are fully determined. By keeping on plugging back the quantities in the chain, the estimate for $\xi(K)$ can be readily obtained.

Analogous expressions can be obtained for ψ, by keeping in mind that $A_{\psi,K-i}$, $B_{\psi,K-i}$, $C_{\psi,K-i}$ and $D_{\psi,K-i}$ for $i = 0, \ldots, K-1$, can be obtained from (4.9) and (4.14) by substituting $\alpha_{\xi,K-i}$ and $\beta_{\xi,K-i}$ with the proper $\alpha_{\psi,K-i}$ and $\beta_{\psi,K-i}$, respectively. By making use of these constants, $\hat{\psi}(0)$, $g_{\psi,k}$ and $G^m_{\psi,j}$ can be defined analogously to (4.20), (4.21) and (4.23).

The closed form expression for the clock offset estimate $\hat{\theta}^{(\mathrm{o})}(K)$ is given by the following theorem.

Theorem 4.2 *The state estimates $\hat{\xi}(K)$ and $\hat{\psi}(K)$ for the posterior pdf in* (4.5) *can be expressed as*

$$\hat{\xi}(K) = \min\left(U(K), G^K_{\xi,K}\left(U(K-1)\right), \ldots, G^K_{\xi,2}\left(U(1)\right), G^K_{\xi,1}\left(\hat{\xi}(0)\right)\right)$$
$$\hat{\psi}(K) = \min\left(V(K), G^K_{\psi,K}\left(V(K-1)\right), \ldots, G^K_{\xi,2}\left(V_1\right), G^K_{\xi,1}\left(\hat{\psi}(0)\right)\right)$$

and the factor graph-based clock offset estimator (FGOE) $\hat{\theta}^{(\mathrm{o})}(K)$ is given by

$$\hat{\theta}^{(\mathrm{o})}(K) = \frac{\hat{\xi}(K) - \hat{\psi}(K)}{2} . \tag{4.24}$$

Proof: The proof follows from the previous discussion and by making use of (3.3). ∎

As it has been previously remarked, the closed form expressions in Theorem 4.2 allow to estimate the time-varying clock offset $\theta^{(\mathrm{o})}(K)$, when the likelihood functions, $f(U(k)|\xi(k))$ and $f(V(k)|\psi(k))$, have either a Gaussian, exponential or log-normal distribution. The following sections present the analytical results when the general framework here adopted is particularized to all these three distributions.

Remark *If the likelihood functions $f_U(U(k)|\xi(k))$ and $f_V(V(k)|\psi(k))$ are unconstrained, it follows that the solution of the maximization problem*

$$\max_{\xi(k)} \exp\left(A_{\xi,k}\xi^2(k) + B_{\xi,k}\xi^2(k-1) + C_{\xi,k}\xi(k)\xi(k-1) + D_{\xi,k}\xi(k)\right)$$

coincides with the unconstrained maximizer, i.e., $\hat{\xi}(k) = \bar{\xi}(k)\ \forall k = 1, \ldots, K$. *Hence, the unconstrained likelihood maximization problem is subsumed in the message passing framework for constrained likelihood maximization.*

Gaussian Distribution

In case of Gaussian likelihoods (see Tab. 4.1), the functions $f_U(U(k)|\xi(k))$ and $f_V(V(k)|\psi(k))$ are unconstrained. By keeping in mind this, it follows from Theorem 4.2 that $\hat{\xi}(K)$ for Gaussian distributed observations $U(k)$ is given by

$$\hat{\xi}(K) = G_{\xi,1}^{K}\left(\hat{\xi}(0)\right) .$$

To determine the constants in (4.9) and (4.14) it is sufficient to compare the Gaussian pdf (see Tab.4.1) with (4.7), to get

$$\alpha_{\xi,k} = -\frac{1}{2\sigma_{\xi,k}^2}, \quad \beta_{\xi,k} = \frac{U(k)}{\sigma_{\xi,k}^2} , \tag{4.25}$$

and using these values in (4.9) and (4.14), thus obtaining

$$\begin{aligned}
A_{\xi,K} &= -\frac{1}{2\sigma^2} - \frac{1}{2\sigma_{\xi,K}^2}, \quad B_{\xi,K} = -\frac{1}{2\sigma^2} \\
C_{\xi,K} &= \frac{1}{\sigma^2}, \quad D_{\xi,K} = \frac{U(K)}{\sigma_{\xi,K}^2} \\
A_{\xi,K-i} &= -\frac{1}{2\sigma^2} - \frac{1}{2\sigma_{\xi,K-i}^2} + B_{\xi,K-i+1} - \frac{C_{\xi,K-i+1}^2}{4A_{\xi,K-i+1}} \\
B_{\xi,K-i} &= -\frac{1}{2\sigma^2}, \quad C_{\xi,K-i} = \frac{1}{\sigma^2} \\
D_{\xi,K-i} &= \frac{U(K-i)}{\sigma_{\xi,K-i}^2} - \frac{C_{\xi,K-i+1}D_{\xi,K-i+1}}{2A_{\xi,K-i+1}}
\end{aligned} \tag{4.26}$$

for $i = 1, \ldots, K-1$. Using similar arguments, the estimate $\hat{\psi}(K)$ is given by

$$\hat{\psi}(K) = G_{\psi,1}^{K}\left(\hat{\psi}(0)\right) .$$

Finally, the FGOE $\hat{\theta}^{(\mathrm{o})}(K)$, is expressed (3.3) as

$$\hat{\theta}^{(\mathrm{o})}(K) = \frac{G_{\xi,1}^{K}\left(\hat{\xi}(0)\right) - G_{\psi,1}^{K}\left(\hat{\psi}(0)\right)}{2} . \tag{4.27}$$

It is interesting to observe the behaviour of the FGOE $\hat{\theta}^{(\mathrm{o})}(K)$ when the Gauss-Markov model variance σ^2 tends to zero. Consider

$$g_{\xi,K}(\xi) = -\frac{C_{\xi,K}\xi + D_{\xi,K}}{2A_{\xi,K}}$$

where the constants $A_{\xi,K}$, $B_{\xi,K}$, $C_{\xi,K}$ and $D_{\xi,K}$ are determined by (4.26). After some manipulation, it holds that

$$g_{\xi,K}(\xi) = \frac{\sigma_\xi^2\xi + \sigma^2 U(K)}{\sigma_\xi^2 + \sigma^2} \ ,$$

and it is quite straightforward to see that

$$\lim_{\sigma^2\to 0} g_{\xi,K}(\xi) = \xi \ ,$$

Similarly, $g_{\xi,K-1}(\xi) \to \xi$ as $\sigma^2 \to 0$. It can be shown that in the low system noise regime, as $\sigma^2 \to 0$, it holds that

$$\hat{\xi}(K) \to \hat{\xi}(0) = \frac{C_{\xi,1}D_{\xi,1}}{4A_{\xi,1}B_{\xi,1} - C_{\xi,1}^2} \ .$$

The same relations are valid for ψ, for which it can be shown that

$$\hat{\psi}(K) \to \hat{\psi}(0) = \frac{C_{\psi,1}D_{\psi,1}}{4A_{\psi,1}B_{\psi,1} - C_{\psi,1}^2} \ .$$

Therefore

$$\hat{\theta}^{(\mathrm{o})}(K) \to \frac{\hat{\xi}(0) - \hat{\psi}(0)}{2} \ ,$$

which can be proven to be equal to the MLE (3.19).

Log-Normal Distribution

When the likelihood functions are log-normally distributed (see Tab. 4.1), it holds that

$$\alpha_{\xi,k} = -\frac{1}{2\sigma_{\xi,k}^2} \quad \beta_{\xi,k} = \frac{\log U(k)}{\sigma_{\xi,k}^2} \ .$$

It is clear that the only difference with the Gaussian case is a different expression of $\beta_{\xi,k}$. Being an unconstrained case of likelihood functions, $\hat{\xi}(K)$ in this case is again

$$\hat{\xi}(K) = G_{\xi,1}^K\left(\hat{\xi}(0)\right) \ ,$$

with the constants in (4.9) and (4.14) given by

$$\begin{aligned}
A_{\xi,K} &= -\frac{1}{2\sigma^2} - \frac{1}{2\sigma^2_{\xi,K}}, \quad B_{\xi,K} = -\frac{1}{2\sigma^2} \\
C_{\xi,K} &= \frac{1}{\sigma^2}, \quad D_{\xi,K} = \frac{\log U(K)}{\sigma^2_{\xi,K}} \\
A_{\xi,K-i} &= -\frac{1}{2\sigma^2} - \frac{1}{2\sigma^2_{\xi,K-i}} + B_{\xi,K-i+1} - \frac{C^2_{\xi,K-i+1}}{4A_{\xi,K-i+1}} \\
B_{\xi,K-i} &= -\frac{1}{2\sigma^2}, \quad C_{\xi,K-i} = \frac{1}{\sigma^2} \\
D_{\xi,K-i} &= \frac{\log U(K-i)}{\sigma^2_{\xi,K-i}} - \frac{C_{\xi,K-i+1}D_{\xi,K-i+1}}{2A_{\xi,K-i+1}}
\end{aligned}$$

for $i = 1, \ldots, K-1$. The same procedure leads to similar results about ψ. Hence, the FGOE $\hat{\theta}^{(\mathrm{o})}(K)$ can be expressed as

$$\hat{\theta}^{(\mathrm{o})}(K) = \frac{G^K_{\xi,1}\left(\hat{\xi}(0)\right) - G^K_{\psi,1}\left(\hat{\psi}(0)\right)}{2} \,. \tag{4.28}$$

Again, as the Gauss-Markov system noise $\sigma^2 \to 0$, the FGOE approaches its ML counterpart derived in Chapter 3, cf. (3.20).

Exponential Distribution

If the likelihood functions are exponentially distributed (see Tab. 4.1), it means that they represent a case of constrained likelihoods. The quantities $\alpha_{\xi,k}$ and $\beta_{\xi,k}$ are given by

$$\alpha_{\xi,k} = 0, \quad \beta_{\xi,k} = \lambda_\xi \,,$$

so that the constants $A_{\xi,k}$, $B_{\xi,k}$, $C_{\xi,k}$ and $D_{\xi,k}$ can be written as

$$\begin{aligned}
A_{\xi,k} &= -\frac{1}{2\sigma^2}, \quad B_{\xi,k} = -\frac{1}{2\sigma^2} \\
C_{\xi,k} &= \frac{1}{\sigma^2}, \quad D_{\xi,k} = \lambda_\xi
\end{aligned}$$

for all $k = 1, \ldots, K$. Using Theorem 4.2, it follows that

$$\begin{aligned}
G^K_{\xi,K}(U(K-1)) &= -\frac{C_{\xi,K}U(K-1) + D_{\xi,K}}{2A_{\xi,K}} \\
&= U(K-1) + \lambda_\xi \sigma^2 \\
G^K_{\xi,K-1}(U(K-2)) &= U(K-2) + 2\lambda_\xi \sigma^2
\end{aligned}$$

and so on. The estimator $\hat{\xi}(0)$ can be evaluated as

$$\hat{\xi}(0) = \frac{C_{\xi,1} D_{\xi,1}}{4A_{\xi,1}B_{\xi,1} - C_{\xi,1}^2} = +\infty \ ,$$

so that

$$G_{\xi,1}^K(\hat{\xi}(0)) = +\infty \ . \tag{4.29}$$

Finally, using (4.29) and Theorem 4.2, it holds that

$$\begin{aligned}\hat{\xi}(K) = \min(&U(K), U(K-1) + \lambda_\xi \sigma^2, U(K-2) + \\ &2\lambda_\xi \sigma^2, U(1) + (K-1)\lambda_\xi \sigma^2) \ .\end{aligned} \tag{4.30}$$

Similar results hold for ψ, so that the estimate $\hat{\psi}(K)$ is given by

$$\begin{aligned}\hat{\psi}(K) = \min(&V(K), V(K-1) + \lambda_\psi \sigma^2, V(K-2) + \\ &2\lambda_\psi \sigma^2, V(1) + (K-1)\lambda_\psi \sigma^2)\end{aligned} \tag{4.31}$$

and the combination of the two estimators above provides the estimate $\hat{\theta}^{(\mathrm{o})}(K)$ provides (4.24)

$$\begin{aligned}\hat{\theta}^{(\mathrm{o})}(K) = &\frac{1}{2}\min(U(K), U(K-1) + \lambda_\xi \sigma^2, U(K-2) + \\ &2\lambda_\xi \sigma^2, U(1) + (K-1)\lambda_\xi \sigma^2) - \\ &\frac{1}{2}\min(V(K), V(K-1) + \lambda_\psi \sigma^2, V(K-2) + \\ &2\lambda_\psi \sigma^2 V(1) + (K-1)\lambda_\psi \sigma^2) \ .\end{aligned} \tag{4.32}$$

As the Gauss-Markov system noise $\sigma^2 \to 0$, the above estimator yields

$$\hat{\theta}^{(\mathrm{o})}(K) \to \hat{\theta}_{\mathrm{ML}} = \frac{\min\left(U(K), \ldots, U(1)\right) - \min\left(V(K), \ldots, V(1)\right)}{2} \ ,$$

which is the MLE (3.26).

4.4 Bayesian Statistical Bounds

The purpose of this section is to provide statistical bounds to the clock offset estimation error in the Bayesian framework, similarly to what performed in Section 3.5 for the classical case. As in that case, the likelihood is assumed belonging to the exponential family of distributions with the general expression for the log-partition function, so that the analytical results are valid for

a large class of distributions. Moreover, the results can be used by themselves in other Bayesian estimation contexts.

By assuming the general notation (3.27) and (3.28) for the likelihoods, the following sections provide Bayesian bounds for the variance of the error performed in estimating ρ. To this purpose, the Bayesian Cramer-Rao bound and a Bayesian version of the Chapman-Robbins bound are derived.

4.4.1 Bayesian Cramér-Rao Lower Bound

In this section, the Bayesian Cramér-Rao lower bound (BCRB) is derived, which requires regularity conditions analogous to (3.29) for being computed.

The BCRB states that the variance of the estimator $\hat{\rho}(k)$ of $\rho(k)$ is bounded below by the lower-right sub-matrix of the inverse of the Bayesian information matrix, $J_{\mathrm{CR}}^{-1}(k)$ [56], i.e.,

$$\mathrm{Var}\left(\hat{\rho}(k)\right) \geq J_{\mathrm{CR}}^{-1}(k) = [\boldsymbol{J}_{\mathrm{CR}}^{-1}(k)]_{kk} \tag{4.33}$$

where the Bayesian information matrix is given by

$$\begin{aligned}[\boldsymbol{J}_{\mathrm{CR}}(k)]_{ij} &= \mathbb{E}\left[\frac{\partial \log f(\mathbf{Z}_k, \boldsymbol{\rho}_k)}{\partial \rho(i)} \frac{\partial \log f(\mathbf{Z}_k, \boldsymbol{\rho}_k)}{\partial \rho(j)}\right] \\ &= -\mathbb{E}\left[\frac{\partial^2 \log f(\mathbf{Z}_k, \boldsymbol{\rho}_k)}{\partial \rho(i) \partial \rho(j)}\right] .\end{aligned}$$

The expectation is taken with respect to the joint pdf and the following definitions hold

$$\begin{aligned}\mathbf{Z}_k &= [Z(1), \ldots, Z(k)]' \\ \boldsymbol{\rho}_k &= [\rho(0), \rho(1), \ldots, \rho(k)]' \\ f_Z(Z(k)|\rho(k)) &\propto \exp\left(\eta(Z(k))\rho(k) - \phi_k(\rho(k))\right) .\end{aligned} \tag{4.34}$$

By assuming that the parameter $\rho(k)$ follows a Gauss-Markov model, the evolution can be described by

$$f_\rho(\rho(k)|\rho(k-1)) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\rho(k) - \rho(k-1))^2}{2\sigma^2}\right) . \tag{4.35}$$

In [57], a recursive formula for evaluating the Bayesian sub-matrix was derived, being

$$\begin{aligned}J_{\mathrm{CR}}(k+1) = &- E_{\mathrm{CR}}^{(2)}(k) \left(J_{\mathrm{CR}}(k) + E_{\mathrm{CR}}^{(1)}(k)\right)^{-1} E_{\mathrm{CR}}^{(2)}(k) \\ &+ E_{\mathrm{CR}}^{(3A)}(k) + E_{\mathrm{CR}}^{(3B)}(k)\end{aligned} \tag{4.36}$$

with

$$E_{\mathrm{CR}}^{(1)}(k) = \mathbb{E}\left[-\frac{\partial^2}{\partial\rho^2(k)}\log f_\rho(\rho(k+1)|\rho(k))\right]$$
$$E_{\mathrm{CR}}^{(2)}(k) = \mathbb{E}\left[-\frac{\partial^2}{\partial\rho(k)\partial\rho(k+1)}\log f_\rho(\rho(k+1)|\rho(k))\right]$$
$$E_{\mathrm{CR}}^{(3A)}(k) = \mathbb{E}\left[-\frac{\partial^2}{\partial\rho^2(k+1)}\log f_\rho(\rho(k+1)|\rho(k))\right]$$
$$E_{\mathrm{CR}}^{(3B)}(k) = \mathbb{E}\left[-\frac{\partial^2}{\partial\rho^2(k+1)}\log f_\rho(Z(k+1)|\rho(k+1))\right]$$

and the expectation is again taken with respect to the joint pdf. The following theorem specializes the recursive formula (4.36) to the Gauss-Markov model.

Theorem 4.3 *Assuming the Bayesian framework in* (4.34) *and* (4.35)*, the recursive Bayesian information matrix in* (4.36) *is given by*

$$J_{\mathrm{CR}}(k+1) = \left(\sigma^2 + J_{\mathrm{CR}}^{-1}(k)\right)^{-1} + \sigma_{\eta_k}^2 \tag{4.37}$$

with $J_{\mathrm{CR}}(0) = 0$.

Proof: Using the density functions, $f_\rho(\rho(k)|\rho(k-1))$ in (4.35) and $f_Z(Z_k|\rho(k))$ in (4.34), it can be proved that

$$E_{\mathrm{CR}}^{(1)}(k) = \frac{1}{\sigma^2}\,, \quad E_{\mathrm{CR}}^{(2)}(k) = -\frac{1}{\sigma^2}\,, \quad E_{\mathrm{CR}}^{(3A)}(k) = \frac{1}{\sigma^2}$$

and

$$\begin{aligned} E_{\mathrm{CR}}^{(3B)}(k) &= \int\int \frac{\partial^2\phi_k(\rho(k+1))}{\partial\rho^2(k+1)} f_\rho(\rho(k+1), Z(k+1))d\rho(k+1)dZ(k+1) \\ &= \frac{\partial^2\phi_k(\rho(k+1))}{\partial\rho^2(k+1)} = \sigma_{\eta_k}^2\,. \end{aligned}$$

The thesis easily follows by plugging these quantities in (4.36). ∎

4.4.2 Bayesian Chapman-Robbins Bound

In the case the regularity assumptions required by the BCRB are not satisfied (such as for the constrained likelihood functions), a Bayesian version of the Chapman-Robbins bound (BCHRB) can be used to provide a lower bound on the variance of an estimator of $\rho(k)$.

The BCHRB states that the variance of an estimator $\hat{\boldsymbol{\rho}}_k$ of $\boldsymbol{\rho}_k$ is lower bounded as

$$\mathrm{Var}(\hat{\boldsymbol{\rho}}_k) - \left[T_k(\mathbf{h}_k) - 1\right]^{-1} \mathbf{h}_k \mathbf{h}_k^T \succeq \mathbf{0}$$

with $\succeq$ in the positive semi-definite sense, where

$$T_k(\mathbf{h}_k) = \mathbb{E}\left[\left(\frac{f(\mathbf{Z}_k, \boldsymbol{\rho}_k + \mathbf{h}_k)}{f(\mathbf{Z}_k, \boldsymbol{\rho}_k)}\right)^2\right] ,$$

and $\mathbf{h}_k = [0, h_1, \ldots, h_k]'$.

Theorem 4.4 *The BCHRB for the parameter $\rho(k)$ can be expressed as*

$$\mathrm{Var}(\hat{\rho}(k)) \geq \frac{1}{J_{CH,k}}$$

where

$$J_{CH,k} = \inf_{\mathbf{h}_k} \frac{T_k(\mathbf{h}_k) - 1}{h_k^2}$$

and

$$\begin{aligned} T_k(\mathbf{h}_k) = &\left(\prod_{j=1}^{k} M_\eta^{-2}(h_j) M_\eta(2h_j)\right) \times \\ &\exp\left[\frac{1}{\sigma^2}\sum_{j=1}^{k}(h_j - h_{j-1})^2\right] . \end{aligned} \tag{4.38}$$

Proof: See Appendix 4.A.1 for details. ∎

4.4.3 MSE and Variance of the Estimation Error

As for the classical case, also under the Bayesian framework the bounds on the variance of the estimation error of ξ and ψ (substituting ρ in the likelihoods) can be used to lower bound the MSE of the clock offset $\theta^{(\mathrm{o})}(K)$, by making use of Proposition 3.8.

Gaussian Distribution - BCRB

When the likelihood function for $\xi(k)$ is Gaussian distributed (see Tab. 4.1), by using (4.25) and (4.37), it follows that

$$J_{\mathrm{CR},\xi}(k+1) = \left(\sigma^2 + J_{\mathrm{CR},\xi}^{-1}(k)\right)^{-1} + \frac{1}{\sigma_{\xi,k}^2} ,$$

with $J_{\mathrm{CR},\xi}(0) = 0$. A similar recursion for $J_{\mathrm{CR},\psi}(k)$ can be derived. The MSE of θ_k can be then lower bounded as

$$\mathrm{Var}(\hat{\theta}^{(\mathrm{o})}(k)) \geq \frac{1}{4}\left(\frac{1}{J_{\mathrm{CR},\xi}(k)} + \frac{1}{J_{\mathrm{CR},\psi}(k)}\right) . \tag{4.39}$$

Exponential Distribution - BCHRB

If the likelihood for $\xi(k)$ is exponentially distributed (see Tab. 4.1), (4.38) becomes

$$T_k(\mathbf{h}_k) = \exp\left(\lambda_\xi \sum_{j=1}^{k} h_j\right) \exp\left[\frac{1}{\sigma^2}\sum_{j=1}^{k}(h_j - h_{j-1})^2\right] .$$

In fact, it just needs to be noticed that $\phi_\xi(\xi(k))$ is constant over $\xi(k)$ and $\eta_\xi(U(j)) = \lambda_\xi$, so that (4.41) turns out to be

$$\mathbb{E}\left[\exp\left(2h_j\eta_\xi(U(j))\right)\right] = \exp\left(\lambda_\xi h_j\right)$$

therefore $S(\mathbf{h}_k) = \exp\left(\lambda_\xi \sum_{j=1}^{k} h_j\right)$.

4.5 Numerical Results

Numerical results are provided in this section to corroborate the theoretical findings of this chapter in the various scenarios. As already stated, the performance metric used to evaluate the proposed estimators is the MSE in estimating $\theta^{(\mathrm{o})}(K)$. Parameters are chosen as $\sigma_\xi = \sigma_\psi = 0.1$ for both Gaussian and log-normally distributed likelihoods, while $\lambda_\xi = \lambda_\psi = 10$ for exponentially distributed likelihood functions, so that, as in Chapter 3, the variance of the observations is the same in all the three cases. Finally, the variance of the Gauss-Markov model (4.4) is set to $\sigma = 10^{-4}$.

4.5.1 MSE Performance and Bounds

Assuming either Gaussian or exponentially distributed likelihoods, performance of the FGOEs proposed in Section 4.3 in these two specific cases are here compared with the Bayesian bounds introduced in Section 4.4. Fig. 4.7 reports the MSE performance of the FGOEs $\hat{\theta}^{(\mathrm{o})}(K)$ (4.27) and (4.32) and compares it with the BCRB and the BCHRB. As in the classical estimation scenario, for Gaussian likelihoods the MSE using (4.27) overlaps the reported

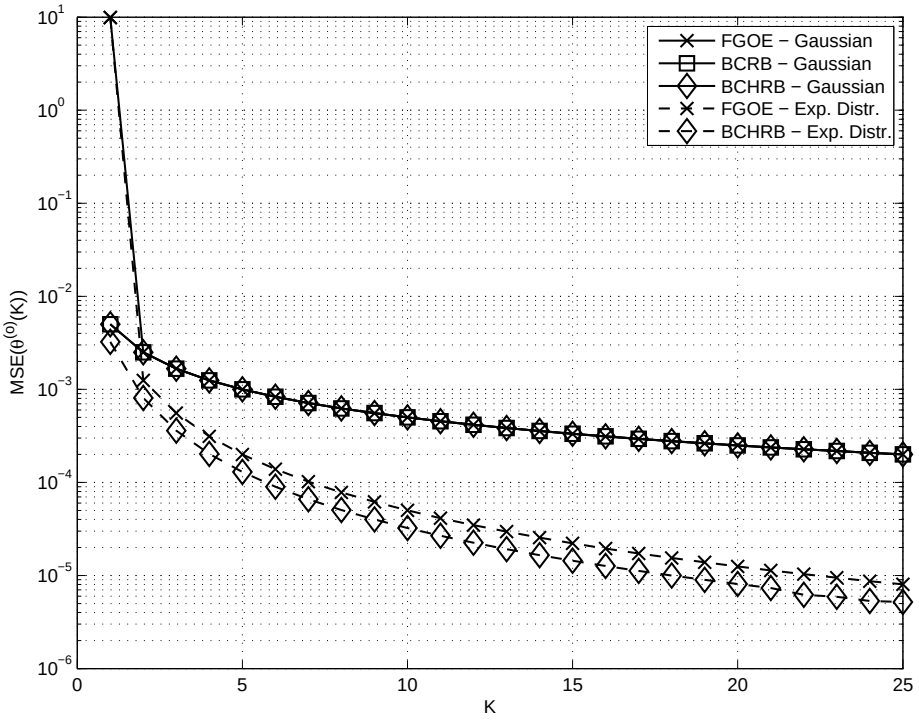

Figure 4.7: MSE and bounds for estimating $\theta^{(\mathrm{o})}(K)$ when using the proposed FGOE with Gaussian and exponentially distributed likelihood.

bounds. The MSE of the FGOE in the case of exponentially distributed likelihoods (4.32) is also plotted against the BCHRB as well in Fig. 4.7. In this case, the MSE is quite close to BCHRB, although it does not coincide with it, as exactly was the case in the classical estimation framework.

4.5.2 MSE Robustness with Log-Normally Distributed Likelihood

Fig. 4.8 shows the MSE in case the likelihood is log-normally distributed and the various FGOEs are used: it can be seen that the FGOEs that (wrongly) assume that the likelihood is either Gaussian or exponentially distributed (dashed and dotted curve, respectively) have a quite bad performance as the MSE is unbounded and unpredictable. On the other hand, the proposed estimator $\hat{\theta}^{(\mathrm{o})}_{\mathrm{ML}}(K)$ (4.28) derived under the assumption of log-normal likelihoods has extremely good performance, establishing a strong improvement with respect the other estimators in this case. Moreover, the MSE curve overlaps the reported BCRB (which, in turn, coincides with the BCHRB), denoting

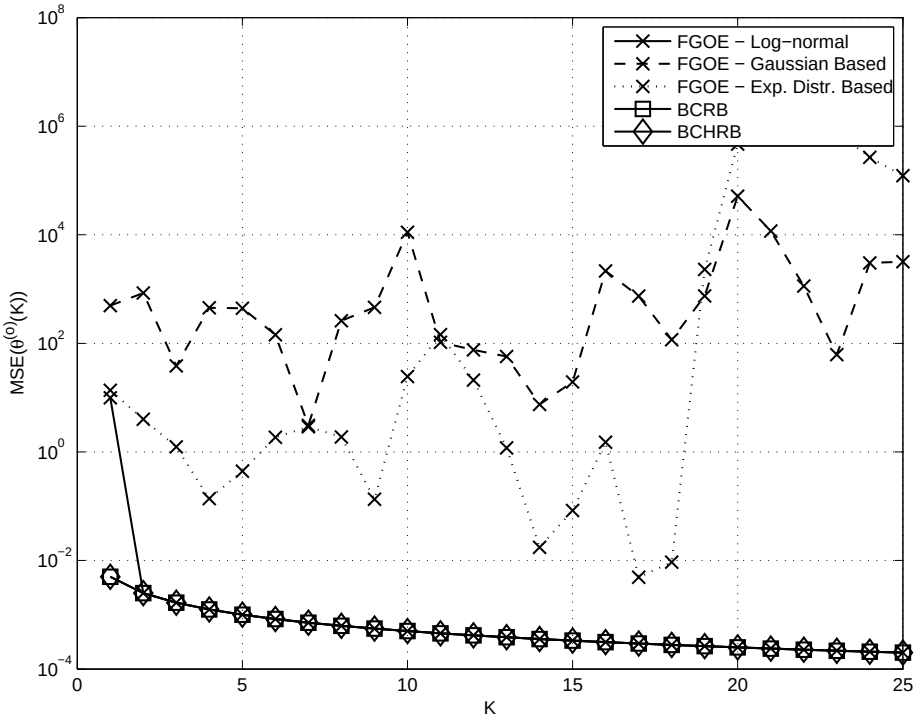

Figure 4.8: MSE and bounds for the estimation of $\theta^{(\mathrm{o})}(K)$ in case of log-normal likelihood.

estimator efficiency.

4.5.3 Classical vs Bayesian framework

The estimators proposed in the classical and the Bayesian framework in Chapter 3 and this chapter, respectively, can also be compared with each other on the basis of their MSE performance as the system noise varies. The goal of this analysis here is to show that the performance of the Bayesian estimators approaches the performance of the MLEs of Chapter 3 as $\sigma \to 0$.

Fig. 4.9 shows the MSE for the cases of Gaussian, exponential and log-normal likelihoods for $K = 25$. In the plot, the MSEs in the classical framework are represented by the horizontal lines obtained with the MLEs using (3.37) and (3.38) (see Appendix 3.A.5). It can be observed how the MSE obtained with the use of the FGOEs for estimating $\theta^{(\mathrm{o})}(K)$ almost approaches the MSE of the MLEs as $\sigma < 10^{-3}$, for all the three considered distributions.

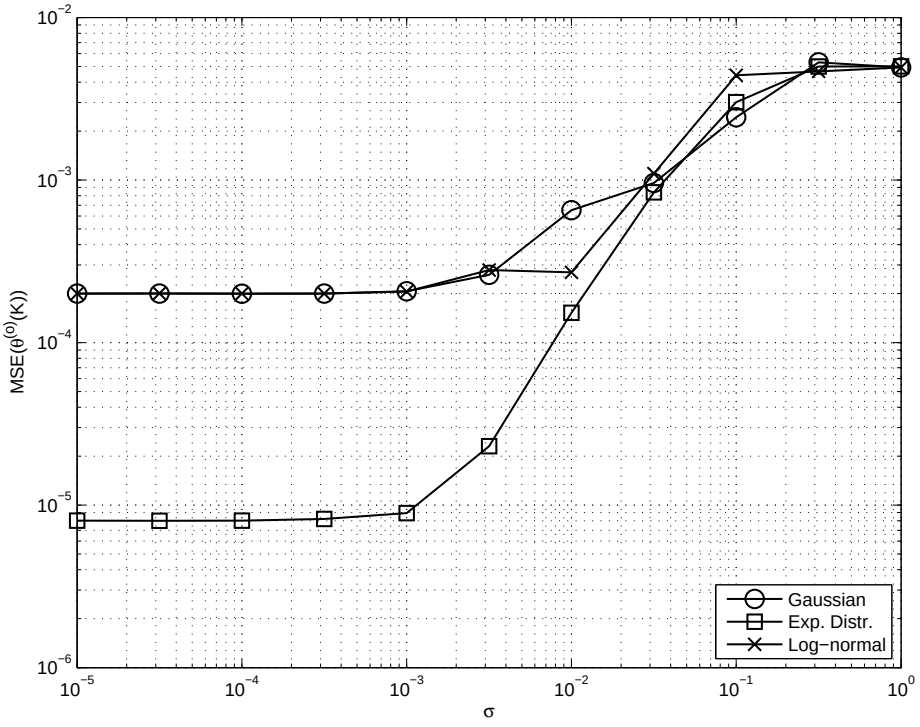

Figure 4.9: MSE in the estimation of $\theta^{(o)}(K)$ vs σ using FGOEs. The MSE in the Bayesian case approaches the curves of the MLE as $\sigma \to 0$.

4.6 Conclusions

In this chapter, the model of Chapter 3 is improved by introducing a Gauss-Markov model regulating the time behaviour of the clock offset related quantities ξ and ψ. Extending the clock model by incorporating time-varying offset provides a more realistic scenario for actual oscillators in sensor nodes. Based on this, the general framework for the likelihood functions of the observations and the theory of factor graphs and message passing are exploited in order to derive closed-form expressions for clock offset estimators under the Bayesian framework, via the use of max-product algorithm. Statistical bounds derived under the Bayesian framework in order to assess the quality of the proposed FGOEs. The provided bounds are valid for a generic distribution of the exponential family, so that they can also be used in scenarios other than clock synchronization. The reported numerical studies confirm the good behaviour of the proposed estimators. It can be concluded that the FGOEs introduced in this chapter are valid alternatives to the state-of-the-art in clock offset estimator when time-varying clock offset is realistically assumed.

4.A Appendix

4.A.1 Proof of Theorem 4.4

It holds that

$$\begin{aligned} T_k(\mathbf{h}_k) =& \mathbb{E}\left[\left(\frac{f(\mathbf{Z}_k, \boldsymbol{\rho}_k + \mathbf{h}_k)}{f(\mathbf{Z}_k, \boldsymbol{\rho}_k)}\right)^2\right] \\ =& \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\left(\frac{f(\mathbf{Z}_k, \boldsymbol{\rho}_k + \mathbf{h}_k)}{f(\mathbf{Z}_k, \boldsymbol{\rho}_k)}\right)^2 f(\mathbf{Z}_k, \boldsymbol{\rho}_k) d\mathbf{Z}_k d\boldsymbol{\rho}_k \\ =& S(\mathbf{h}_k)\int_{-\infty}^{+\infty}\frac{f(\boldsymbol{\rho}_k + \mathbf{h}_k)^2}{f(\boldsymbol{\rho}_k)} d\boldsymbol{\rho}_k \end{aligned}$$

with

$$S(\mathbf{h}_k) = \int_{-\infty}^{+\infty}\left(\frac{f(\mathbf{Z}_k|\boldsymbol{\rho}_k + \mathbf{h}_k)}{f(\mathbf{Z}_k|\boldsymbol{\rho}_k)}\right)^2 f(\mathbf{Z}_k|\boldsymbol{\rho}_k) d\mathbf{Z}_k \,. \tag{4.40}$$

Continuing with the calculations

$$\begin{aligned} T_h(\mathbf{h}_k) =& S(\mathbf{h}_k)\int_{-\infty}^{+\infty}\frac{f(\boldsymbol{\rho}_k + \mathbf{h}_k)^2}{f(\boldsymbol{\rho}_k)} d\boldsymbol{\rho}_k \\ =& S(\mathbf{h}_k)\int_{-\infty}^{+\infty}\frac{f^2(\rho_0 + h_0)}{f(\rho_0)} \times \\ & \prod_{j=1}^{k}\frac{f^2(\rho(j) + h_j|\rho_{j-1} + h_{j-1})}{f(\rho(j)|\rho_{j-1})} d\boldsymbol{\rho}_k \\ =& S(\mathbf{h}_k)\int_{-\infty}^{+\infty}\prod_{j=1}^{k}\frac{f^2(\rho(j) + h_j|\rho_{j-1} + h_{j-1})}{f(\rho(j)|\rho_{j-1})} d\boldsymbol{\rho}_k \,. \end{aligned}$$

Since the quantity $\frac{f^2(\rho(j)+h_j|\rho_{j-1}+h_{j-1})}{f(\rho(j)|\rho_{j-1})}$ results to be equal to $(j = 1, \ldots, k)$

$$\begin{aligned} & \frac{1}{\sigma\sqrt{2\pi}}\exp\left[-\frac{\rho_{j-1}^2}{2\sigma^2} + \frac{\rho(j) + 2(h_j - h_{j-1})}{\sigma^2}\rho_{j-1}\right] \times \\ & \exp\left[-\frac{(h_j - h_{j-1})^2}{\sigma^2}\right]\exp\left[-\frac{\rho(j)^2}{2\sigma^2} - \frac{2\rho(j)(h_j - h_{j-1})}{\sigma^2}\right] \end{aligned}$$

it follows that

$$\int_{-\infty}^{+\infty}\frac{f^2(\rho(j) + h_j|\rho_{j-1} + h_{j-1})}{f(\rho(j)|\rho_{j-1})} d\rho_{j-1} = \exp\left[\frac{(h_j - h_{j-1})^2}{\sigma^2}\right]$$

to get

$$T_k(\mathbf{h}_k) = S(\mathbf{h}_k) \exp\left[\frac{1}{\sigma^2}\sum_{j=1}^{k}(h_j - h_{j-1})^2\right] .$$

Moreover, (4.40) can be written as

$$S(\mathbf{h}_k) = \prod_{j=1}^{k}\int_{-\infty}^{+\infty}\left(\frac{f(Z_j|\rho(j)+h_j)}{f(Z_j|\rho(j))}\right)^2 f(Z_j|\rho(j))dZ_j ,$$

and it can be noted that

$$\left(\frac{f(Z_j|\rho(j)+h_j)}{f(Z_j|\rho(j))}\right)^2 = \exp\left[-2\left(\phi_\rho(\rho(j)+h_j) - \phi_\rho(\rho(j))\right)\right] \times \exp\left(2h_j\eta_\rho(Z_j)\right) .$$

Therefore

$$\int_{-\infty}^{+\infty}\left(\frac{f(Z_j|\rho(j)+h_j)}{f(Z_j|\rho(j))}\right)^2 f(Z_j|\rho(j))dZ_j = M_{\eta_\rho}^{-2}(h_j)\times \mathbb{E}\left[\exp\left(2h_j\eta_\rho(Z_j)\right)\right] .$$

Then, since

$$\mathbb{E}\left[\exp\left(2h_j\eta_\rho(Z_j)\right)\right] = \exp\left(\phi_\rho(\rho(j)+2h_j) - \phi_\rho(\rho(j))\right) \tag{4.41}$$

it can be easily proved that

$$\int_{-\infty}^{+\infty}\left(\frac{f(Z_j|\rho(j)+h_j)}{f(Z_j|\rho(j))}\right)^2 f(Z_j|\rho(j))dZ_j = M_{\eta_\rho}^{-2}(h_j)M_{\eta_\rho}(2h_j) ,$$

so yielding

$$S(\mathbf{h}_k) = \prod_{j=1}^{k} M_{\eta_\rho}^{-2}(h_j)M_{\eta_\rho}(2h_j) .$$

Part II

Fast Consensus in WSNs and its Application to Clock Synchronization

Clock coupling is well represented by the relation between oscillators when consensus is run for achieving clock synchronization. In Part II of this thesis an innovative consensus algorithm is presented, as well as its application to the clock synchronization problem.

Chapter 5

Consensus and Clock Synchronization

Differently from the results presented in Part I, the second part of this thesis relies on synchronization of coupled oscillators. Coupling between clocks of different wireless sensor nodes involves the implementation of a correcting signal to the oscillators (acting on the clock phase and/or on the clock frequency) which modifies their running behaviors. In fact, while in Part I the goal is the estimation of clock offsets (or skews) between different nodes, the algorithms proposed in this part instead actively modify the oscillation frequency of the oscillators by physically applying a voltage signal to the oscillators through well-known suited structures, such as PLLs or DLLs. Since the correcting signals are functions of neighbors clock-related readings, after the first iteration the clocks of the network will result coupled together, requiring an analysis fundamentally different from that performed in Part I. Consensus algorithms are well suited distributed procedures whose implementation for clock synchronization purposes cause coupling of the network oscillators, as it will be clear next.

This chapter describes the fundamentals of consensus algorithms, so that the relevant performance improvement permitted by the innovative consensus algorithm presented in Chapter 6 can be fully appreciated. Of course, the primary objective of the present thesis is reaching clock synchronization, therefore the reader will not be surprised of seeing in Chapter 7 the application to clock synchronization of the just mentioned innovative results on consensus.

5.1 Distributed Consensus in WSNs

The recent technology advancements in sensor equipment allowed the research community to propose (and eventually implement on real hardware) efficient techniques for distributed processing in WSNs [1]. In fact, sensors are now equipped with on-board processors that allow for local pre-processing of raw sensed data, thus reducing the need of data transfer to the hub nodes. This concurs to make the network a robust and decentralized entity.

Given local sensor readings, a common form of distributed processing consists of reaching an agreement on a common value. More specifically, starting from sensed data at each node, the process of exchanging data-related information in order to agree on a common representative value is called *distributed consensus*.

Applications of consensus can be found in a variety of scenarios in WSNs. Sensors gather data from the environment, so that each sensor node can get information regarding environmental conditions such as temperature, pressure and humidity, or about ongoing phenomena like target movements. Distributed averaging on sensed data is useful for reducing the uncertainty on the measures. It also represents one of the most studied cases in which distributed consensus can be successfully applied. In this case, since the common agreement value is the average of the initial readings, the research community refers to it as *average consensus* [58–61]. Consensus also finds application in other areas, such as sensor fusion [62], just to cite one. Also, clock synchronization can be recast as a double consensus problem (over both clock offsets and skews) [9, 16–18], as it will be shown in Section 5.4 and in Chapter 7.

The classical implementation of consensus is through a linear update of the readings, as it will be explained later in Section 5.2. On the other hand, alternative consensus approaches exist, like, for example, asynchronous gossip algorithms [63, 64]. A method which uses similarities with fluid mechanics is described in [61], but it is currently applicable only to highly structured networks. A further interesting alternative approach is to apply the *alternating direction multipliers method* (ADMM) for reaching an agreement throughout the network (see [65, p. 253] and [66–69]). The whole Chapter 6 is devoted to the analysis of this method.

As in most distributed algorithms, the main design issues are convergence speed and resilience to noise. For maximizing the first, in the context of (linear) average consensus [70] provides an iterative method for computing the best update weights at each sensor node. Throughout this thesis, such method is denoted as the *optimal Boyd's solution*. Although the compu-

tational burden for computing it prevents from an implementation on real sensor nodes, this optimized solution is anyway taken into account in this thesis, at least as a benchmark. Applying a polynomial filter to the network matrix is the method chosen in [71] for increasing the convergence rate. In the realm of gossip algorithms, techniques for increasing the convergence speed can be found in [72]. However, the state-of-the-art of fast consensus for convergence rate is [73], where the past values of the states are incorporated in the algorithm in order to speed up the convergence. Unfortunately, this comes at the cost of a bad noise resilience, which can potentially prevent from recovery of the correct consensus value.

It appears evident that the problem of distributed consensus has been well studied in the literature, both in continuous and in discrete time. In this chapter, basics of discrete time linear consensus are introduced as well as the framework for its application to the problem of clock synchronization.

5.2 Consensus System Model

In a WSN, just couples of nodes that are in the transmission range of one another are actually able to communicate reliably. In order to deal with these network communication constraints, some tools borrowed from the graph theory are used. Here it is assumed that the communication properties of the network are described by an undirected[1] graph of N nodes, one for each sensor. An edge connects nodes i and j if they are in the transmission range one of each other, or, in other words, if i can receive packets from j reliably and vice-versa. The link existence information is summarized in the adjacency matrix $\mathbf{\Omega}$, which has 1 in position (i, j) if node i can transmit information to node j, 0 otherwise. In other words,

$$[\mathbf{\Omega}]_{i,j} = \begin{cases} 1 & \text{if node } j \text{ can communicate to node } j \\ 0 & \text{otherwise .} \end{cases} \tag{5.1}$$

The adjacency matrix of an undirected graph is symmetric by definition, i.e. $\mathbf{\Omega} = \mathbf{\Omega}^{'}$.

In this thesis it is assumed that the underlying communication graph is *strongly connected* besides undirected. This means that there exists a path connecting each node to every other node in the network, so that communication is possible among any pair of nodes, via multi-hop transmissions.

An example of a graph and its adjacency matrix is provided in Figg. 5.1 and 5.2, respectively.

[1] In this thesis, communication links are assumed symmetric.

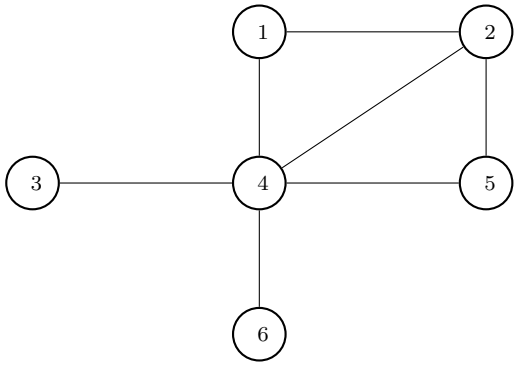

Figure 5.1: Example of an undirected and strongly connected graph.

$$\boldsymbol{\Omega} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} .$$

Figure 5.2: Adjacency matrix relative to the network whose graph is represented in Fig. 5.1.

In consensus theory, convergence to a common value is reached through the iterative exchange and processing of data coming from neighbors at each sensor node. Mathematically speaking, let $x_i(0) = \theta_i \in \mathbb{R}$ be the initial reading[2] of sensor $i \in [1, \ldots, N]$. Let also $x_i(t)$ be the estimated consensus value at time t. The estimated consensus value at time $t+1$, $x_i(t+1)$, is linearly dependent on the values $x_j(t)$, $j \in \mathcal{N}_i$, where $\mathcal{N}_i$ denotes the set of neighbors of node i. Let $\boldsymbol{x}(t) = [x_1(t), \ldots, x_N(t)]^{'}$ be the column vector form of the estimated values at time t. The classical formulation of the *linear* consensus problem relates the states at time $t+1$ to those at time t as follows [58, 63, 70, 74, 75]

$$\boldsymbol{x}(t+1) = \boldsymbol{S}\boldsymbol{x}(t), \qquad t \geq 0, \qquad \boldsymbol{x}(0) = \boldsymbol{\theta} \, , \tag{5.2}$$

where $\boldsymbol{S} \in \mathbb{R}^{N \times N}$ is a square matrix coherent with the communication constraints of the graph, i.e. $[\boldsymbol{S}]_{i,j} = 0$ if i cannot communicate with j.

[2]The work can be easily extended to vectorial case.

5.3 Requirements for Consensus Matrix

A lot of approaches assimilable to (5.2) are actually identified under the name of consensus algorithms, but their common property is the convergence toward a common value to all the agents in the network. What makes the difference in terms of convergence speed and noise resilience is the choice of update matrix $\boldsymbol{S}$.

Consensus is reached when vector $\boldsymbol{x}(t)$ has equal components, corresponding to a linear combination of the elements in the initial readings vector $\boldsymbol{x}(0)$, i.e.,

$$\lim_{t\to\infty} \boldsymbol{x}(t) = \mathbf{1}\left(\boldsymbol{\ell}'\boldsymbol{x}(0)\right) , \qquad \boldsymbol{\ell} \in \mathbb{R}^N , \tag{5.3}$$

where $\mathbf{1}$ is the $N \times 1$ vector of all ones. By an iterative application of (5.2), it is true also that

$$\lim_{t\to\infty} \boldsymbol{x}(t) = \lim_{t\to\infty} \boldsymbol{S}^t\boldsymbol{x}(0) , \tag{5.4}$$

therefore, from (5.3) and (5.4) it must be that

$$\lim_{t\to\infty} \boldsymbol{S}^t = \mathbf{1}\boldsymbol{\ell}' , \tag{5.5}$$

which means that the limit (5.5) must exist and be a rank 1 matrix. The following theorem states equivalent conditions for reaching consensus, based on properties of matrix $\boldsymbol{S}$.

Theorem 5.1 *Let $\lambda_1, \ldots, \lambda_N$ be the eigenvalues of matrix $\boldsymbol{S}$ sorted in a decreasing order. Necessary and sufficient conditions for reaching consensus, i.e., for which* (5.5) *holds, are*

C1) $\boldsymbol{S}\mathbf{1} = \mathbf{1}$

C2) $\boldsymbol{\ell}'\boldsymbol{S} = \boldsymbol{\ell}'$, *with* $\|\boldsymbol{\ell}\| : \boldsymbol{\ell}'\mathbf{1} = 1$

C3) $\rho(\boldsymbol{S}) = 1$

C4) $\mu(\lambda_1 = 1) = 1$,

where $\rho(\boldsymbol{S})$ denotes the spectral radius of matrix $\boldsymbol{S}$, and $\mu(\lambda)$ is the algebraic multiplicity of the eigenvalue λ.

Proof: See Appendix 5.A.1. ∎

5.3.1 Average Consensus

In this thesis, the focus is on linear *average consensus*, that is consensus on the average of the initial readings

$$\overline{\theta} = \frac{1}{N}\mathbf{1}^{'}\boldsymbol{\theta} = \frac{1}{N}\sum_{i=1}^{N}\theta_i \ ,$$

which means that it must be

$$\boldsymbol{\ell}^{'}\boldsymbol{x}(0) = \boldsymbol{\ell}^{'}\boldsymbol{\theta} = \overline{\theta} \ . \tag{5.6}$$

The following corollary simply follows from (5.6) [76].

Corollary 5.2 *Under the assumptions of Theorem 5.1, average consensus is achieved if and only if*

C5) $\boldsymbol{\ell} = \frac{1}{N}\mathbf{1}$.

5.3.2 Popular Average Consensus Matrices

From the set of all matrices satisfying the properties C1)-C5) for achieving average consensus, only four relevant matrices are chosen and described. Three of them are relatively easy to implement since they require simple local knowledge of the neighborhood at each node. On the other hand, the last matrix is reported as a benchmark since it is the result of an optimization process, not easy at all to implement in a distributed fashion.

Consensus Laplacian

The Laplacian matrix $\boldsymbol{S}_{\mathrm{LA}}$ of the graph [16,58,60] is the most popular matrix for reaching average consensus, being easy to implement distributively. It is constructed with the following procedure.

First compute the diagonal matrix $\boldsymbol{\Gamma}$ whose diagonal element g_i in position i is the number of neighbors of node i, i.e. $g_i = \sum_{j\neq i}[\boldsymbol{\Omega}]_{i,j}$. The Laplacian consensus matrix is computed as $\boldsymbol{S}_{\mathrm{LA}} = \boldsymbol{I} - (\boldsymbol{\Gamma} - \boldsymbol{\Omega})$, where $\boldsymbol{I}$ is the $N \times N$ identity matrix.

Consensus Metropolis

The Metropolis-Hastings matrix $\boldsymbol{S}_{\mathrm{MH}}$ [70,77] associated to the graph requires a two-hop neighbors knowledge of the network by each node. It constructed with the following procedure.

First consider the symmetric stochastic matrix $\boldsymbol{S}_{\mathrm{s}}$ defined as follows

$$[\boldsymbol{S}_{\mathrm{s}}]_{i,j} = \begin{cases} \frac{1}{\max\{g_i, g_j\}} & \text{if } (i,j) \text{ is an edge of the graph and } i \neq j \\ 1 - \sum_{j \neq i} [\boldsymbol{S}_{\mathrm{s}}]_{i,j} & \text{if } i = j \\ 0 & \text{otherwise .} \end{cases} \tag{5.7}$$

The Metropolis-Hastings matrix is defined as $\boldsymbol{S}_{\mathrm{MH}} = \boldsymbol{I} - \boldsymbol{S}_{\mathrm{s}}$.

Averaged Consensus Matrix

The averaged consensus matrix $\boldsymbol{S}_{\mathrm{AC}}$ [78] is as simple to implement as the Laplacian matrix, since the required network knowledge is the same. It is directly derived from the adjacency matrix $\boldsymbol{\Omega}$ of the graph.

The procedure for constructing it is to set each element of a row of $\boldsymbol{S}_{\mathrm{AC}}$ equal to 1 over the number of non-zero elements of that specific row.

Boyd's Optimum Solution

The three previous matrices are constructed via an empirical procedure. On the other hand, Boyd *et. al.* in [70] provide an iterative procedure to find the specific matrix $\boldsymbol{S}_{\mathrm{boyd}}$ which optimizes the convergence rate of the algorithm (5.2), through the minimization of the second eigenvalue in modulus. Its implementation requires a high computational cost, therefore in this thesis such matrix is used as a benchmark rather than being suggested for practical implementation.

5.4 Application of Consensus to Clock Synchronization

Clock synchronization is a well suited application of distributed consensus in which agreement is required both for clock offsets and for clock skews. There is an extensive literature on consensus-based clock synchronization algorithms. Simeone *et. al.* in [16] applied the average consensus for reaching an agreement on clock frequencies only. On the other hand, in [17, 18] both clock phases and clock frequencies are corrected, even if in [18] it is not clear how frequency observations are obtained, since they are generally not available at the nodes. An interesting algorithm was proposed in [59], where link delays are carefully taken into account into the analysis.

A formal statement of the problem of clock synchronization in consensus terms is now provided.

5.4.1 System Model

By stacking the clock counter values (1.1) and the clock periods (1.2) in the vectors $\boldsymbol{T}(t) \triangleq [T_1(t), \ldots, T_N(t)]'$ and $\boldsymbol{\Upsilon}(t) \triangleq [\Upsilon_1(t), \ldots, \Upsilon_N(t)]'$, respectively, the problem of clock synchronization is now recast as a double consensus problem, both for agreement on counters and on periods. By assuming initial conditions $\boldsymbol{T}(0)$ and $\boldsymbol{\Upsilon}(0)$, for clock phases and periods, respectively, the adopted reference model is [9, 16–18]

$$\boldsymbol{T}(t+1) = \boldsymbol{T}(t) + \boldsymbol{\Upsilon}(t) + \boldsymbol{u}(t)\,, \tag{5.8a}$$

$$\boldsymbol{\Upsilon}(t+1) = \boldsymbol{\Upsilon}(t) + \boldsymbol{v}(t) \tag{5.8b}$$

where $\boldsymbol{u}(t)$ and $\boldsymbol{v}(t)$ are $N \times 1$ real-valued vectors, which represent the consensus-based control signals to be designed for reaching synchronization.

5.4.2 Relevant Works

The update equations (5.8) are the reference model for a number of consensus-based clock synchronization algorithms, which obviously differ in the expressions of the correcting signals $\boldsymbol{u}(t)$ and $\boldsymbol{v}(t)$. The most relevant works are:

a) Simeone *et. al.* in [16] proposed a clock synchronization algorithm in which only correction on the clock counters is performed, i.e., $\boldsymbol{v}(t) \equiv 0$. A consensus algorithm is instead applied to clock phases, so that $\boldsymbol{u}(t) = (\boldsymbol{S} - \boldsymbol{I})\,\boldsymbol{T}(t)$, with $\boldsymbol{S}$ being a row-stochastic matrix compatible with the underlying graph. Weights in matrix $\boldsymbol{S}$ are chosen in order to give more importance to more reliable (in terms of received power) channels.

b) On the other hand, Schenato *et. al* in [17] designed a consensus-based clock synchronization algorithm in which both consensus on clock phases and frequencies are considered, so that the control signals are expressed as $\boldsymbol{u}(t) = (\boldsymbol{S} - \boldsymbol{I})\,\boldsymbol{T}(t)$ and $\boldsymbol{v}(t) = (\boldsymbol{S} - \boldsymbol{I})\,\boldsymbol{\Upsilon}(t)$. The implementation of such an algorithm is assured by the fact that sensors are assumed able to measure the period ratios $\Upsilon_i(t)/\Upsilon_j.(t)$.

c) Another relevant clock synchronization protocol is the one proposed by Carli *et. al.* in [18]. Consensus both in clock phases and in clock periods is applied, so that the control signal can still be expressed as $\boldsymbol{u}(t) = (\boldsymbol{S} - \boldsymbol{I})\,\boldsymbol{T}(t)$ and $\boldsymbol{v}(t) = \alpha\,(\boldsymbol{S} - \boldsymbol{I})\,\boldsymbol{T}(t)$, with $\alpha \in \mathbb{R}$. On the other hand, while the control $\boldsymbol{u}(t)$ can be straightforwardly applied, for an application of $\boldsymbol{v}(t)$ the values of $\boldsymbol{\Upsilon}(t)$ are needed, but they are not directly measurable at the sensors.

In Chapter 6 an innovative consensus algorithm is presented, which allows a convergence speed on the order of the state-of-the-art of fast consensus and an outstanding noise resilience. This algorithm will be applied to clock synchronization in Chapter 7.

5.A Appendix

5.A.1 Proof of Theorem 5.1

For the sufficiency, first note that C1) and C2) imply that a right and a left eigenvector relative to eigenvalue $\lambda_1 = 1$ are $\mathbf{1}$ and $\boldsymbol{\ell}$, respectively. Let $\boldsymbol{Q}$ be the non singular matrix such that it holds

$$\boldsymbol{S} = \boldsymbol{Q}\boldsymbol{\Lambda}_P\boldsymbol{Q}^{-1} \quad \boldsymbol{\Lambda}_P = \begin{bmatrix} 1 & \\ & \boldsymbol{S}_2 \end{bmatrix} ,$$

where it is assured from C3) and C4) that $\boldsymbol{S}_2$ is a $(N-1)\times(N-1)$ matrix with $\rho(\boldsymbol{S}_2) < 1$. It then holds that

$$\boldsymbol{S} - \mathbf{1}\boldsymbol{\ell}^{'} = \boldsymbol{Q} \begin{bmatrix} 0 & \\ & \boldsymbol{S}_2 \end{bmatrix} \boldsymbol{Q}^{*} ,$$

so that it is

$$\rho\left(\boldsymbol{S} - \mathbf{1}\boldsymbol{\ell}^{'}\right) < 1 . \tag{5.9}$$

Therefore, by following the steps in [76, Thm. 1]

$$\begin{aligned} \boldsymbol{S}^t - \mathbf{1}\boldsymbol{\ell}^{'} &= \boldsymbol{S}^t\left(\boldsymbol{I} - \mathbf{1}\boldsymbol{\ell}^{'}\right) \\ &= \boldsymbol{S}^t\left(\boldsymbol{I} - \mathbf{1}\boldsymbol{\ell}^{'}\right)^t \\ &= \left[\boldsymbol{S}\left(\boldsymbol{I} - \mathbf{1}\boldsymbol{\ell}^{'}\right)\right]^t \\ &= \left(\boldsymbol{S} - \mathbf{1}\boldsymbol{\ell}^{'}\right)^t , \end{aligned} \tag{5.10}$$

where the first equality descends from C1) and the second from C2). By using (5.9), result (5.10) implies (5.5).

For proving necessity, it must be noted first [76] that the limit $\lim_{t\to\infty}\boldsymbol{S}^t$ exists if and only if there is a non singular matrix $\boldsymbol{Q}$ such that

$$\boldsymbol{S} = \boldsymbol{Q} \begin{bmatrix} \boldsymbol{I}_r & \\ & \boldsymbol{S}_3 \end{bmatrix} \boldsymbol{Q}^{-1} ,$$

with $\boldsymbol{I}_r$ is the $r\times r$ identity matrix ($0 \le r \le N$) and $\rho(\boldsymbol{S}_3) < 1$. This implies that $\rho(\boldsymbol{S}) \le 1$. By denoting with $\boldsymbol{\pi}_i$ the columns of matrix $\boldsymbol{Q}$ and with $\boldsymbol{\omega}_i$ the rows of matrix $\boldsymbol{Q}^{-1}$, it is

$$\lim_{t\to\infty}\boldsymbol{S}^t = \lim_{t\to\infty}\boldsymbol{Q}\begin{bmatrix} \boldsymbol{I}_r & \\ & \boldsymbol{S}_3^t \end{bmatrix}\boldsymbol{Q}^{-1} = \boldsymbol{Q}\begin{bmatrix} \boldsymbol{I}_r & \\ & \mathbf{0} \end{bmatrix}\boldsymbol{Q}^{-1} = \sum_{i=1}^{r}\boldsymbol{\pi}_i\boldsymbol{\omega}_i^{'} .$$

Since (5.5) holds, r must be equal to 1 and $\boldsymbol{\pi}_i\boldsymbol{\omega}_i^{'} = \mathbf{1}\boldsymbol{\ell}^{'}$. This implies that C1) and C2) hold and that $\lambda_1 = 1$ is a simple eigenvalue of matrix $\boldsymbol{S}$, therefore proving also C3) and C4).

Chapter 6

Fast Consensus via ADMM

The results presented in this chapter refer to a collaboration of Prof. Vangelista and me with Prof. Tomaso Erseghe[1] and Dr. Emiliano Dall'Anese[2]. They have been published in [78].

6.1 Introduction

As differently from the linear consensus presented in Chapter 5, in the present chapter the problem of distributed agreement is faced from a rather different perspective. The application of the ADMM leads in fact to an innovative reformulation of the average consensus problem, as it will be clear next.

The adopted approach here is a recasting of the average consensus problem presented in Section 5.3.1 as an instance of convex optimization. In fact, here the ADMM [65] is exploited to solve the optimization problem in a distributed fashion. The ADMM is an iterative algorithm by which convex optimization problems can be solved (for a comprehensive description of the ADMM, the reader is redirected to [65, p. 253]).

The average consensus problem over parameters θ_i can be recast as the following convex optimization problem,

$$\overline{\theta} = \underset{x}{\operatorname{argmin}} \sum_{i=1}^{N} (x - \theta_i)^2 \ . \tag{6.1}$$

The ADMM has found several applications to (6.1), such those proposed in [66,67] (that here is called Method A) in the context of parameter estimation.

[1]T. Erseghe is with the Department of Information Engineering, University of Padova, via G. Gradenigo 6/b, 35131 Padova, Italy.

[2]E. Dall'Anese is with the Department of Electrical Engineering, University of Minnesota, 200 Union St., MN 55455, Minneapolis, USA.

In network decoding and cognitive radio sensing, the ADMM has also been proven useful [68, 69] (referred as Method B).

Since in both methods the application of this interesting distributed algorithm produced satisfying results and it resulted to be robust to noise, the present work is intended to perform a deep study of an ADMM-based form of consensus in order to place it in the literature of fast consensus and to derive simple analytical bounds for performance evaluation. A comparison with optimal and suboptimal consensus algorithms [59, 63, 70, 73–75] is made in such sense. Moreover, a generalization of the two most interesting and promising reformulations of ADMM, proposed in [66] and [68] respectively, is done. It is also shown how Method A and B differ in convergence speed and noise resilience. The forthcoming analysis can be considered complementary to that performed in [66–69]. By relaxing the ADMM augmentation constants it is possible to represent the resulting algorithm in matrix form, thus allowing for an optimization study of the convergence and noise resilience properties of the algorithm.

In this chapter, it is shown that the proposed ADMM-based algorithm can be optimized for assuring:

i) a convergence speed comparable to [73] which is the state-of-the-art of fast consensus

ii) a significantly improved robustness to quantization/communication noise with respect to [73];

iii) a faster convergence speed and a stronger resilience to noise than that of the optimal Boyd's solution;

iv) a convergence performance comparable to [61], applicable to any network.

If suboptimal parameters are selected, the performance is just slightly worse with respect to the optimum, once a unique parameter has been carefully chosen. On the other hand, the improved noise resilience makes the ADMM-based consensus algorithm be the state-of-the-art in fast consensus.

6.2 ADMM-based Average Consensus

This section, as well as the following, assumes the absence of noise due either to quantization or to communication errors. Noisy measurements will be dealt later in Section 6.4.

Method A

The reformulation of ADMM in [66] implies the introduction of the dummy variables $\{z_j\}$ to facilitate the derivation of the algorithm. Such variables will be finally taken away when the optimization problem will be solved. By considering all nodes being "bridge" nodes, the formulation of Method A can therefore be stated as follows

$$\begin{aligned} \overline{\theta} = & \underset{\{x_i\},\{z_j\}}{\operatorname{argmin}} \sum_{i=1}^{N} (x_i - \theta_i)^2 \\ & \text{subject to } x_i = z_j, \, \forall i \text{ and } j \in \mathcal{N}_i \end{aligned} \tag{6.2}$$

which is conceptually equivalent to (6.1). Since the network is assumed strongly connected, the set of equality constraints $x_i = z_j$ assures that the estimate values $\{x_i\}_{i=1}^N$ will finally coincide across the whole network.

In Appendix 6.A.1 it is shown that the following notation can be used to express the ADMM Method A of (6.2):

$$a_{i,j} = \frac{c_{i,j}}{\sum_{\ell \in \mathcal{N}_j} c_{\ell,j}} \, , \; b_{i,j} = \frac{c_{i,j}}{1 + \sum_{\ell \in \mathcal{N}_i} c_{i,\ell}} \, , \; d_i = \sum_{j \in \mathcal{N}_i} b_{i,j} \tag{6.3}$$

where $c_{i,j} > 0$ are augmentation constants, and where the condition $\sum_i a_{i,j} = 1$ holds. For $t \geq 1$, the update equations can be expressed as

$$\begin{aligned} x_i(t+1) &= (1 - d_i)\theta_i + d_i \, x_i(t) + \overline{\lambda}_i(t) + 2u_i(t) \\ \overline{\lambda}_i(t+1) &= \overline{\lambda}_i(t) + u_i(t) \end{aligned} \tag{6.4}$$

with initial conditions

$$x_i(1) = (1 - d_i) \, \theta_i \, , \qquad \overline{\lambda}_i(1) = 0 \tag{6.5}$$

and with

$$u_i(t) = \sum_{j \in \mathcal{N}_i} b_{i,j} \Big(\sum_{k \in \mathcal{N}_j} a_{k,j} x_k(t) \Big) - d_i x_i(t) \tag{6.6}$$

which needs to be evaluated in a distributed fashion by means of two message exchanges: the first for the summation in $a_{k,j}$, the second for the summation in $b_{i,j}$.

Method B

The second reformulation of the problem (6.1) descends from [68], which introduces the auxiliary variables $\{z_{i,j}\}$, allowing to recast the optimization

problem as follows

$$\begin{aligned} \overline{\theta} = \underset{\{x_i\},\{z_{i,j}\}}{\operatorname{argmin}} & \sum_{i=1}^{N}(x_i - \theta_i)^2 \\ & \text{subject to } x_i = z_{i,j} \text{ and to } z_{i,j} = z_{j,i},\ \forall i \text{ and } j \in \mathcal{N}_i\ . \end{aligned} \tag{6.7}$$

Similarly to the constraints present in (6.2), since the network is assumed strongly connected, again the set of equality constraints in (6.7) assures that the estimate values $\{x_i\}_{i=1}^N$ will finally coincide across the whole network.

In Appendix 6.A.1 it is shown how the ADMM implementation of (6.7) can be expressed by (6.4) and (6.5), where (6.6) is replaced by

$$\begin{aligned} u_i(t) &= \sum_{j\in\mathcal{N}_i} e_{i,j}x_j(t) - \Big(\sum_{j\in\mathcal{N}_i} e_{i,j}\Big)x_i(t)\ , \\ e_{i,j} &= (1-d_i)\frac{c_{i,j}c_{j,i}}{c_{i,j}+c_{j,i}}\ . \end{aligned} \tag{6.8}$$

As opposite to Method A, the application of (6.7), requires just one message exchange.

Remark

Compared to the literature, in the present work the augmentation constants are relaxed, in the sense that in [68] a unique coefficient is used, i.e., $c_{i,j} = c$, while in [66] the augmentation constant is dependent on the specific node, i.e. $c_{i,j} = c_j$.

6.3 ADMM-based Consensus in Vector Form

In order to facilitate the convergence analysis and the study of noise resilience of ADMM, in this Section a matrix formulation of both Method A and Method B is proposed. Defining $\boldsymbol{I}$ as the $N \times N$ identity matrix, the

following notation is introduced:

$$\begin{aligned}
\boldsymbol{C} &= [c_{i,j}]_{ij} \ , \quad c_{ij} > 0 \\
\tilde{\boldsymbol{C}} &= \begin{cases} [\frac{c_{i,j}c_{j,i}}{c_{i,j}+c_{j,i}}]_{ij} & \text{for } j \in \mathcal{N}_i \\ 0 & \text{for } j \notin \mathcal{N}_i \end{cases} \\
\boldsymbol{\Delta}_1 &= \mathrm{diag}(\boldsymbol{C}'\boldsymbol{1}) \\
\boldsymbol{\Delta}_2 &= \mathrm{diag}(\boldsymbol{C}\boldsymbol{1}) \\
\boldsymbol{A} &= [a_{j,i}]_{ij} = \boldsymbol{\Delta}_1^{-1}\boldsymbol{C}' \\
\boldsymbol{B} &= [b_{i,j}]_{ij} = (\boldsymbol{I} + \boldsymbol{\Delta}_2)^{-1}\boldsymbol{C} \\
\boldsymbol{E} &= [e_{i,j}]_{ij} = (\boldsymbol{I} + \boldsymbol{\Delta}_2)^{-1}\tilde{\boldsymbol{C}} \\
\boldsymbol{D} &= \mathrm{diag}(d_i) = (\boldsymbol{I} + \boldsymbol{\Delta}_2)^{-1}\boldsymbol{\Delta}_2 \ ,
\end{aligned} \tag{6.9}$$

where in the definition of $\boldsymbol{A}$ the interchange of i and j is intended; moreover, it must be noted that matrix $\tilde{\boldsymbol{C}}$ is symmetric by construction. With these definitions,

$$\boldsymbol{U} = \begin{cases} \boldsymbol{B}\boldsymbol{A} - \boldsymbol{D} & \text{, Method A} \\ \boldsymbol{E} - \mathrm{diag}(\boldsymbol{E}\boldsymbol{1}) & \text{, Method B} \end{cases} \tag{6.10}$$

summarizes the multiplications in (6.6) (Method A) and in (6.8) (Method B).

Since $\boldsymbol{A}$ is by construction a row stochastic matrix, i.e., $\boldsymbol{A}\boldsymbol{1} = \boldsymbol{1}$, it is true that $\boldsymbol{D} = \mathrm{diag}(\boldsymbol{B}\boldsymbol{A}\boldsymbol{1})$. This property allows to state that matrix $\boldsymbol{U}$ has the same structure in both methods, and it also satisfies $\boldsymbol{U}\boldsymbol{1} = \boldsymbol{0}$.

In Appendix 6.A.2 it is shown how the linear systems described by (6.4), (6.6), and (6.8) can be represented by the following linear update equation

$$\boldsymbol{x}(t+1) = (\boldsymbol{I} + \boldsymbol{D} + 2\boldsymbol{U})\boldsymbol{x}(t) - (\boldsymbol{D} + \boldsymbol{U})\boldsymbol{x}(t-1) \tag{6.11}$$

or, in a more compact and convenient (for performance analysis purposes) form

$$\begin{aligned}
\boldsymbol{s}(t) = \boldsymbol{M}\,\boldsymbol{s}(t-1) \ , \qquad \boldsymbol{M} &= \begin{bmatrix} 2\boldsymbol{F} - \boldsymbol{G} & \boldsymbol{G} - \boldsymbol{F} \\ \boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \ , \\
\boldsymbol{s}(t) &= \begin{bmatrix} \boldsymbol{x}(t+1) \\ \boldsymbol{x}(t) \end{bmatrix} \ ,
\end{aligned} \tag{6.12}$$

with $\boldsymbol{F} = \boldsymbol{U} + \boldsymbol{I}$ and $\boldsymbol{G} = \boldsymbol{I} - \boldsymbol{D}$. The update rule (6.12) is valid for $t > 0$ with the initial state

$$\boldsymbol{s}(0) = \begin{bmatrix} \boldsymbol{G}\boldsymbol{\theta} \\ \boldsymbol{0} \end{bmatrix} \ . \tag{6.13}$$

The matrix framework of (6.12) corresponds to the standard form of consensus problems (5.2), even if matrix $\boldsymbol{M}$ is not in standard row-stochastic form, as requested from properties C1)-C4) in Theorem 5.1 for reaching consensus.

On the other hand, matrix $\boldsymbol{M}$ has other important properties that allows to reach consensus anyway, in fact (see Appendix 6.A.2):

P1) Defined the eigenvalues of matrix $\boldsymbol{M}$ as λ_i, it is true that $\lambda_1 = 1$ with single multiplicity, while all the other eigenvalues satisfy $|\lambda_i| < 1$.

P2) The left and right eigenvector of matrix $\boldsymbol{M}$ corresponding to eigenvalue 1 are, respectively,

$$\boldsymbol{\ell}_1' = \tfrac{1}{N}\Big[(\mathbf{1} + \boldsymbol{C}\mathbf{1})',\ -(\boldsymbol{C}\mathbf{1})'\Big]\ , \quad \boldsymbol{r}_1 = \mathbf{1}\ ,$$

where $\boldsymbol{\ell}_1'\boldsymbol{r}_1 = 1$ holds.

Such properties lead to some consequences on the limit power behavior of matrix $\boldsymbol{M}$, which is described by (see [79])

$$\boldsymbol{M}^{\infty} = \lim_{t\to\infty} \boldsymbol{M}^t = \boldsymbol{r}_1\boldsymbol{\ell}_1'\ . \tag{6.14}$$

Since it is true that

$$\boldsymbol{\ell}_1'\boldsymbol{s}(0) = \tfrac{1}{N}\mathbf{1}'\boldsymbol{\theta} = \overline{\theta}\ , \tag{6.15}$$

convergence of the series $\{\boldsymbol{s}(t)\}$ to the average consensus is assured, that is

$$\boldsymbol{s}(\infty) = \lim_{t\to\infty} \boldsymbol{s}(t) = \boldsymbol{M}^{\infty}\boldsymbol{s}(0) = \mathbf{1}\overline{\theta}\ .$$

Moreover, since the eigenvalue 1 has single multiplicity (see P1)), the t-th power of $\boldsymbol{M}$ can be splitted (for $t > 0$) in the sum of its limit power (6.14) and a t-dependent matrix, i.e.,

$$\boldsymbol{M}^t\breve{\boldsymbol{M}}^t + \boldsymbol{M}^{\infty}\ , \qquad \breve{\boldsymbol{M}} = \boldsymbol{M} - \boldsymbol{M}^{\infty}\ ,$$

so that the state evolution is described, for $t > 0$, by

$$\boldsymbol{s}(t) = \breve{\boldsymbol{M}}^t\boldsymbol{s}(0) + \mathbf{1}\overline{\theta}\ . \tag{6.16}$$

Matrix $\breve{\boldsymbol{M}}$ results to have the same spectral structure of $\boldsymbol{M}$, with the substantial difference that the eigenvalue $\lambda_1 = 1$ is mapped to eigenvalue $\breve{\lambda}_1 = 0$, so that the whole set of eigenvalues $\breve{\lambda}_i$ of $\breve{\boldsymbol{M}}$ satisfies $|\breve{\lambda}_i| < 1$.

6.3.1 Assumption on the augmentation constants

Properties P1) and P2) are enough to state that matrix $\boldsymbol{M}$ is appropriate for consensus; on the other hand, they do not say anything about the convergence speed and the noise resilience of the update equation (6.12); such properties, in fact, are evinced by the analysis of the Jordan structure of $\boldsymbol{M}$. In order to evaluate these fundamental properties of the algorithm, matrix $\boldsymbol{C}$ is chosen such that $\boldsymbol{D} = d\boldsymbol{I}$, or equivalently $\boldsymbol{G} = \gamma\boldsymbol{I}$, with $\gamma = 1 - d$. This implies the following form for the matrix $\boldsymbol{C}$:

$$\boldsymbol{C} = \epsilon\boldsymbol{S} \;, \qquad \boldsymbol{S}\mathbf{1} = \mathbf{1} \;, \qquad \epsilon > 0 \;, \tag{6.17}$$

which means that $\boldsymbol{C}$ has to be chosen equal to a row-stochastic matrix $\boldsymbol{S}$ (that can be considered its *shape*) times an *amplitude* factor ϵ.

Given that (6.17) holds, it is true that

$$d = \frac{\epsilon}{1+\epsilon} \;, \quad \gamma = \frac{1}{1+\epsilon} \;,$$

therefore matrix $\boldsymbol{F}$ assumes the following expression

$$\begin{aligned} &\boldsymbol{F} = 2d(\boldsymbol{\Phi} - \boldsymbol{I}) + \boldsymbol{I} \;, \\ &\boldsymbol{\Phi} = \begin{cases} \frac{1}{2}\boldsymbol{S}\mathrm{diag}(\boldsymbol{S}'\mathbf{1})^{-1}\boldsymbol{S}' + \frac{1}{2}\boldsymbol{I} & \text{, Method A} \\ \frac{1}{2}\tilde{\boldsymbol{S}} - \frac{1}{2}\mathrm{diag}(\tilde{\boldsymbol{S}}\mathbf{1}) + \boldsymbol{I} & \text{, Method B} \;. \end{cases} \end{aligned} \tag{6.18}$$

It naturally follows that, for $\epsilon = 1$, $\boldsymbol{F} = \boldsymbol{\Phi}$ and $\tilde{\boldsymbol{C}} = \tilde{\boldsymbol{S}}$.

The choice (6.17) brings some consequences in the Jordan structure of matrix $\boldsymbol{M}$. The proofs of the theorems can be found in Appendix 6.A.2.

Theorem 6.1 *Under condition (6.17) matrix $\boldsymbol{F}$ is positive definite, i.e., it is unitarily diagonalizable with positive real valued eigenvalues f_i. The eigenvalues of $\boldsymbol{F}$ satisfy*

$$f_1 = 1 \;, \qquad 0 < \gamma \leq f_i < 1, \; i \neq 1 \;. \tag{6.19}$$

As a consequence, matrix $\boldsymbol{M}$ is unitarily similar to a block diagonal matrix, $\boldsymbol{M} = \boldsymbol{\Sigma}\,\mathrm{diag}(\boldsymbol{B}_i)\,\boldsymbol{\Sigma}^*$ where $\boldsymbol{\Sigma}^* = \boldsymbol{\Sigma}^{-1}$, with 2×2 blocks

$$\boldsymbol{B}_i = \begin{bmatrix} 2f_i - \gamma & -(f_i - \gamma) \\ 1 & 0 \end{bmatrix} \;. \tag{6.20}$$

□

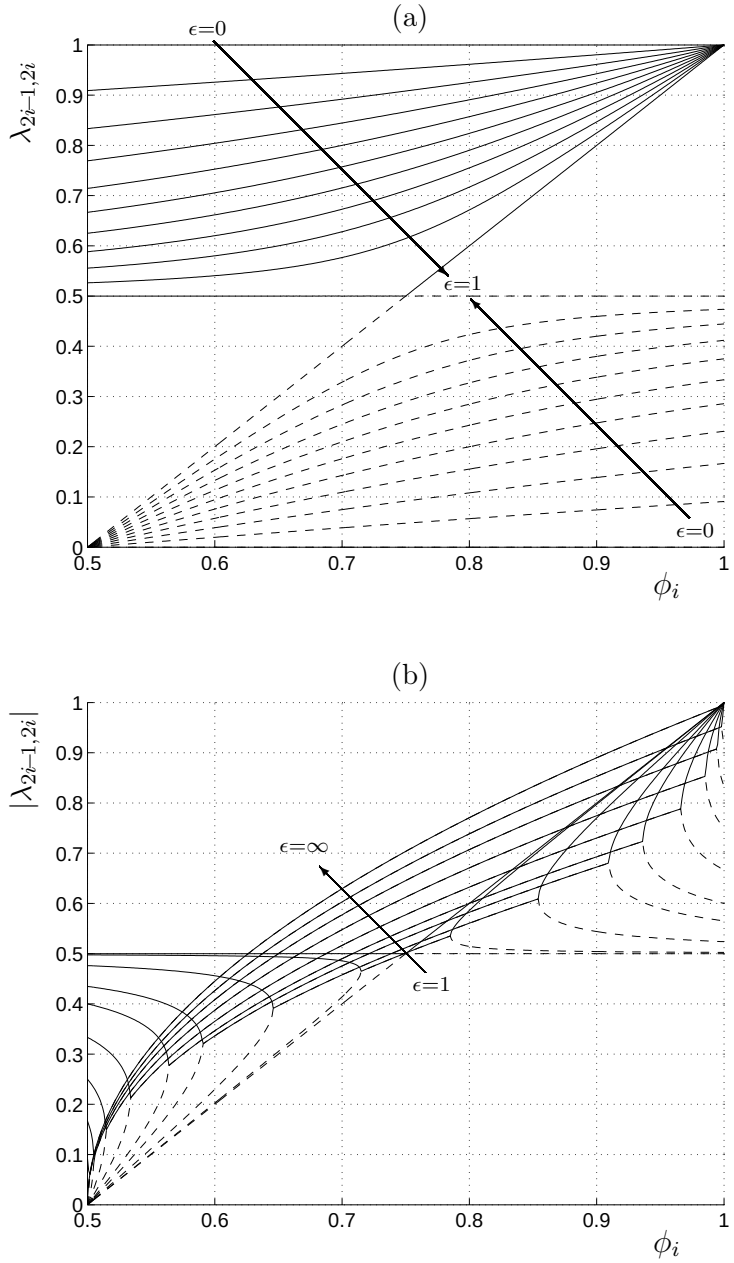

Figure 6.1: Eigenvalues λ_i of matrix $\boldsymbol{M}$ as functions of ϕ_i for $\epsilon \leq 1$ (a) and $\epsilon \geq 1$ (b). Solid lines for odd i and dashed lines for even i.

The eigenvalues of $\boldsymbol{F}$ can be related to those of matrix $\boldsymbol{\Phi}$ and to the amplitude factor ϵ. Sorted eigenvalues $\phi_1 = 1 \geq \phi_2 \geq \ldots \geq \phi_N$ are assumed for $\boldsymbol{\Phi}$; moreover, because of (6.19), it follows that $1 \geq \phi_i \geq \frac{1}{2}$, and therefore that

$$f_i = 2d\phi_i + 1 - 2d \; .$$

From Theorem 6.1, the eigenvalues of matrix $\boldsymbol{M}$ are the solutions of the characteristic polynomials of matrices $\boldsymbol{B}_i$, that is of the equations $\det(\lambda \boldsymbol{I} - \boldsymbol{B}_i) = 0$, therefore they are of the form

$$\lambda_{2i-1,2i} = \frac{(2f_i - \gamma) \pm \sqrt{(2f_i - \gamma)^2 - 4(f_i - \gamma)}}{2} \; . \tag{6.21}$$

In Fig. 6.1 the location of the eigenvalues (6.21) is represented, as a function of ϕ_i. It can be shown that, for $i = 1$, (6.21) provides $\lambda_1 = 1$ and $\lambda_2 = d$. For $i > 1$ different cases need to be considered. More specifically, if $\epsilon < 1$, it follows that $\lambda_{2i-1} \neq \lambda_{2i}$ both being real valued. On the other hand, when $\epsilon \geq 1$, if $f_i = f_{\text{low}}$ or $f_i = f_{\text{high}}$, where

$$f_{\text{low}} = \frac{\gamma + 1 - \sqrt{1 - 2\gamma}}{2} \; , \quad f_{\text{high}} = \frac{\gamma + 1 + \sqrt{1 - 2\gamma}}{2} \; ,$$

then (6.21) provides two real valued coinciding eigenvalues $\lambda_{2i-1} = \lambda_{2i} = f_i - \frac{1}{2}\gamma$. If $f_{\text{low}} < f_i < f_{\text{high}}$, λ_{2i-1} and λ_{2i} constitute a complex conjugate couple, i.e., $\lambda_{2i-1} = \lambda_{2i}^*$. In the last case, for $f_i < f_{\text{low}}$ and $f_i > f_{\text{high}}$, λ_{2i-1} and λ_{2i} are distinct and real valued. In all cases, the following relations hold

$$\begin{aligned} \lambda_{2i-1}\lambda_{2i} &= f_i - \gamma \; , \\ \lambda_{2i-1} + \lambda_{2i} &= 2f_i - \gamma \; , \\ (1 - \lambda_{2i-1})(1 - \lambda_{2i}) &= 1 - f_i \; . \end{aligned} \tag{6.22}$$

The just described considerations imply the following properties on the Jordan structure of $\boldsymbol{M}$:

P3) If $\epsilon < 1$, then $\boldsymbol{M}$ is diagonalizable.

P4) If $\epsilon \geq 1$, then $\boldsymbol{M}$ is diagonalizable if and only if $f_i \neq f_{\text{low}}$ and $f_i \neq f_{\text{high}}$, for all i. If not, the maximum size of the Jordan blocks is $Q = 2$ (and in this case $\boldsymbol{B}_i$ results non diagonalizable).

This aforementioned properties allow a complete understanding of the powers of $\boldsymbol{M}$ and $\breve{\boldsymbol{M}}$, as it is summarized by the following theorem.

Theorem 6.2 *Under condition (6.17), the powers of* $\boldsymbol{M}$ *and* $\breve{\boldsymbol{M}}$ *are unitarily similar to a block diagonal matrix, namely* $\boldsymbol{M}^t = \boldsymbol{\Sigma}\, \text{diag}(\boldsymbol{B}_i^t)\, \boldsymbol{\Sigma}^*$ *and* $\breve{\boldsymbol{M}}^t = \boldsymbol{\Sigma}\, \text{diag}(\breve{\boldsymbol{B}}_i^t)\, \boldsymbol{\Sigma}^*$ *with* $\boldsymbol{\Sigma}^* = \boldsymbol{\Sigma}^{-1}$*, where:*

- *The first block has the form*

$$\begin{aligned} \boldsymbol{B}_1^t &= \frac{1}{1-d}\begin{bmatrix} 1-d^{t+1} & -d(1-d^t) \\ 1-d^t & -d(1-d^{t-1}) \end{bmatrix}, \\ \breve{\boldsymbol{B}}_1^t &= \frac{d^t}{1-d}\begin{bmatrix} -d & d \\ -1 & 1 \end{bmatrix}. \end{aligned} \tag{6.23}$$

- *For $\epsilon < 1$ all remaining blocks have the form*

$$\boldsymbol{B}_1^t = \breve{\boldsymbol{B}}_i^t = \frac{\begin{bmatrix} \lambda_{2i-1}^{t+1} - \lambda_{2i}^{t+1} & -\lambda_{2i-1}\lambda_{2i}(\lambda_{2i-1}^t - \lambda_{2i}^t) \\ \lambda_{2i-1}^t - \lambda_{2i}^t & -\lambda_{2i-1}\lambda_{2i}(\lambda_{2i-1}^{t-1} - \lambda_{2i}^{t-1}) \end{bmatrix}}{\lambda_{2i-1} - \lambda_{2i}} \tag{6.24}$$

  with roots taken from (6.21). Alternatively, (6.24) can be rewritten in the form (sometimes more meaningful)

$$\boldsymbol{B}_i^t = \breve{\boldsymbol{B}}_i^t = \begin{bmatrix} \sum_{k=0}^{t} \lambda_{2i-1}^{t-k}\lambda_{2i}^k & -\sum_{k=0}^{t-1} \lambda_{2i-1}^{t-k}\lambda_{2i}^{k+1} \\ \sum_{k=0}^{t-1} \lambda_{2i-1}^{t-1-k}\lambda_{2i}^k & -\sum_{k=0}^{t-2} \lambda_{2i-1}^{t-1-k}\lambda_{2i}^{k+1} \end{bmatrix} \tag{6.25}$$

  which is valid for $t \geq 2$, but also for $t = 1$ when the bottom right element is set to 0.

- *For $\epsilon \geq 1$ all the blocks satisfying $f_i \neq f_{\text{low}}$ and $f_i \neq f_{\text{high}}$ have the form (6.24) (or (6.25)), while the blocks that satisfy either $f_i = f_{\text{low}}$ or $f_i = f_{\text{high}}$ have the form*

$$\boldsymbol{B}_i^t = \breve{\boldsymbol{B}}_i^t = \begin{bmatrix} (1+t)\lambda_{2i}^t & -t\lambda_{2i}^{t+1} \\ t\lambda_{2i}^{t-1} & -(t-1)\lambda_{2i}^t \end{bmatrix}, \tag{6.26}$$

  where $\lambda_{2i} = f_i - \frac{1}{2}\gamma$.

All blocks $\boldsymbol{B}_i^t$ and $\breve{\boldsymbol{B}}_i^t$ are real valued. □

In this section, it has been shown that with the assumption (6.17) a full spectral analysis of matrix $\boldsymbol{M}$ can be performed. For the rest of the paper, (6.17) it is assumed to hold.

6.4 ADMM Performance Analysis

A performance evaluation of the model (6.12) is now performed, in terms of convergence speed and, introducing the proper noise term, of the noise resilience.

6.4.1 Noiseless communications

In case of noiseless communications, (6.16) holds with no noise terms added. It is well known that the convergence speed of the algorithm is regulated by

$$\rho = \mathrm{sr}(\breve{\boldsymbol{M}}) = \max_i |\breve{\lambda}_i| = \max_{i \neq 1} |\lambda_i| \,, \tag{6.27}$$

which is the *spectral radius*, or the largest eigenvalue modulus of a matrix. It is possible to derive a bound based on ρ, that is generally fairly loose (if identified by exploiting [80, Fact 3], or similar results) but indicative of the convergence speed of the consensus algorithm.

In Appendix 6.A.3, the reader can find the proof of the following theorem:

Theorem 6.3 *When (6.17) applies and in the absence of noise, the convergence of ADMM to its limit value is regulated, for $t > \epsilon$, by the norm-2 bound*

$$\|\boldsymbol{x}(t) - \boldsymbol{x}(\infty)\|_2 \leq \gamma \, \|\boldsymbol{\theta}\|_2 \, t \, \rho^{t-1} \,, \tag{6.28}$$

where $\boldsymbol{x}(\infty) = \mathbf{1}\overline{\theta}$. □

6.4.2 Encompassing for communication noise

Noise coming from wireless communications among nodes can be kept into account in the model (6.12) by adding a noise term $\boldsymbol{n}(t)$, as follows

$$\boldsymbol{s}(t) = \boldsymbol{M}\, \boldsymbol{s}(t-1) + \boldsymbol{n}(t) \tag{6.29}$$

where the vector $\boldsymbol{n}(t)$ is as follows

$$\boldsymbol{n}(t) = \begin{bmatrix} \boldsymbol{w}(t) \\ \mathbf{0} \end{bmatrix} \,, \tag{6.30}$$

with $\boldsymbol{w}(t)$ collecting the N noise samples at each sensor at time t.

Noise $\boldsymbol{n}(t)$ accounts for noise that affects the exchanged readings. Possible sources of noise in the readings $\{x_i(t)\}$ encompass the quantization errors, which are unavoidable for transmitting values in finite precision, and the thermal noise produced by the receivers.

The noise resilience of the algorithm can be studied through the subsequent application of (6.29), i.e.,

$$\boldsymbol{s}(t) = \breve{\boldsymbol{M}}^t \boldsymbol{s}(0) + \sum_{k=1}^{t-1} \breve{\boldsymbol{M}}^k \boldsymbol{n}(t-k) + \boldsymbol{n}(t) + \mathbf{1}\,\overline{\theta} + \mathbf{1} \sum_{k=1}^{t-1} \boldsymbol{\ell}_1' \boldsymbol{n}(t-k) \,, \tag{6.31}$$

which gives a representation of $\boldsymbol{s}(t)$ in terms of $\boldsymbol{s}(0)$ and the noise terms. Because of such corrupting terms, it can happen that consensus on the average value $\overline{\theta}$ cannot be achieved, therefore a study of the noise resilience properties of the algorithm is mandatory.

It can be observed from (6.31) that the last term on the right hand side of the equation is indeed a *random walk*, whose variance increases with time if the condition

$$\boldsymbol{\ell}_1' \, \boldsymbol{n}(k) = \frac{1+\epsilon}{N} \mathbf{1}' \boldsymbol{w}(k) = 0 \tag{6.32}$$

is not verified. To overcome this issue (which in general prevents from reaching average consensus even if the random walk is has zero mean), in [68] the authors only consider noise terms that are orthogonal to $\boldsymbol{\ell}_1$. Such impairment can be overcome also by proper changes in the consensus algorithm, like in [81].

Before proceeding to the performance analysis, it is necessary to design a communication protocol for the algorithm (6.29), so that the noise model can be subsequently characterized. Given that node i has knowledge of its first-hop neighborhood $\mathcal{N}_i$ only, it is assumed, without losing generality, that the only source of noise is the quantization error in the exchanged data. The upcoming analysis can be easily extended if also thermal noise is considered.

Method A

By recalling what done in [66], the description of the communication protocol follows.

1. Node i sends to its first-hop neighbors a message containing the difference

   $$m_{i,1}(t) = 2x_i(t) - x_i(t-1) \, .$$

   As it has already been said, quantization error is present, which is modeled by an additive noise $q_{i,1}(t)$ with zero mean and common variance σ_q^2. Therefore, the noise-corrupted quantity received by all neighbors of node i is

   $$m_{i,1q}(t) = 2x_i(t) - x_i(t-1) + q_{i,1}(t) \, .$$

   By stacking all the $m_{i,1}$'s together in a vector, one obtains

   $$\boldsymbol{m}_1(t) = 2\boldsymbol{x}(t) - \boldsymbol{x}(t-1) \, ,$$

   and the corresponding noisy version is equal to $\boldsymbol{m}_{1q}(t) = \boldsymbol{m}_1(t) + \boldsymbol{q}_1(t)$.

2. The second step requires nodes to transmit an elaboration of $\boldsymbol{m}_1(t)$, which is given by

$$\boldsymbol{m}_2(t) = \boldsymbol{A}\boldsymbol{m}_{1q}(t) \ , \quad \boldsymbol{A} = \mathrm{diag}(\boldsymbol{S}'\boldsymbol{1})^{-1}\boldsymbol{S}' \ ,$$

and its noisy version is equal to $\boldsymbol{m}_{2q}(t) = \boldsymbol{m}_2(t) + \boldsymbol{q}_2(t)$, with $\boldsymbol{q}_2(t)$ having the same statistics of $\boldsymbol{q}_1(t)$ while being statistically independent. The message is then broadcast to the neighbors.

3. The final step involves the update of the estimate values $\boldsymbol{x}(t)$ as follows

$$\begin{aligned} \boldsymbol{x}(t+1) = (1-d)\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{m}_{2q}(t) \\ - d\Big[\boldsymbol{q}_1(t) + \mathrm{diag}(\boldsymbol{S}'\boldsymbol{1})\boldsymbol{q}_2(t)\Big] \ , \end{aligned} \tag{6.33}$$

with $\boldsymbol{B} = d\boldsymbol{S}$, and where the last term has been designed to satisfy (6.32). Moreover, the update step (6.33) can be actually implemented since $q_{i,1}(t)$ and $q_{i,2}(t)$ are known quantities at node i.

The algorithm (6.33) is then equivalent to (6.29) and (6.30), with

$$\boldsymbol{w}(t) = (\boldsymbol{F} - \boldsymbol{I})\boldsymbol{q}_1(t) + d(\boldsymbol{S} - \mathrm{diag}(\boldsymbol{S}'\boldsymbol{1}))\boldsymbol{q}_2(t) \ , \tag{6.34}$$

which satisfies (6.32) since $\boldsymbol{1}'\boldsymbol{w}(t) = 0$, as a consequence of the double stocasticity of matrix $\boldsymbol{F}$ (see Appendix 6.A.2). It is easy to prove that noise $\boldsymbol{w}(t)$ is zero mean, while it has correlation matrix equal to

$$\begin{aligned} \boldsymbol{R}_{\boldsymbol{w}} &= \mathbb{E}\left[\boldsymbol{w}(t)\boldsymbol{w}^*(t)\right] \\ &= \Big[(\boldsymbol{F}-\boldsymbol{I})^2 + d^2(\boldsymbol{S}-\mathrm{diag}(\boldsymbol{S}'\boldsymbol{1}))(\boldsymbol{S}'-\mathrm{diag}(\boldsymbol{S}'\boldsymbol{1}))\Big]\sigma_q^2 \ . \end{aligned}$$

Method B

In Method B just one message exchange is required, therefore the whole communication protocol is simpler than in the previous case.

1. As for Method A.

2. Once the $\{m_{i,1q}\}$ have been received, the update step is implemented as follows

$$\begin{aligned} \boldsymbol{x}(t+1) = (1+d)\boldsymbol{x}(t) - d\boldsymbol{x}(t-1) \\ + (\boldsymbol{F} - \boldsymbol{I})\boldsymbol{m}_{1q}(t) \ . \end{aligned} \tag{6.35}$$

From (6.35) it can be inferred that noise $\boldsymbol{w}(t)$ satisfies (6.32), since

$$\boldsymbol{w}(t) = (\boldsymbol{F} - \boldsymbol{I})\boldsymbol{q}_1(t) \ . \tag{6.36}$$

Also in this case, noise has zero mean but correlation matrix

$$\boldsymbol{R}_{\boldsymbol{w}} = (\boldsymbol{F} - \boldsymbol{I})^2\sigma_q^2 \ .$$

6.4.3 Noise resilience characterization

Noise resilience of (6.29) is analysed through the evaluation of the distance of the state vector $\boldsymbol{x}(t)$ from its average $\mathbf{1}\overline{x}(t)$ at time t, i.e.,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{x}(t) - \mathbf{1}\overline{x}(t) = \boldsymbol{K}\boldsymbol{x}(t) \ , \quad \boldsymbol{K} = \boldsymbol{I} - \tfrac{1}{N}\mathbf{1}\mathbf{1}'$$

where

$$\overline{x}(t) = \tfrac{1}{N}\mathbf{1}'\boldsymbol{x}(t) \ .$$

It is straightforward to note that the two signals, $\dot{\boldsymbol{x}}(t)$ and $\mathbf{1}\overline{x}(t)$ are orthogonal by construction. The aforementioned decomposition has already been applied to similar contexts (see [64, 75]).

Average component

By defining the average quantities

$$\overline{\boldsymbol{s}}(t) = \begin{bmatrix} \overline{x}(t+1) \\ \overline{x}(t) \end{bmatrix}, \ \overline{\boldsymbol{s}}(0) = \begin{bmatrix} \gamma\,\overline{\theta} \\ 0 \end{bmatrix} \ .$$

the following relation holds

$$\overline{\boldsymbol{s}}(t) = \boldsymbol{B}_1\overline{\boldsymbol{s}}(t-1) \ , \quad \boldsymbol{B}_1 = \begin{bmatrix} 1+d & -d \\ 1 & 0 \end{bmatrix} \ . \tag{6.37}$$

It needs to be noticed that the update matrix that regulates the dynamics of (6.37) is actually the block (6.20) with $i = 1$, with eigenvalues $\lambda_1 = 1$ and $\lambda_2 = d$. A recursive application of (6.37) brings to

$$\overline{\boldsymbol{s}}(t) = \boldsymbol{B}_1^t\overline{\boldsymbol{s}}(0)$$

from which an expression for the average signal can be obtained as

$$\overline{x}(t) = (1 - d^t)\overline{\theta} \ . \tag{6.38}$$

The above guarantees the convergence of the average signal $\overline{x}(t)$ to $\overline{\theta}$ with a rate that depends on the parameter d.

Distance from the average

The signal representing the distance from the average has the following dynamics

$$\dot{\boldsymbol{s}}(t) = \dot{\boldsymbol{M}}\dot{\boldsymbol{s}}(t-1) + \dot{\boldsymbol{n}}(t) \ , \quad \dot{\boldsymbol{M}} = \begin{bmatrix} \boldsymbol{K} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K} \end{bmatrix} \boldsymbol{M} \tag{6.39}$$

where

$$\dot{\boldsymbol{s}}(t) = \begin{bmatrix} \dot{\boldsymbol{x}}(t+1) \\ \dot{\boldsymbol{x}}(t) \end{bmatrix}, \ \dot{\boldsymbol{s}}(0) = \begin{bmatrix} \gamma\,\dot{\boldsymbol{\theta}} \\ \mathbf{0} \end{bmatrix}, \ \dot{\boldsymbol{n}}(t) = \begin{bmatrix} \boldsymbol{w}(t) \\ \mathbf{0} \end{bmatrix}$$

with $\dot{\boldsymbol{\theta}} = \boldsymbol{K}\boldsymbol{\theta}$ and with noise $\boldsymbol{w}(t)$ satisfying the property $\boldsymbol{w}(t) = \boldsymbol{K}\boldsymbol{w}(t)$. Matrix $\dot{\boldsymbol{M}}$ is known, and it has a similar eigenstructure to matrix $\boldsymbol{M}$ with the two eigenvalues $\lambda_1 = 1$ and $\lambda_2 = d$ mapped to zero, therefore the Jordan structure is the same ($\dot{\boldsymbol{B}}_i = \boldsymbol{B}_i$ for $i \neq 1$) except for $\dot{\boldsymbol{B}}_1$ which is the 2×2 zero matrix. As a consequence, the powers of $\dot{\boldsymbol{M}}$ are fully described recalling (6.24) and (6.26).

A recursive application of (6.39) gives

$$\dot{\boldsymbol{s}}(t) = \dot{\boldsymbol{M}}^t \dot{\boldsymbol{s}}(0) + \sum_{k=0}^{t-1} \dot{\boldsymbol{M}}^k \dot{\boldsymbol{n}}(t-k) \ .$$

whose convergence speed is regulated by

$$\xi = \mathrm{sr}(\dot{\boldsymbol{M}}) \ , \tag{6.40}$$

which satisfies the condition $\xi \leq \rho = \max(\xi, d) < 1$.

The evolution of the signal expressing the distance from the average can also be written as

$$\dot{\boldsymbol{x}}(t) = \gamma \boldsymbol{Q}\boldsymbol{H}_t\boldsymbol{Q}^*\dot{\boldsymbol{\theta}} + \dot{\boldsymbol{\eta}}(t) \tag{6.41}$$

where $\boldsymbol{Q}$ is the unitary matrix diagonalizing $\boldsymbol{F}$ ($\boldsymbol{F} = \boldsymbol{Q}\mathrm{diag}(f_i)\boldsymbol{Q}^*$) and where $\boldsymbol{H}_t = \mathrm{diag}(h_{i,t})$ contains the bottom left elements of $\dot{\boldsymbol{B}}_i^t$, which are

$$h_{i,t} = \begin{cases} 0 & , i = 1 \\ \frac{\lambda_{2i-1}^t - \lambda_{2i}^t}{\lambda_{2i-1} - \lambda_{2i}} & , i \neq 1, \ \lambda_{2i-1} \neq \lambda_{2i} \\ t\lambda_{2i}^{t-1} & , i \neq 1, \lambda_{2i-1} = \lambda_{2i} \ . \end{cases} \tag{6.42}$$

Moreover,

$$\dot{\boldsymbol{\eta}}(t) = \sum_{k=0}^{t-1} \boldsymbol{Q}\boldsymbol{H}_k\boldsymbol{Q}^*\boldsymbol{w}(t-k) \tag{6.43}$$

is the overall noise term.

Convergence to zero of the elements in (6.42) implies the convergence in mean to zero of $\dot{\boldsymbol{x}}(t)$, in fact

$$\lim_{t\to\infty} \mathbb{E}\left[\dot{\boldsymbol{x}}(t)\right] = \lim_{t\to\infty} \gamma \boldsymbol{Q}\boldsymbol{H}_t\boldsymbol{Q}^*\dot{\boldsymbol{\theta}} = \mathbf{0} \ .$$

On the other hand, the convergence speed properties of (6.41) have to be analysed through the mean squared error, $\dot{\mathrm{MSE}}_t = \mathbb{E}\left[\|\dot{\boldsymbol{x}}(t)\|_2^2\right]$. In Appendix 6.A.3 the following theorem on a bound on $\dot{\mathrm{MSE}}_t$ is proved:

Theorem 6.4 *When (6.17) applies, the MSE*

$$\begin{aligned} \dot{\mathrm{MSE}}_t &= \mathbb{E}\left[\|\dot{\boldsymbol{x}}(t)\|_2^2\right] = m_t^2 + \sigma_t^2 \,, \\ m_t &= \|\mathbb{E}\left[\dot{\boldsymbol{x}}(t)\right]\|_2 = \|\gamma \boldsymbol{Q}\boldsymbol{H}_t\boldsymbol{Q}^*\dot{\boldsymbol{\theta}}\|_2 \,, \\ \sigma_t^2 &= \mathbb{E}\left[\|\dot{\boldsymbol{\eta}}(t)\|_2^2\right] \end{aligned}$$

for $t > \epsilon$ satisfies

$$\begin{aligned} m_t &\leq \gamma\, t\, \xi^{t-1} \|\dot{\boldsymbol{\theta}}\|_2 \\ \sigma_t^2 &\leq K\, \sigma_q^2 \,, \quad K = \sum_{i=2}^{N} K_i\, U_i \end{aligned} \tag{6.44}$$

where

$$K_i = \frac{(1 + f_i - \gamma)}{(1 - f_i)(1 + 3f_i - 2\gamma)(1 - f_i + \gamma)}$$

$$U_i = \begin{cases} (f_i-1)^2 + d^2 \mathrm{sr}((\boldsymbol{S} - \mathrm{diag}(\boldsymbol{S}'\boldsymbol{1}))(\boldsymbol{S}' - \mathrm{diag}(\boldsymbol{S}'\boldsymbol{1}))) & \\ \qquad\qquad , \textit{Method A} & \\ (f_i - 1)^2 + 4d^2 \quad , \textit{Method A}, \boldsymbol{S} = \boldsymbol{S}' & \\ (f_i - 1)^2 + d(\sqrt{f_i - \gamma} - \sqrt{d})^2 \quad , \textit{Method A}, \boldsymbol{S} = \boldsymbol{S}' \succeq \boldsymbol{0} & \\ (f_i - 1)^2 \quad , \textit{Method B} & \end{cases} . \tag{6.45}$$

□

A few remarks on (6.44) are needed. The reader can notice how the convergence speed is regulated by ξ, while the noise level is dependent on K. Moreover, the resulting bound on the noise variance at time t is not dependent on matrix eigenvectors (such as in [66,68]) and, as it will be shown in Section 6.6, it is also very strict, especially when in Method A matrix $\boldsymbol{S}$ has a specific structure.

6.4.4 Initial states choice

The condition (6.15), identifies a vectorial subspace of dimension $2N-1$ from which the initial condition $\boldsymbol{s}(0)$ can be chosen, therefore there is no special constraint to use (6.13), while the noise bound (6.44) is not dependent on the choice of $\boldsymbol{s}(0)$ and the convergence results are not fundamentally affected. As an example, the choice

$$\boldsymbol{s}(0) = \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{\theta} \end{bmatrix} \,, \tag{6.46}$$

is another valid initial state. What makes this specific choice interesting is that the average signal is equal to the average of the initial readings, at each time instant t, i.e.,

$$\overline{x}(t) = \overline{\theta} \, .$$

The convergence bounds (6.28) and (6.44) are just slightly modified, the product γt being substituted by $2t - 1$.

6.5 Optimization Issues

Given the results stated in the previous sections, now the focus is on their analysis in order to optimize the algorithm (6.29) in terms of convergence speed or noise resilience. The parameters to be tuned for optimization purposes are the shape matrix $\boldsymbol{S}$ and the amplitude factor ϵ.

6.5.1 Convergence speed

The aim of this section is to choose $\boldsymbol{S}$ and ϵ in order to maximize the convergence speed. To begin with, $\boldsymbol{S}$ is assumed given and the aim is to choose ϵ such that the convergence speed is maximized. If $\boldsymbol{s}(0)$ is given by (6.46), the convergence speed is described by ξ (6.40), and its minimization can be performed by inspection of Fig. 6.1 (see Appendix 6.A.4); it turns out that

$$\epsilon_{\text{opt}} = \begin{cases} \frac{1}{\sqrt{1-(4\phi_2-3)^2}} & , \phi_2 > \frac{3}{4} \\ 1 & , \phi_2 = \frac{3}{4} \\ \frac{\phi_2 - \frac{1}{2}}{(\phi_2-\phi_N)^2+(\phi_2+\phi_N-1)\sqrt{(\phi_2-\phi_N)^2+4(\phi_2-\frac{1}{2})(1-\phi_N)}} & , \phi_2 < \frac{3}{4} \end{cases} \tag{6.47}$$

for having

$$\xi_{\text{opt}} = \begin{cases} \frac{1+\sqrt{2d_{\text{opt}}-1}}{2} & , \phi_2 > \frac{3}{4} \\ \frac{1}{2} & , \phi_2 = \frac{3}{4} \\ \sqrt{d_{\text{opt}}(2\phi_2 - 1)} & , \phi_2 < \frac{3}{4} \end{cases} , \quad d_{\text{opt}} = \frac{\epsilon_{\text{opt}}}{1 + \epsilon_{\text{opt}}} \, . \tag{6.48}$$

In Fig. 6.2 such results can be observed, with $\phi_N = \frac{1}{2}$, as it is in common scenarios.

If, instead, the initial state is chosen to be equal to (6.13), the convergence speed depends on both d and ξ (see, respectively, (6.38) and (6.44)), therefore $\rho = \max(\xi, d)$ in (6.27) is the key parameter to minimize. In Appendix 6.A.4

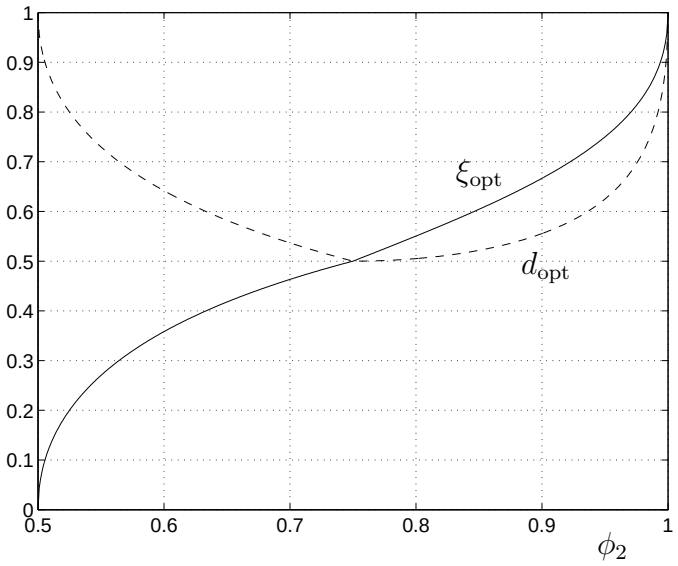

Figure 6.2: Convergence speed maximization choices, with initial state (6.46) and $\phi_N = \frac{1}{2}$.

it is shown that, if $\phi_2 \geq \frac{3}{4}$ results (6.47) and (6.48) keep holding. On the other hand, if $\phi_2 < \frac{3}{4}$, $\epsilon_{\text{opt}} = 1$ is the best choice, with $\xi_{\text{opt}} = \frac{1}{2}$. As a general remark, the case $\phi_2 < \frac{3}{4}$ is a rare case in common networks, therefore generally the convergence properties are regulated by the same equations, (6.47) and (6.48).

If the aim is jointly optimizing $\boldsymbol{S}$ and ϵ and $\boldsymbol{S}$ is a generic $N \times N$ matrix, the optimization problem is still an open issue. On the other hand, the best choice for matrix $\boldsymbol{S}$ already exists if only symmetric matrices are considered, and it is $\boldsymbol{S} = \boldsymbol{S}_{\text{boyd}}$ with $\boldsymbol{S}_{\text{boyd}}$ the optimum Boyd solution (see Section 5.3.2). In this case

$$\boldsymbol{\Phi} = \begin{cases} \frac{1}{2}\boldsymbol{S}_{\text{boyd}}^2 + \frac{1}{2}\boldsymbol{I} & \text{, Method A} \\ \frac{1}{4}\boldsymbol{S}_{\text{boyd}} + \frac{3}{4}\boldsymbol{I} & \text{, Method B} \end{cases}$$

therefore ϕ_2 depends on the second largest eigenvalue modulus of $\boldsymbol{S}_{\text{boyd}}$ denoted as ρ_{boyd}, i.e.,

$$\boldsymbol{\phi}_2 = \begin{cases} \frac{1}{2}(1 + \rho_{\text{boyd}}^2) & \text{, Method A} \\ \frac{1}{4}(3 + \rho_{\text{boyd}}) & \text{, Method B} \end{cases} . \tag{6.49}$$

Fig. 6.3 shows the value of ξ obtained from substituting (6.49) in (6.47) and (6.48), therefore such parameter is there plotted as a function of ρ_{boyd}. For Method A, $\sqrt{\xi}$ is instead plotted since, differently from the other al-

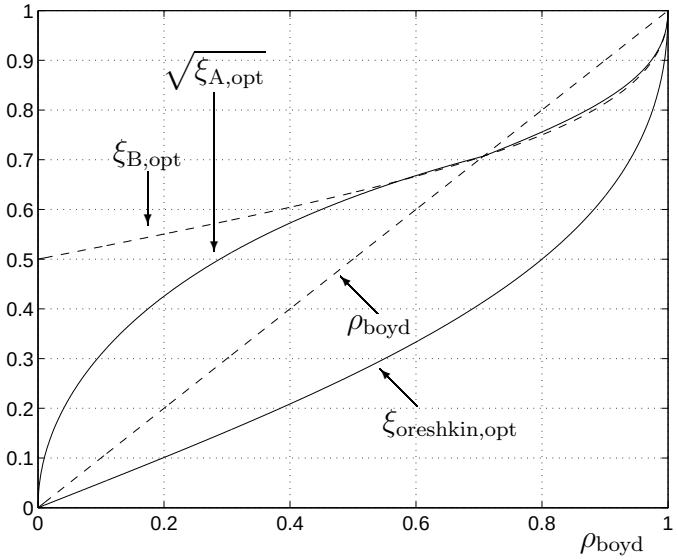

Figure 6.3: Spectral radius of ADMM and of other state-of-the-art consensus algorithm– Boyd's optimal matrix and Oreshkin's method.

gorithms, the data dissemination involves two message exchanges. It is interesting to notice that, for large networks ($\rho_{\mathrm{boyd}} > \frac{1}{\sqrt{2}}$), which is a common scenario, ADMM provides a better convergence speed with respect to the optimum Boyd solution. In Fig. 6.3, the spectral radius for the consensus algorithm proposed by Oreshkin *et al.* [73] is also plotted; it is the current state-of-the-art for fast consensus, and its performance outperforms ADMM's.

To further study the convergence speed, for large networks it results that ρ_{boyd} is very close to 1 so that it can be written as $\rho_{\mathrm{boyd}} = 1 - \Psi(N)$, with small $\Psi(N)$, and, by asymptotic Taylor series expansion it turns out that

$$-\frac{1}{2}\log \xi_{\mathrm{opt,A}} = -\log \xi_{\mathrm{opt,B}} = \frac{\sqrt{2\Psi(N)}}{2} + \mathcal{O}(\Psi(N)) \tag{6.50}$$

so that the speed gap with respect to [73] is only of a factor of 2 (see [73, eq. (15)]). For ρ_{boyd}, it holds that $-\log \rho_{\mathrm{boyd}} = \Psi(N) + \mathcal{O}(\Psi(N)^2)$, therefore the speed gain of ADMM over the Boyd's solution is substantial and comparable to other state-of-the-art fast consensus algorithms [61].

In order to realize which of the two parameters, $\boldsymbol{S}$ or ϵ, is more important for maximization of convergence speed, in Fig. 6.4 the speed measure $-\frac{1}{2}\log \xi_{\mathrm{A}}$ is shown when the underlying network is a mesh grid consisting of $N = L^2$ nodes, in which each node is connected to the four closest neighbors in the grid (right, left, below, above). Discontinuities in the results coming

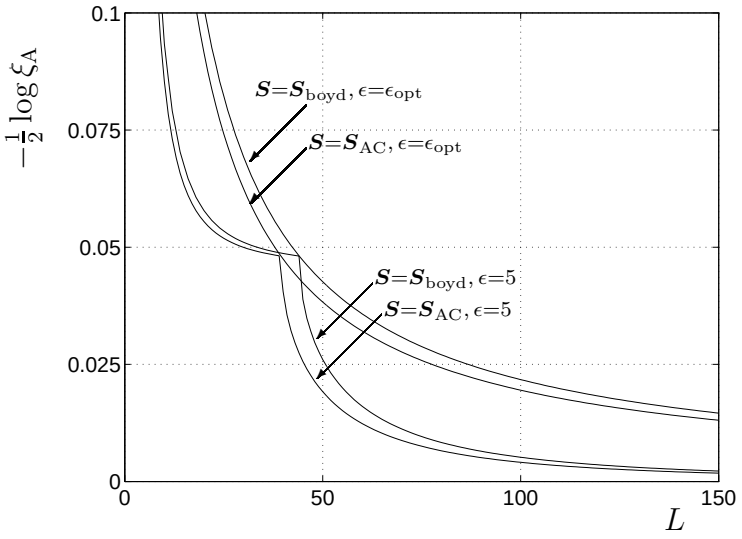

Figure 6.4: Convergence speed $-\frac{1}{2}\log\xi_{\mathrm{A}}$ dependent on ϵ and $\boldsymbol{S}$ when the underlying network graph is a mesh grid where each node has 4 neighbors.

from boundary conditions are overcome by assuming that the grid lies on a torus. It appears evident how ϵ is the most important parameter for optimizing the convergence speed, since just sub-optimal performance is reached if $\boldsymbol{S}$ is chosen to be the averaged consensus matrix $\boldsymbol{S}_{\mathrm{AC}}$ (see Section 5.3.2), with non zero entries equal to $\frac{1}{5}$. Similar results apply to Method A and, in general, to less structured networks.

6.5.2 Resilience to noise

For ADMM-based consensus, the key parameter for describing the noise resilience properties is K, as it is stated in (6.44). From (6.45) it is evident that Method B has a better resilience to noise with respect to Method A, therefore only Method B will be considered in the forthcoming analysis for comparison with the state-of-the-art of standard and fast consensus algorithms. Fig. 6.5 shows the behaviour of K_iU_i as a function of d for different values of ϕ_i, where

$$K_iU_i = \frac{d(1-\phi_i)\left[1+d(2\phi_i-1)\right]}{\left[1-d(2\phi_i-1)\right]\left[1+d(3\phi_i-2)\right]} \quad \text{(Method B)}\ . \tag{6.51}$$

A first observation is that all curves stay below 1, which means that the overall noise is smaller than the quantization noise in all cases, i.e., $\sigma_t^2 \leq (N-1)\sigma_q^2$. Moreover, since K_iU_i tends to 0 as d decreases (i.e., as ϵ goes to

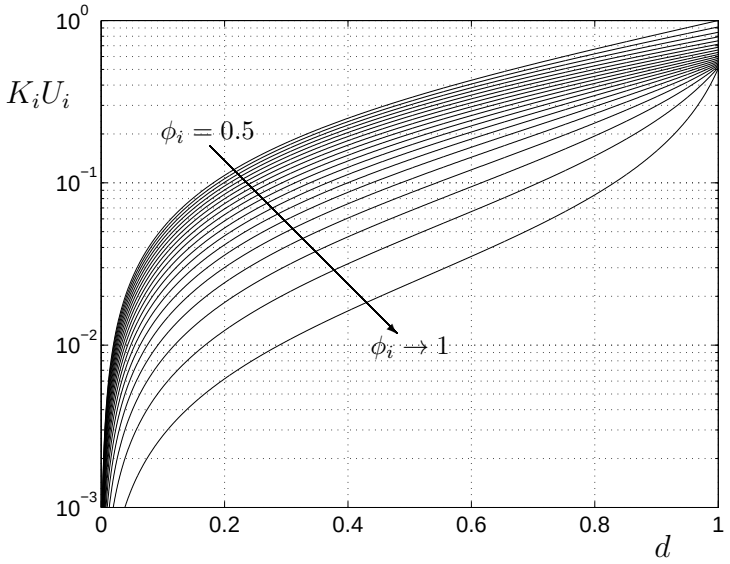

Figure 6.5: Method B: Noise values K_iU_i as functions of d for different values of ϕ_i.

0), then the noise can be controlled at a desired level, by carefully choosing the design parameter ϵ.

Making the comparison with standard consensus, it can be seen that

$$K_iU_i = \frac{1-\rho_i}{1+\rho_i} \qquad \text{(Standard consensus)} \tag{6.52}$$

with $-1 < \rho_i < 1$ being the real valued eigenvalues of the (symmetric) consensus matrix. Getting $K_iU_i < 1$ is assured by having $\rho_i > 0$, which is not guaranteed for all i, even using Boyd's optimal matrix. This means that the expected level of noise in standard consensus is higher than that in ADMM Method B.

A similar conclusion can be drawn for the solution proposed by Oreshkin; in fact, with $(\theta_1, \theta_2, \theta_3) = (-\delta, 0, 1+\delta)$ (see [73, eq. (13)]), in Appendix 6.A.4 by similar arguments it is shown that

$$K_iU_i = \frac{(1+\alpha\delta)(1-\rho_i)}{(1-\alpha\delta)(1+\rho_i)} \quad \text{(Oreshkin } \textit{et al.}\text{, [73])} \tag{6.53}$$

where $-1 < \rho_i < 1$ are again the real valued eigenvalues of the (symmetric) consensus matrix used, and where $\alpha\delta \geq 0$ is carefully designed for the fastest convergence. It is straightforward to see that (6.52) is always less than (6.53), therefore the algorithm proposed by Oreshkin is expected to provide worse

noise performance with respect to standard consensus. This is also supported by the fact that an optimal choice of $\alpha\delta$ asymptotically provides

$$\alpha\delta = 1 - 2\sqrt{2\Psi(N)} + \mathcal{O}(\Psi(N)) \, .$$

6.5.3 Results for highly structured networks

In highly structured networks, performance can be further analysed. In fact, in the following sections line graphs and mesh grids will be characterized in terms of convergence speed and noise resilience, when ADMM-based consensus is used. Some closed form solutions will be provided.

Line graph

In the line graph, every node is connected to two nodes only, the preceding and the following one. Boundary conditions are avoided by connecting the first and the last node in the chain. In this case, matrix $\boldsymbol{S}_{\text{boyd}}$ is circulant, characterized by the first row of the form $[1-2a, a, 0, \ldots, 0, a]$, with

$$a = \begin{cases} \frac{1}{3-\cos(2\pi/N)} & , \text{even } N \\ \frac{1}{2+\cos(\pi/N)-\cos(2\pi/N)} & , \text{odd } N \end{cases}$$

and with eigenvalues having the form $\lambda_{n,\text{boyd}} = 1 + 2a(\cos(2\pi n/N) - 1)$, $n = 0, \ldots, N-1$. It is also $\rho_{\text{boyd}} = 1 + 2a(\cos(2\pi/N) - 1)$.

Mesh grid

The mesh grid is a graph in which each node is connected to the four closest nodes (right, left, below, above). Boundary conditions are avoided through the assumption that the grid lies on a torus. Matrix $\boldsymbol{S}_{\text{boyd}}$ is block circulant and the first block-row is given by $[\boldsymbol{P}, a\boldsymbol{I}, \boldsymbol{0}, \ldots, \boldsymbol{0}, a\boldsymbol{I}]$, with $\boldsymbol{P}$ being a circulant matrix whose first row equal to $[1-4a, a, 0, \ldots, 0, a]$, and

$$a = \begin{cases} \frac{1}{5-\cos(2\pi/L)} & , \text{even } L \\ \frac{1}{3+2\cos(\pi/L)-\cos(2\pi/L)} & , \text{odd } L \, . \end{cases}$$

Eigenvalues are given by the expressions $\lambda_{ij,\text{boyd}} = 1 - 4a + 2a\cos(2\pi i/L) + 2a\cos(2\pi j/L)$, $i, j = 0, \ldots, L-1$, with $\rho_{\text{boyd}} = 1 + 2a(\cos(2\pi/L) - 1)$.

Table 6.1 summarizes the aforementioned results, according to (6.50) for high values of N, while in Fig. 6.6 the noise resilience results are shown, with an evident performance improvement through the use of ADMM with respect to standard and fast consensus approaches.

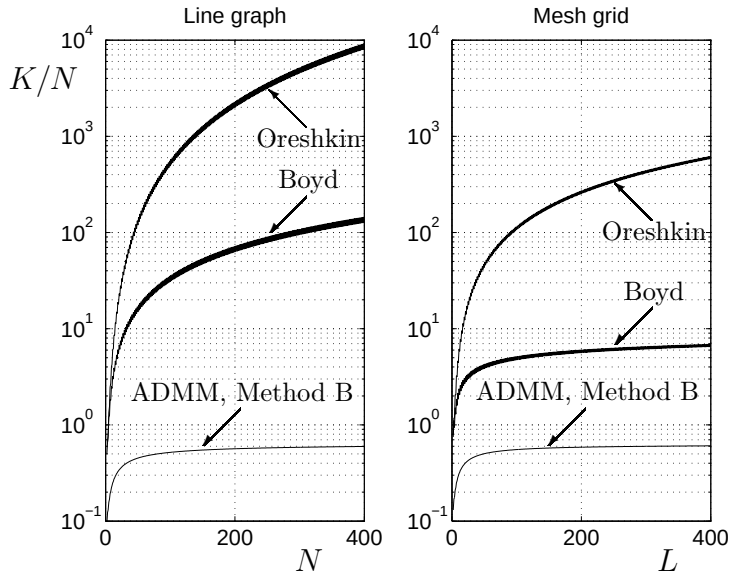

Figure 6.6: Noise resilience in line graph (left) and mesh grid (right).

	Line graph	Mesh grid
$-\log \rho_{\text{boyd}}$	$\sim \frac{2\pi^2}{N^2}$	$\sim \frac{\pi^2}{N}$
$-\frac{1}{2}\log \xi_{\text{A,opt}} \sim -\log \xi_{\text{B,opt}}$	$\sim \frac{\pi}{N}$	$\sim \frac{\pi}{\sqrt{2N}}$
$-\log \xi_{\text{oreshkin,opt}} \sim$	$\sim \frac{2\pi}{N}$	$\sim \frac{\sqrt{2}\pi}{\sqrt{N}}$

Table 6.1: Convergence speed results for line graph and mesh grid.

6.6 Numerical Results

Performance of ADMM are here tested by assuming a network of N nodes whose communication properties are described by a random geometric graph (RGG). More specifically, nodes are disposed in a disc of area 1 m^2 (radius $R = 1/\sqrt{\pi}$), and the coverage radius[3] τ is chosen such that the average number of neighbors is N_{neigh}, to have[4] $\tau = \sqrt{N_{\text{neigh}}/(\pi N)}$.

[3]The coverage area of a node is assumed circular, with radius τ.

[4]Under the assumption of uniform spatial distribution for the nodes, and by neglecting boundary conditions, the probability that any two nodes are in the communication range

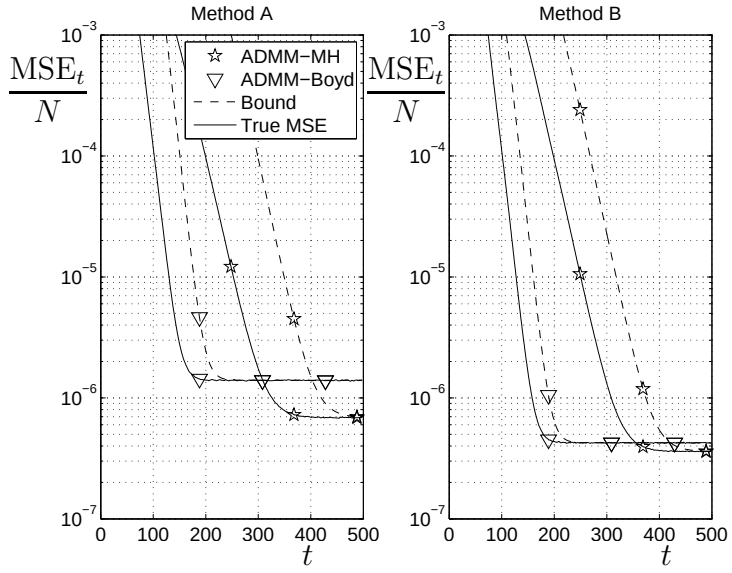

Figure 6.7: MSE with a RGG of $N = 100$ nodes, $N_{\text{neigh}} = 8$, and $\sigma_q^2 = 10^{-6}$.

Comparison between Method A and Method B

In Fig. 6.7 the MSE when using ADMM Method A and B is shown as a function of t, with $N = 100$, $N_{\text{neigh}} = 8$ and $\sigma_q^2 = 10^{-6}$. For statistical confidence, the MSE is averaged over 1000 realizations of the noise.

The used shape matrices are the Boyd's optimal solution $\boldsymbol{S}_{\text{boyd}}$, and the MH matrix $\boldsymbol{S}_{\text{MH}}$ (see Section 5.3.2). The amplitude parameter ϵ is optimally chosen for the maximization of the convergence speed, according to (6.47).

Method B clearly outperforms Method A in terms of noise resilience, confirming the theoretical results of Section 6.4.3. On the other hand, the two methods are pretty much equivalent in terms of convergence speed. Moreover, the choice of a suboptimal shape matrix leads to just limited loss, as it was already been noticed in Fig. 6.4 for mesh grids. In Fig. 6.7 also the MSE bound (6.44) is reported, and it can be observed that such a bound is indeed very strict far large values of t (where it saturates to the noise floor level $K\,\sigma_q^2$) while for smaller t it provides an excellent indication of the convergence speed.

of each other is $P_{\text{conn}} = \pi\tau^2$, which is the ratio between the area covered by a node, and the entire area. The result follows by also noting that the average number of connected nodes is approximately $N_{\text{neigh}} = NP_{\text{conn}}$.

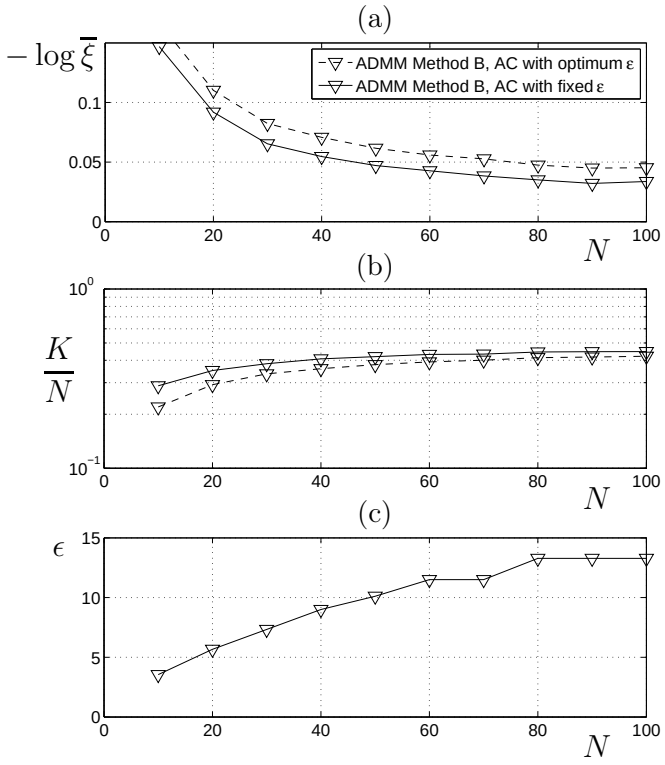

Figure 6.8: ADMM Method B with fixed ϵ in a RGG with $N_{\text{neigh}} = 8$: (a) convergence speed; (b) MSE at steady state; (c) optimal choices of ϵ used.

Suboptimal choice for ϵ

Computing the optimized values of ϵ according to (6.47) can be hard to implement. For such reason, it is interesting to see what is the performance loss if ϵ is chosen differently from ϵ_{opt} in (6.47). In Fig. 6.8, the performance of ADMM in terms of convergence speed and noise level is shown as well as the fixed values of ϵ used, this giving a flavour of the loss induced by this suboptimal, but simpler to implement, choice for the amplitude factor ϵ. The used values for ϵ are those that maximize the average convergence speed, through the minimization of the average spectral radius definition $\overline{\xi} = (\mathbb{E}\left[\xi^{t_{\text{ref}}}\right])^{1/t_{\text{ref}}}$, with $t_{\text{ref}} = 100$. It is clear from Figg. 6.8a and 6.8b that this suboptimal choice for ϵ leads to a small loss both in convergence speed and noise resilience.

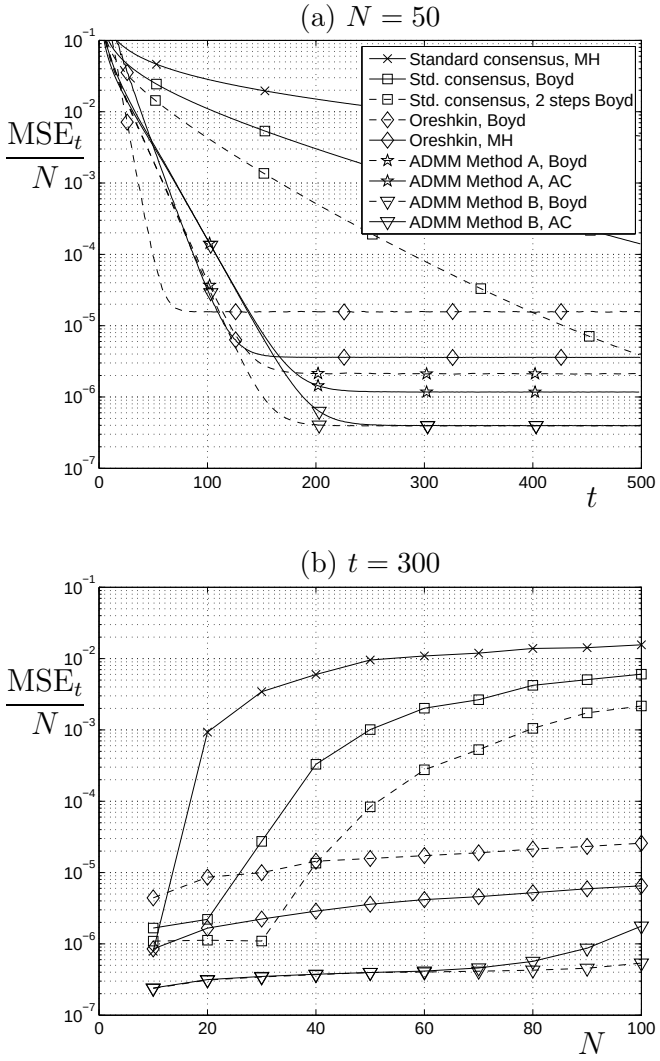

Figure 6.9: MSE with an underlying RGG: (a) as a function of t for $N = 50$ nodes, and (b) as a function of N for $t = 300$. $N_{\text{neigh}} = 4$, $\sigma_q^2 = 10^{-6}$, and ϵ optimized for convergence speed.

Comparison with the state-of-the-art

In order to compare ADMM with the standard and fast consensus algorithms, in Fig. 6.9a the MSE (averaged over 1000 noise realizations) is shown for a RGG with $N = 50$ and $\sigma_q^2 = 10^{-6}$. The used shape matrices are the Boyd's optimum solution $\boldsymbol{S}_{\text{boyd}}$, and the AC matrix $\boldsymbol{S}_{\text{AC}}$, the latter providing better

performance with respect to the MH matrix. The amplitude factor ϵ is again chosen for optimizing the convergence speed according to (6.47).

Fig. 6.9a shows how ADMM allows a dramatic speed improvement with respect to the optimal Boyd's solution. On the other hand, the slight speed loss with respect to [73] is balanced by a consistent improvement in noise resilience when ADMM Method B is used. In Fig. 6.9b, the MSE is shown for $t = 300$ and for different values of N. It can be seen that the relations between the various algorithms hold, independently on the network size.

6.7 Conclusions

This chapter went through the application of ADMM to the problem of average consensus. The approach used for analysing the ADMM properties was the relaxation of the ADMM augmentation constants, thanks to which a closed form matrix description of ADMM has been provided. Convergence speed, noise resilience and simple and tight analytical bounds have been derived. These last turned out to be particularly useful for optimizing ADMM parameters, and for analytically assessing ADMM properties. It has been shown how the ADMM-based consensus has excellent properties in terms of convergence speed, which place it among the best algorithms for fast consensus, getting performance similar to the state-of-the-art [73]. Moreover, the ADMM-based consensus showed the best noise resilience among the standard and the fast consensus algorithms in the literature. All these properties make the ADMM-based consensus an attractive tool for reaching clock synchronization in a WSN quickly and with a very low noise level, as it is better described in Chapter 7.

6.A Appendix

6.A.1 Proof of Methods A and B

Proof of Method A

The augmented Lagrangian function corresponding to (6.2) can be derived through the application of the results in [65, p. 255], therefore getting

$$\begin{aligned} L_{\boldsymbol{c}}(\boldsymbol{x},\boldsymbol{z},\boldsymbol{\lambda}) = \sum_{i=1}^{N}(x_i-\theta_i)^2 + \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i} 2\lambda_{i,j}(x_i-z_j) \\ + \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i} c_{i,j}(x_i-z_j)^2 \end{aligned} \tag{6.54}$$

where vectors $\boldsymbol{x}$, $\boldsymbol{z}$, $\boldsymbol{\lambda}$ and $\boldsymbol{c}$ respectively collect the states x_i,z_j, the Lagrange multipliers $\lambda_{i,j}$, and the augmentation constants $c_{i,j} > 0$. By using the equivalence $\sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i} = \sum_{j=1}^{N}\sum_{i\in\mathcal{N}_j}$, the ADMM update equations for $t \geq 0$ are derived from (6.54), i.e.,

$$\begin{aligned} x_i(t+1) &= \frac{\theta_i + \sum_{j\in\mathcal{N}_i} c_{i,j} z_j(t) - \sum_{j\in\mathcal{N}_i}\lambda_{i,j}(t)}{1+\sum_{j\in\mathcal{N}_i} c_{i,j}} \\ z_j(t+1) &= \frac{\sum_{i\in\mathcal{N}_j} c_{i,j}x_i(t+1) + \sum_{i\in\mathcal{N}_j}\lambda_{i,j}(t)}{\sum_{i\in\mathcal{N}_j} c_{i,j}} \\ \lambda_{i,j}(t+1) &= \lambda_{i,j}(t) + c_{i,j}(x_i(t+1) - z_j(t+1)) \end{aligned} \tag{6.55}$$

where convergence to $x_i(\infty) = \overline{\theta}$ is assured by the fact that $c_{i,j} > 0$, independently on the choice for the initial conditions $\lambda_{i,j}(0)$ and $z_j(0)$. A more convenient form for (6.55) comes from setting null initial conditions, $\lambda_{i,j}(0) = 0$ and $z_j(0) = 0$, and by defining the new parameters

$$\overline{\lambda}_i(t) = -\frac{\sum_{j\in\mathcal{N}_i}\lambda_{i,j}(t-1)}{1+\sum_{j\in\mathcal{N}_i} c_{i,j}}\,, \tag{6.56}$$

so that storing all the Lagrange multipliers is not needed any more. In fact, by first noting that, from the second of (6.55) it is $\sum_{i\in\mathcal{N}_j}\lambda_{i,j}(t) +$

$\sum_{i\in\mathcal{N}_j} c_{i,j}(x_i(t+1) - z_j(t+1)) = 0$ which, in turn, from the third of (6.55), gives

$$\sum_{i\in\mathcal{N}_j} \lambda_{i,j}(t+1) = 0 \ , \qquad t \geq 0 \ , \tag{6.57}$$

by setting null initial conditions for $\lambda_{i,j}(0)$, (6.57) results to hold for $t \geq -1$. Substituting such result into the second of (6.55), it turns out that

$$z_j(t) = \sum_{i\in\mathcal{N}_j} a_{i,j} x_i(t) \ , \qquad t \geq 1 \ . \tag{6.58}$$

The update equation (6.4) is therefore obtained by substituting (6.58) in the first of (6.55) and using definitions (6.3) and (6.6). It is valid for $t \geq 1$. The initial conditions (6.5) come from the first of (6.55) and from (6.56).

Proof of Method B

In Method B, the augmented Lagrangian function is

$$\begin{aligned} L_{\boldsymbol{c}}(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\lambda}) = \sum_{i=1}^{N} (x_i - \theta_i)^2 + \sum_{i=1}^{N} \sum_{j\in\mathcal{N}_i} 2\lambda_{i,j}(x_i - z_{i,j}) + \\ \sum_{i=1}^{N} \sum_{j\in\mathcal{N}_i} c_{i,j}(x_i - z_{i,j})^2 \end{aligned} \tag{6.59}$$

where an additional constraint, $z_{i,j} = z_{j,i}$, must be kept into account. The ADMM update equations for $t \geq 0$ in this case are

$$\begin{aligned} x_i(t+1) &= \frac{\theta_i + \sum_{j\in\mathcal{N}_i} c_{i,j} z_{i,j}(t) - \sum_{j\in\mathcal{N}_i} \lambda_{i,j}(t)}{1 + \sum_{j\in\mathcal{N}_i} c_{i,j}} \\ z_{i,j}(t+1) &= \frac{c_{i,j} x_i(t+1) + c_{j,i} x_j(t+1)}{c_{i,j} + c_{j,i}} + \frac{\lambda_{i,j}(t) + \lambda_{j,i}(t)}{c_{i,j} + c_{j,i}} \\ \lambda_{i,j}(t+1) &= \lambda_{i,j}(t) + c_{i,j}(x_i(t+1) - z_{i,j}(t+1)) \end{aligned} \tag{6.60}$$

with $c_{i,j} > 0$, which again assures convergence to $x_i(\infty) = \overline{\theta}$ for any initial condition $\lambda_{i,j}(0)$ and $z_j(0)$. The second of (6.60) comes from the additional constraint $z_{i,j} = z_{j,i}$ in (6.7). By proceeding analogously to what done in Method A, null initial conditions, $x_i(0) = 0$ and $z_{i,j}(0) = 0$, are considered, thus obtaining $c_{i,j} z_{i,j}(t+1) + c_{j,i} z_{j,i}(t+1) = \lambda_{i,j}(t) + \lambda_{j,i}(t) + c_{i,j} x_i(t+1) + c_{j,i} x_j(t+1)$ from the second of (6.60). By also using the third of (6.60), it

turns out that $\lambda_{i,j}(t+1) + \lambda_{j,i}(t+1) = 0$ for $t \geq 0$. Moreover, by recalling $\lambda_{i,j}(0) = 0$, it also holds that for $t \geq 0$ it is

$$z_{i,j}(t) = \frac{c_{i,j}x_i(t) + c_{j,i}x_j(t)}{c_{i,j} + c_{j,i}} \,. \tag{6.61}$$

The substitution of (6.61) into the first and third of (6.60) finally gets (6.4), with $u_i(t)$ as defined in (6.8) and $\overline{\lambda}_i$ still defined by (6.56).

6.A.2 Proof of Results of 6.3

Proof of (6.11), (6.12) and (6.13)

The update equations (6.4) can be written in matrix form for $t \geq 1$ as

$$\begin{aligned} \boldsymbol{x}(t+1) &= (\boldsymbol{I} - \boldsymbol{D})\boldsymbol{\theta} + (\boldsymbol{D} + 2\boldsymbol{U})\boldsymbol{x}(t) + \boldsymbol{\lambda}(t) \\ \boldsymbol{\lambda}(t+1) &= \boldsymbol{\lambda}(t) + \boldsymbol{U}\boldsymbol{x}(t) \end{aligned} \tag{6.62}$$

with initial conditions $\boldsymbol{x}(1) = (\boldsymbol{I} - \boldsymbol{D})\boldsymbol{\theta}$ and $\boldsymbol{\lambda}(1) = \boldsymbol{0}$. To reformulate (6.62) as an update of $\boldsymbol{x}(t)$ as a function of $\boldsymbol{x}(t)$ itself and of the previous values $\boldsymbol{x}(t-1)$, $\boldsymbol{x}(t)$ is added and subtracted to the first of (6.62), thus getting

$$\begin{aligned} \boldsymbol{x}(t+1) = &\Big[(\boldsymbol{I} - \boldsymbol{D})\boldsymbol{\theta} + (\boldsymbol{D} + 2\boldsymbol{U})\boldsymbol{x}(t) + \boldsymbol{\lambda}(t)\Big] + \boldsymbol{x}(t) \\ &- \Big[(\boldsymbol{I} - \boldsymbol{D})\boldsymbol{\theta} + (\boldsymbol{D} + 2\boldsymbol{U})\boldsymbol{x}(t-1) + \boldsymbol{\lambda}(t-1)\Big] \end{aligned}$$

which gives (6.11) through the use of the second of (6.62), thus proving also (6.12). Being $t \geq 1$, the initial state would be

$$\boldsymbol{s}(1) = \begin{bmatrix} (\boldsymbol{I} + \boldsymbol{D} + 2\boldsymbol{U})(\boldsymbol{I} - \boldsymbol{D})\boldsymbol{\theta} \\ (\boldsymbol{I} - \boldsymbol{D})\boldsymbol{\theta} \end{bmatrix} .$$

On the other hand, the same recursion is found by using the initial state given by (6.13), as $\boldsymbol{s}(1) = \boldsymbol{M}\boldsymbol{s}(0)$.

Proof of P1) and P2)

The eigenvalues of $\boldsymbol{M}$ as described in P1) are assured by the convergence properties of ADMM [65]. P2) is instead proved by the following properties

$$\boldsymbol{F}\boldsymbol{1} = \boldsymbol{1} \,, \quad (\boldsymbol{1} + \boldsymbol{C}\boldsymbol{1})'\boldsymbol{U} = \boldsymbol{0} \,, \quad (\boldsymbol{1} + \boldsymbol{C}\boldsymbol{1})'\boldsymbol{D} = (\boldsymbol{C}\boldsymbol{1})' \,. \tag{6.63}$$

Proof of Theorem 6.1

The properties of matrix $\boldsymbol{F}$ can be derived from standard matrix properties [79, 82]. It must be first noted that, since (6.17) holds, it is $\boldsymbol{\Delta}_2 = \frac{d}{1-d}\boldsymbol{I}$, so that

$$\boldsymbol{F} = \begin{cases} (1-d)\Big[\boldsymbol{C}\boldsymbol{\Delta}_1^{-1}\boldsymbol{C}' + \boldsymbol{I}\Big] & \text{, Method A} \\ (1-d)\Big[\tilde{\boldsymbol{C}} + \boldsymbol{I} + \boldsymbol{\Delta}_2 - \mathrm{diag}(\tilde{\boldsymbol{C}}\boldsymbol{1})\Big] & \text{, Method B} \end{cases} . \tag{6.64}$$

As a consequence, the following properties hold:

a) Elements in $\boldsymbol{F}$ are all non negative, this fact being assured from $c_{i,j} > 0$ in Method A, while in Method B it must be also noted that

$$\mathrm{diag}(\tilde{\boldsymbol{C}}\boldsymbol{1}) = \mathrm{diag}\Big[\sum_j \tfrac{c_{i,j}c_{j,i}}{c_{i,j}+c_{j,i}}\Big] \leq \mathrm{diag}\Big[\sum_j c_{i,j}\Big] = \mathrm{diag}(\boldsymbol{\Delta}_2) \tag{6.65}$$

with $\leq$ applied entrywise.

b) $\boldsymbol{F}\boldsymbol{1} = \boldsymbol{1}$, since $\boldsymbol{U}\boldsymbol{1} = \boldsymbol{0}$ (see (6.63)).

c) $\boldsymbol{F}$ is row-stochastic as a consequence of a) and b).

d) Since the underlying graph is strongly connected, $\boldsymbol{F}$ is *irreducible*.

e) c) and d) assure that $\boldsymbol{F}$ has an eigenvalue $\phi_1 = 1$ with single multiplicity, while the others satisfy $|\phi_i| < 1$, $i \neq 1$ [79, Theorem 10, p. 84].

f) Inspection of (6.64) leads to the conclusion that $\boldsymbol{F}$ is a real symmetric matrix.

g) Inspection of (6.64) and by recalling (6.65) leads to the conclusion that $\boldsymbol{F}$ is positive definite.

h) As a consequence of g), $\boldsymbol{F}$ has real valued and positive eigenvalues.

i) As a consequence of f) and g), $\boldsymbol{F}$ is diagonalizable by a unitary matrix.

j) As a consequence of $\boldsymbol{F} - (1-d)\boldsymbol{I}$ being a positive semidefinite matrix (seen by inspection), the eigenvalues of $\boldsymbol{F}$ satisfy $f_i \geq 1 - d$.

The block diagonalization (6.20) of matrix $\boldsymbol{M}$ is a consequence of the unitary diagonalization of matrix $\boldsymbol{F}$.

Proof of Theorem 6.2

Since $\boldsymbol{M}$ and $\breve{\boldsymbol{M}}$ have the same eigenstructure except for the first eigenvalue ($\lambda_1 = 1$ and $\breve{\lambda}_1 = 0$), Theorem 6.1 assures that $\breve{\boldsymbol{M}} = \boldsymbol{\Sigma}\,\mathrm{diag}(\breve{\boldsymbol{B}}_i)\,\boldsymbol{\Sigma}^*$ where $\breve{\boldsymbol{B}}_1 \neq \boldsymbol{B}_1$ and $\breve{\boldsymbol{B}}_i = \boldsymbol{B}_i$, $i \neq 1$. Blocks $\boldsymbol{B}_i$ can be further analysed:

a) If $\lambda_{2i-1} \neq \lambda_{2i}$, block $\boldsymbol{B}_i$ can be written as $\boldsymbol{B}_i = \boldsymbol{V}_i\boldsymbol{S}_i\boldsymbol{V}_i^{-1}$ where

$$\begin{aligned} \boldsymbol{V}_i &= \begin{bmatrix} \lambda_{2i-1} & \lambda_{2i} \\ 1 & 1 \end{bmatrix}, \quad \boldsymbol{S}_i = \begin{bmatrix} \lambda_{2i-1} & 0 \\ 0 & \lambda_{2i} \end{bmatrix}, \\ \boldsymbol{V}_i^{-1} &= \frac{1}{\lambda_{2i-1} - \lambda_{2i}} \begin{bmatrix} 1 & -\lambda_{2i} \\ -1 & \lambda_{2i-1} \end{bmatrix}. \end{aligned} \tag{6.66}$$

As a consequence,

$$\boldsymbol{B}_1^t = \boldsymbol{V}_1 \begin{bmatrix} 1 & 0 \\ 0 & d^t \end{bmatrix} \boldsymbol{V}_1^{-1}, \quad \boldsymbol{B}_i^t = \boldsymbol{V}_i \begin{bmatrix} \lambda_{2i-1}^t & 0 \\ 0 & \lambda_{2i}^t \end{bmatrix} \boldsymbol{V}_i^{-1}$$

which proves the first of (6.23) and (6.24). Standard properties of the geometric series lead to the expression (6.25). For $\breve{\boldsymbol{B}}_1$ it holds that

$$\breve{\boldsymbol{B}}_1^t = \boldsymbol{V}_1 \begin{bmatrix} 0 & 0 \\ 0 & d^t \end{bmatrix} \boldsymbol{V}_1^{-1}$$

which proves the second of (6.23) by substitution of $\boldsymbol{V}_1$.

b) If $\lambda_{2i-1} = \lambda_{2i}$, block $\boldsymbol{B}_i$ can be expressed as $\boldsymbol{B}_i = \boldsymbol{V}_i \boldsymbol{J}_i \boldsymbol{V}_i^{-1}$, with

$$\begin{aligned} \boldsymbol{V}_i &= \begin{bmatrix} \lambda_{2i} & 1-\lambda_{2i} \\ 1 & -1 \end{bmatrix}, \quad \boldsymbol{J}_i = \begin{bmatrix} \lambda_{2i} & 1 \\ 0 & \lambda_{2i} \end{bmatrix}, \\ \boldsymbol{V}_i^{-1} &= \begin{bmatrix} 1 & 1-\lambda_{2i} \\ 1 & -\lambda_{2i} \end{bmatrix}, \end{aligned} \tag{6.67}$$

where $\boldsymbol{J}_i$ is a Jordan block of size $Q = 2$. It follows that

$$\boldsymbol{B}_i^t = \boldsymbol{V}_i \boldsymbol{J}_i^t \boldsymbol{V}_i^{-1} = \boldsymbol{V}_i \begin{bmatrix} \lambda_{2i}^t & t\lambda_{2i}^{t-1} \\ 0 & \lambda_{2i}^t \end{bmatrix} \boldsymbol{V}_i^{-1}$$

which, by use of (6.67), proves (6.26).

6.A.3 Proof of Theorems of 6.4

Proof of Theorem 6.3

By letting

$$\breve{\boldsymbol{B}}_i^t = \begin{bmatrix} \beta_{1,i}^t & \beta_{2,i}^t \\ \beta_{3,i}^t & \beta_{4,i}^t \end{bmatrix}$$

with $\boldsymbol{Q}$ the unitary matrix which diagonalizes $\boldsymbol{F}$, it holds that

$$\breve{\boldsymbol{M}}^t = \begin{bmatrix} \boldsymbol{Q} & \\ & \boldsymbol{Q} \end{bmatrix} \begin{bmatrix} \operatorname{diag}(\beta_{1,i}^t) & \operatorname{diag}(\beta_{2,i}^t) \\ \operatorname{diag}(\beta_{3,i}^t) & \operatorname{diag}(\beta_{4,i}^t) \end{bmatrix} \begin{bmatrix} \boldsymbol{Q}^* & \\ & \boldsymbol{Q}^* \end{bmatrix}.$$

It follows from (6.16) that $\boldsymbol{x}(t) - \boldsymbol{x}(\infty) = \gamma \boldsymbol{Q} \operatorname{diag}(\beta_{3,i}^t) \boldsymbol{Q}^* \boldsymbol{\theta}$, therefore the application of standard matrix properties (see [82]) lead to $\|\boldsymbol{x}(t) - \boldsymbol{x}(\infty)\|_2 \leq \gamma \|\boldsymbol{\theta}\|_2 \max_i |\beta_{3,i}^t|$. Now, from (6.23) it is $|\beta_{3,i}^t| = d^t/(1-d)$, which is valid for $t > 1$. On the other hand, from (6.26) it holds that $|\beta_{3,i}^t| \leq t\rho^{t-1}$, and the same bound is valid for (6.25). Therefore, for $t > d/(1-d) = \epsilon$ it is $d^t/(1-d) < t\rho^{t-1}$ as $d \leq \rho$, and this proves (6.28).

Proof of Theorem 6.4

Taking the expected value of $\dot{\boldsymbol{x}}(t)$ and using (6.41) leads to

$$\|\gamma \boldsymbol{Q}\boldsymbol{H}_t\boldsymbol{Q}^*\dot{\boldsymbol{\theta}}\|_2 \leq \gamma \|\boldsymbol{H}_t\|_2 \|\dot{\boldsymbol{\theta}}\|_2 \, .$$

By inspection of (6.42) (the reader is also referred to (6.25)), it can be seen that $\|\boldsymbol{H}_t\|_2 \leq t\xi^{t-1}$, which proves the first of (6.44). For the noise signal $\dot{\boldsymbol{\eta}}(t)$, (6.43) and the statistical independence of $\boldsymbol{w}(t)$ lead to

$$\begin{aligned}
\mathbb{E}\left[\|\dot{\boldsymbol{\eta}}(t)\|_2^2\right] &= \sum_{k=0}^{t-1} \mathbb{E}\left[\|\boldsymbol{Q}\boldsymbol{H}_k\boldsymbol{Q}^*\boldsymbol{w}(t-k)\|_2^2\right] \\
&= \sum_{k=0}^{t-1} \operatorname{trace}\left(\boldsymbol{H}_k\boldsymbol{Q}^*\boldsymbol{R}_{\boldsymbol{w}}\boldsymbol{Q}\boldsymbol{H}_k^*\right) \\
&\leq \sum_{k=0}^{+\infty} \operatorname{trace}\left(\boldsymbol{H}_k\boldsymbol{Q}^*\boldsymbol{R}_{\boldsymbol{w}}\boldsymbol{Q}\boldsymbol{H}_k^*\right) = \mathbb{E}\left[\|\dot{\boldsymbol{\eta}}(\infty)\|_2^2\right] .
\end{aligned}$$

By recalling that the $\boldsymbol{H}_k$ are real-valued and diagonal matrices, it holds that

$$\mathbb{E}\left[\|\dot{\boldsymbol{\eta}}(\infty)\|_2^2\right] = \sigma_q^2 \sum_{i=2}^{N} \left(\sum_{k=0}^{\infty} h_{i,k}^2 \right) \frac{[\boldsymbol{Q}^*\boldsymbol{R}_{\boldsymbol{w}}\boldsymbol{Q}]_{i,i}}{\sigma_q^2} \, .$$

Simple algebra manipulation allows to express

$$\sum_{k=0}^{\infty} h_{i,k}^2 = \begin{cases} 0 & , i = 1 \\ \frac{1+\lambda_{2i-1}\lambda_{2i}}{(1-\lambda_{2i-1}^2)(1-\lambda_{2i}^2)(1-\lambda_{2i-1}\lambda_{2i})} & \\ & , i \neq 1, \lambda_{2i-1} \neq \lambda_{2i} \\ \frac{1+\lambda_{2i-1}\lambda_{2i}}{(1-\lambda_{2i-1}\lambda_{2i})^3} & , i \neq 1, \lambda_{2i-1} = \lambda_{2i} \end{cases} .$$

The bound K_i in (6.45) is obtained from $(1-\lambda_{2i-1}\lambda_{2i})^2 > (1-\lambda_{2i-1}^2)(1-\lambda_{2i}^2)$, and from

$$\begin{aligned}
&\frac{1+\lambda_{2i-1}\lambda_{2i}}{(1-\lambda_{2i-1}^2)(1-\lambda_{2i}^2)(1-\lambda_{2i-1}\lambda_{2i})} \\
&\qquad = \frac{1+f_i-\gamma}{(1-f_i)(1+3f_i-2\gamma)(1-f_i+\gamma)}
\end{aligned} \tag{6.68}$$

where (6.22) is also used. The bound U_i for Method B descends on the fact that matrix $\boldsymbol{Q}$ diagonalizes $\boldsymbol{F}$, therefore

$$\frac{[\boldsymbol{Q}^*\boldsymbol{R}_{\boldsymbol{w}}\boldsymbol{Q}]_{i,i}}{\sigma_q^2} = [\boldsymbol{Q}^*(\boldsymbol{F}-\boldsymbol{I})^2\boldsymbol{Q}]_{i,i} = (f_i-1)^2 \, .$$

For Method A, the issue is more involved, since $\boldsymbol{Q}$ does not necessarily diagonalize $(\boldsymbol{S}-\mathrm{diag}(\boldsymbol{S}'\boldsymbol{1}))(\boldsymbol{S}'-\mathrm{diag}(\boldsymbol{S}'\boldsymbol{1}))$. Generally speaking, an upper bound U_i can be derived by exploiting the positive semidefinite inequality

$$\begin{aligned}\mathrm{sr}\Big((\boldsymbol{S}-\mathrm{diag}(\boldsymbol{S}'\boldsymbol{1}))(\boldsymbol{S}'-\mathrm{diag}(\boldsymbol{S}'\boldsymbol{1}))\Big)\boldsymbol{I}&\\ \succeq (\boldsymbol{S}-\mathrm{diag}(\boldsymbol{S}'\boldsymbol{1}))(\boldsymbol{S}'-\mathrm{diag}(\boldsymbol{S}'\boldsymbol{1}))\ .&\end{aligned}$$

If $\boldsymbol{S}$ is symmetric, then (6.18) leads to $\boldsymbol{F}=d(\boldsymbol{S}^2-\boldsymbol{I})+\boldsymbol{I}$, therefore $\boldsymbol{Q}$ results to diagonalize $\boldsymbol{S}$, obtaining

$$\begin{aligned}\frac{[\boldsymbol{Q}^*\boldsymbol{R_w}\boldsymbol{Q}]_{i,i}}{\sigma_q^2} &= (f_i-1)^2+d^2[\boldsymbol{Q}^*(\boldsymbol{S}-\boldsymbol{I})^2\boldsymbol{Q}]_{i,i}\\ &= (f_i-1)^2+d^2(s_i-1)^2\\ &\le (f_i-1)^2+4d^2\end{aligned}$$

where s_i are the real valued eigenvalues of $\boldsymbol{S}$ which satisfy $-1<s_i\le 1$. Finally, if $\boldsymbol{S}$ is also positive semidefinite, then the relation $\boldsymbol{F}=d(\boldsymbol{S}^2-\boldsymbol{I})+\boldsymbol{I}$ leads to

$$f_i=d(s_i^2-1)+1 \quad\Longrightarrow\quad s_i=\sqrt{\frac{f_i-\gamma}{d}}$$

which provides the last bound U_i by substitution.

6.A.4 Proof of Results of 6.5

Sketch of the proof of (6.47) and (6.48)

The cases $\epsilon\le 1$ and $\epsilon>1$ are treated separately, when the initial state is given by (6.46).

- $\epsilon\le 1$: By inspection of Fig. 6.1a, it must be noted that $\lambda_{2i-1}\ge\lambda_{2i}$ for all i, so that just λ_{2i-1} can be considered for the optimization purposes. It results that λ_{2i-1} is a monotone increasing function of ϕ_i, so that ξ is determined by ϕ_2, hence $\xi=\lambda_3$. The minimum is given by the lowest curve with $\epsilon=1$, to have

$$\xi=\lambda_3=\begin{cases}2\phi_2-1 & ,\phi_2\ge\frac{3}{4}\\ \frac{1}{2} & ,\phi_2<\frac{3}{4}\end{cases}\ . \tag{6.69}$$

- $\epsilon>1$: By inspection of Fig. 6.1b, it is $|\lambda_{2i-1}|\ge|\lambda_{2i}|$, so (again) just λ_{2i-1} can be considered. The two cases, $\phi_2\ge\frac{3}{4}$ and $\phi_2<\frac{3}{4}$, need to be treated separately.

- $\phi_2 \geq \frac{3}{4}$: It holds that $|\lambda_3| > |\lambda_{2i-1}|$ for all $i > 2$, hence ξ is (again) determined by ϕ_2 as $\xi = |\lambda_3|$. The minimum here is reached by choosing ϵ in such a way that ϕ_2 corresponds to the point where the eigenvalues locuses separate, that is, when $f_2 = f_{\text{high}}$. This corresponds to the first line of (6.47).
- $\phi_2 < \frac{3}{4}$: In this case, the minimum is guaranteed when ϕ_2 and ϕ_N are equal, that is when $|\lambda_{2N-1}| = |\lambda_3|$. This provides the second line of (6.47).

It must be observed that the values of ξ in (6.48) (derived by choice (6.47)) are the optimum choices, since they are smaller than those of (6.69).

On the other hand, when the initial state is given by (6.13), the eigenvalue d also plays a role. For $\phi_2 \geq \frac{3}{4}$ the solution provided by (6.47) is still optimal because $\xi_{\text{opt}} \geq d$ (see plots in Fig. 6.1b where $d = \lambda_2$ set by $\phi_1 = 1$). For $\phi_2 < \frac{3}{4}$, the two cases $\epsilon > 1$ (for which $d > \frac{1}{2}$, and so $\rho > \frac{1}{2}$) and $\epsilon \leq 1$ (for which $\xi \geq d$, and so $\rho = \xi$) need again to be treated separately. Thus, (6.69) is still the optimum choice, which gives $\rho = \frac{1}{2}$ for $\epsilon = 1$.

Proof of (6.53)

According to [73] the parameters $(\theta_1, \theta_2, \theta_3) = (-\delta, 0, 1+\delta)$ are used, so that the block $\boldsymbol{B}_i$ has the form

$$\boldsymbol{B}_i = \begin{bmatrix} (1+\alpha\delta)\rho_i & -\alpha\delta \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} \lambda_{2i-1}+\lambda_{2i} & -\lambda_{2i-1}\lambda_{2i} \\ 1 & 0 \end{bmatrix} .$$

Using the above in (6.68) leads to

$$K_i = \frac{1+\lambda_{2i-1}\lambda_{2i}}{(1-\lambda_{2i-1}^2)(1-\lambda_{2i}^2)(1-\lambda_{2i-1}\lambda_{2i})} = \frac{1}{(1-\alpha^2\delta^2)(1-\rho_i^2)} .$$

By then assuming the noise model $\boldsymbol{w}(t) = (1+\alpha\delta)(\boldsymbol{W}-\boldsymbol{I})\boldsymbol{q}_1$, equivalent to the one used in (6.36), it turns out that $U_i = (1+\alpha\delta)^2(\rho_i - 1)^2$, from which (6.53) follows.

Chapter 7

An ADMM-Based Clock Synchronization Algorithm

The results presented in this chapter refer to a collaboration of Prof. Vangelista and me with Dr. Emiliano Dall'Anese[1] and Prof. Tomaso Erseghe[2]. They have been published in [83].

7.1 Introduction

Differently from the algorithms presented in Part I of this thesis, in this chapter a clock synchronization algorithm is proposed in which a correction signal is applied at each step of the algorithm. As it has already been said, this makes the clocks be coupled together. The applied control signals result to be linear functions of the observations of each node's neighborhood, so that this algorithm falls down in the class of linear consensus-based clock synchronization. t will be clear next that the full synchronization issue (both in frequencies and counters) is recast as a double consensus problem.

In Chapter 6 an innovative algorithm for reaching average consensus with convergence speed among the best in the literature has been proposed: in the present chapter the ADMM-based consensus is indeed applied to the consensus on clock frequencies through the implementation of the required message passing procedure based on the measurements of frequency displacements. Plain consensus is applied for reaching agreement on the clock counters, but this only affects marginally the results. In fact, it is shown that the innovative

[1]E. Dall'Anese is with the Department of Electrical Engineering, University of Minnesota, 200 Union St., MN 55455, Minneapolis, USA.

[2]T. Erseghe is with the Department of Information Engineering, University of Padova, via G. Gradenigo 6/b, 35131 Padova, Italy.

application of the ADMM-based of Chapter 6 to the problem of consensus on clock frequencies allows to substantially reduce the time needed to get the whole WSN synchronized, with respect to what happens with the application of the standard consensus in both clock frequencies and clock counters. The increased convergence speed has not a bad reflection on the noise resilience, since it is clearly comparable to other consensus-based approaches.

7.2 Problem Statement

The adopted system model is the one described by equations (5.8). As it has already been said, in this chapter the synchronization on clock phases is achieved via the application of a plain consensus algorithm, while agreement on clock periods is obtained via the ADMM-based consensus, so that the control signal $\boldsymbol{u}(t)$ comes from standard consensus, while $\boldsymbol{v}(t)$ is designed according to the guidelines described in Chapter 6. By summarizing what is to be described next, the proposed solution comprises a plain consensus algorithm running in parallel to an ADMM-based consensus algorithm, ensuring fast convergence to commons values for $\boldsymbol{T}(t)$ and $\boldsymbol{\Upsilon}(t)$.

7.3 ADMM-based Consensus on Clock Periods

Consider the goal of computing the average value $\bar{\Upsilon} \triangleq (1/N)\sum_{i=1}^{N} \Upsilon_i(0)$. The application of ADMM to this problem leads to the update equation (6.11), in which $\boldsymbol{x}$ is replaced by $\boldsymbol{\Upsilon}$, thus providing

$$\boldsymbol{\Upsilon}(t+1) = (\boldsymbol{I} + \boldsymbol{D} + 2\boldsymbol{U})\boldsymbol{\Upsilon}(t) - (\boldsymbol{D} + \boldsymbol{U})\boldsymbol{\Upsilon}(t-1) \;, \tag{7.1}$$

with initial conditions $\boldsymbol{\Upsilon}(1) = \boldsymbol{\Upsilon}(0) = [\Upsilon_1(0), \ldots, \Upsilon_N(0)]'$, and where the involved matrices are those defined in (6.9) and (6.11) (Method A). Under the assumption (6.17), it holds that

$$\boldsymbol{C} = \epsilon \boldsymbol{S} \;, \quad \boldsymbol{D} = d\boldsymbol{I} \;, \quad d = \frac{\epsilon}{\epsilon + 1} \;, \tag{7.2}$$

with $\boldsymbol{S}$ the row-stochastic shape matrix and $\epsilon > 0$ the amplitude factor. It is possible to optimize the aforementioned parameters (shape matrix and amplitude factor) in order to maximize the convergence speed, by choosing a well suited matrix $\boldsymbol{S}$ (such as $\boldsymbol{S}_{\text{boyd}}$ or $\boldsymbol{S}_{\text{AC}}$) and, more importantly, an optimized value for the amplitude factor ϵ. In fact, while a careful choice for the amplitude factor is fundamental for reaching higher convergence speed, the choice of the shape matrix is less relevant, as it has already been showed

in Chapter 6. For speeding up the convergence toward $\bar{\Upsilon}$, throughout the remaining of this chapter ϵ will be set equal to ϵ_{opt} according to (6.47).

Using (7.2), a new matrix can be defined, being

$$\boldsymbol{L} = \boldsymbol{S}\boldsymbol{Q} \,, \qquad \boldsymbol{Q} = \, [\text{diag}(\boldsymbol{S}^{'}\mathbf{1})]^{-1}\, \boldsymbol{S}^{'} \,,$$

a symmetric doubly stochastic ($\boldsymbol{L} = \boldsymbol{L}^{'}$, $\boldsymbol{L}\mathbf{1} = \mathbf{1}$) and positive semi-definite matrix, which allows to rewrite (7.1) as

$$\boldsymbol{\Upsilon}(t+1) = \boldsymbol{\Upsilon}(t) + d\Big[(\boldsymbol{L} - \boldsymbol{I})\boldsymbol{\Upsilon}(t) + \boldsymbol{L}(\boldsymbol{\Upsilon}(t) - \boldsymbol{\Upsilon}(t-1))\Big] \,. \tag{7.3}$$

In general, (7.3) provides fast convergence (much faster than the optimal Boyd's solution) and resilience to noise impairments.

From (7.3) it follows that the enforcing control $\boldsymbol{v}(t)$ at time t in (5.8b) can be expressed as

$$\boldsymbol{v}(t) = d\Big[(\boldsymbol{L} - \boldsymbol{I})\boldsymbol{\Upsilon}(t) + \boldsymbol{L}(\boldsymbol{\Upsilon}(t) - \boldsymbol{\Upsilon}(t-1))\Big] \,, \tag{7.4}$$

and it can be seen that it is only dependent upon the clock periods vectors $\boldsymbol{\Upsilon}(t)$ and $\boldsymbol{\Upsilon}(t-1)$.

7.3.1 Algorithm implementation

The implementation of the control signal (7.4) is not an issue, since standard PLL or DLL structures [34] are well suited techniques for correcting the oscillation frequency. On the other hand, the values $\boldsymbol{\Upsilon}(t)$ are not measurable quantities. Fortunately, other related quantities can actually be measured via the application of largely used techniques. In fact, methods that rely on time differences [15,17,84], or efficient frequency estimation techniques (such as the Schmidl-Cox [85]) can be effectively used to measure the (relative) frequency displacements between node pairs, i.e.,

$$\Delta f_{i,j}(t) \triangleq \frac{\Upsilon_j(t)}{\Upsilon_i(t)} - 1 \,, \quad \forall \; j \in \mathcal{N}_i \,, \tag{7.5}$$

sequentially gathered at each node i.

By exploiting the definition of the frequency correlation factor for sensor i at time t, i.e.,

$$\Delta F_i(t) \triangleq \frac{\Upsilon_i(t-1)}{\Upsilon_i(t)} - 1 \,, \tag{7.6}$$

the control signal (7.4) can be rewritten as a function of (7.5) as

$$\Delta F_i(t+1) \simeq d \sum_{j \in \mathcal{N}_i^{(2)}} L_{i,j}\Big[\Delta F_j(t) - \Delta f_{i,j}(t)\Big] \,, \tag{7.7}$$

where $\mathcal{N}_i^{(2)}$ denotes the set of two-hop neighboring nodes of sensor i. Since the frequency offsets of practical systems is quite small (usually it ranges from 10 to 100 ppm [43]), in the derivation of the above the approximations $[1 + \Delta F_i(t+1)]^{-1} \simeq 1 - \Delta F_i(t+1)$ and $\Delta F_i(t+1)\Delta f_{i,j}(t) \simeq 0$ have been used.

The two-hop information flow represented by (7.7) requires a double information exchange at each iteration t. More in the specific, if n is the intermediate node between nodes i and j, it is $n \in \mathcal{N}_i$ and $j \in \mathcal{N}_n$. Moreover, again under the assumption of small frequency offsets, the approximation $\Delta f_{i,j}(t) \simeq \Delta f_{i,n}(t) + \Delta f_{n,j}(t)$ can be made. With this in mind, (7.7) can be expressed as

$$\Delta F_i(t+1) \simeq d \sum_{n \in \mathcal{N}_i} S_{i,n} \Big[\Delta G_n(t) - \Delta f_{i,n}(t) \Big] , \tag{7.8}$$

with

$$\Delta G_n(t) = \sum_{j \in \mathcal{N}_n} Q_{n,j} \Big[\Delta F_j(t) - \Delta f_{n,j}(t) \Big] . \tag{7.9}$$

It must be recalled that the quantities $\Delta f_{i,n}(t)$ and $\Delta f_{n,j}(t)$ are measured at node i and n respectively, while $\Delta F_j(t)$ is transmitted during the first message exchange and $\Delta G_n(t)$ during the second exchange.

7.3.2 Noise resilience

The frequency estimation process as well as the message exchanges involve the presence of corrupting noise. Referring to eq. (7.8) and (7.9), an AWGN model is assured both for the estimation noise affecting $\{\Delta f_{i,n}(t), \Delta f_{n,j}(t)\}$ and for the quantization/communication noise in $\{\Delta F_j(t), \Delta G_n(t)\}$. All sources of noise are assumed mutually independent; this comes after the assumption that $\{\Delta f_{i,n}(t), \Delta f_{n,j}(t)\}$ are estimated independently.

Noise components are incorporated in the $N \times 1$ vector $\boldsymbol{w_v}(t)$, to be added to the noiseless control signal (7.4), to obtain

$$\boldsymbol{v}(t) = d \Big[(\boldsymbol{L} - \boldsymbol{I}) \boldsymbol{\Upsilon}(t) + \boldsymbol{L}(\boldsymbol{\Upsilon}(t) - \boldsymbol{\Upsilon}(t-1)) \Big] + \boldsymbol{w_v}(t) . \tag{7.10}$$

Because of the aforementioned assumptions, $\boldsymbol{w_v}(t)$ is AWGN, zero mean and with covariance matrix given by

$$\begin{aligned} \boldsymbol{R_v} &\triangleq \mathbb{E}\left[\boldsymbol{w_v}(t) \boldsymbol{w_v^*}(t) \right] \\ &\simeq d^2 T_{\text{nom}}^2 \Big[\boldsymbol{S} \boldsymbol{\Sigma}_G \boldsymbol{S}' + ((\boldsymbol{S} \circ \boldsymbol{\Sigma}_f) \boldsymbol{S}') \circ \boldsymbol{I} \\ &\qquad + \boldsymbol{L} \boldsymbol{\Sigma}_F \boldsymbol{L}' + \boldsymbol{S}[((\boldsymbol{Q} \circ \boldsymbol{\Sigma}_f) \boldsymbol{Q}') \circ \boldsymbol{I}] \boldsymbol{S}' \Big] , \end{aligned}$$

where $\circ$ stays for the Hadamard (entry-wise) matrix product, Υ_{nom} denotes the *nominal* clock period, and

$$\begin{aligned}\boldsymbol{\Sigma}_F &\triangleq \text{diag}(\sigma^2_{F_1}, \ldots, \sigma^2_{F_N}) \ , \\ \boldsymbol{\Sigma}_G &\triangleq \text{diag}(\sigma^2_{G_1}, \ldots, \sigma^2_{G_N}) \ , \\ \boldsymbol{\Sigma}_f &\triangleq [\sigma^2_{f_{i,j}}]_{i,j=1,\ldots,N} \ ,\end{aligned}$$

collect estimation and transmission noise variances.

Because of uncorrelated noise, the following result holds from Theorem 6.4:

Theorem 7.1 *The noisy control signal* (7.10) *assures consensus on clock periods* $\boldsymbol{\Upsilon}(t)$*. Letting the deviation from the average value of* $\boldsymbol{\Upsilon}(t)$ *be written as* $\dot{\boldsymbol{\Upsilon}}(t) \triangleq \boldsymbol{K}\boldsymbol{\Upsilon}(t)$*, with* $\boldsymbol{K} = \boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}'$*, upon defining the mean squared error on estimating* $\bar{\Upsilon}$*,*

$$MSE_\Upsilon(t) \triangleq \mathbb{E}\left[\|\dot{\boldsymbol{\Upsilon}}(t)\|_2^2\right] \ , \tag{7.12}$$

the asymptotic convergence can be expressed by the norm-2 result

$$\lim_{t\to\infty} MSE_\Upsilon(t) \leq \sum_{i=2}^{N} \frac{(1+d\ell_i)\sigma^2_{\max}}{d(1-\ell_i)(1-d\ell_i)(2+3d\ell_i-d)} \tag{7.13}$$

where ℓ_i*,* $i = 2, \ldots, N$ *are the eigenvalues of* $\boldsymbol{L}$ *excluding* $\ell_1 = 1$*, and where* $\sigma^2_{\max}$ *is the largest eigenvalue of matrix* $\boldsymbol{K}\boldsymbol{R}_{\boldsymbol{v}}\boldsymbol{K}$*.*

7.4 Consensus on Clock Counters

Although a straightforward application of the ADMM-based consensus to clock phases $\boldsymbol{T}(t)$ is possible since clock counters are directly observable values, it eventually leads to a bias on the control signal $\boldsymbol{u}(t)$ at steady state. Therefore, a plain consensus algorithm is applied for reaching agreement also on such readings, so that the faster convergence property coming from the application of ADMM to clock periods can be exploited anyway in order to get faster full synchronization.

The noise corrupted control signal $\boldsymbol{u}(t)$ can be expressed as

$$\boldsymbol{u}(t) = (\boldsymbol{S} - \boldsymbol{I})\boldsymbol{T}(t) + \boldsymbol{w}_{\boldsymbol{u}}(t) \ , \tag{7.14}$$

where $\boldsymbol{S}$ is any standard consensus matrix, such those described in Section 5.3.2. Noise $\boldsymbol{w}_{\boldsymbol{u}}(t)$ encompasses AWGN noise sources analogously to $\boldsymbol{w}_{\boldsymbol{v}}(t)$

in (7.10). More specifically, $\boldsymbol{w_u}(t)$ comes from the noisy readings $\boldsymbol{T}(t)$ and it is assumed to be zero mean and with covariance matrix

$$\boldsymbol{R_u} = (\boldsymbol{S}-\boldsymbol{I})\boldsymbol{\Sigma}_T(\boldsymbol{S}-\boldsymbol{I})^{'}, \quad \boldsymbol{\Sigma}_T \triangleq \mathrm{diag}(\sigma^2_{T_1},\ldots,\sigma^2_{T_N}) \ ,$$

with $\sigma^2_{T_i}$ being the variances of communication and/or quantization noise affecting the readings $\boldsymbol{T}(t)$.

As for clock periods, the used metric for assessing the algorithm performance is the mean squared error in agreeing on common clock counters, i.e.,

$$\mathrm{MSE}_T(t) \triangleq \mathbb{E}\left[\|\dot{\boldsymbol{c}}(t)\|_2^2\right] \ , \tag{7.15}$$

where

$$\dot{\boldsymbol{T}}(t) \triangleq \boldsymbol{K}\boldsymbol{T}(t) \ . \tag{7.16}$$

7.5 Numerical Results

Performance of the proposed ADMM-based clock synchronization algorithm are now evaluated and compared with existing approaches, in terms of synchronization error $\dot{\boldsymbol{T}}(t)$ and MSE.

The considered setup consists of a WSN of $N = 50$ nodes whose connectivity is represented by a RGG with an average number of neighbors equal to 6. In this scenario (a representation can be observed in Fig. 7.1), nodes are uniformly disposed in a disc area of 1 m^2.

ADMM-based consensus is applied for reaching consensus on clock periods, with averaged consensus shape matrix $\boldsymbol{S}_{\mathrm{AC}}$ (see Section 5.3.2) and amplitude factor ϵ optimized for convergence speed (see (6.47)). Matrix $\boldsymbol{S}$ is also set equal to the averaged consensus matrix (ADMM+AC),. Noise is characterized by the (diagonal) covariance matrices $\boldsymbol{R}_u = \sigma_u^2\boldsymbol{I}$ and $\boldsymbol{R}_v = \sigma_v^2\boldsymbol{I}$, with $\sigma_u^2 = 10^{-2}$ and $\sigma_v^2 = 10^{-6}$ respectively for the control signals $\boldsymbol{u}(t)$ and $\boldsymbol{v}(t)$. The nominal oscillation frequency is set to $f_{\mathrm{nom}} = 1/\Upsilon_{\mathrm{nom}} = 32768$ Hz. The initial conditions $\boldsymbol{\Upsilon}(0)$ are chosen as independent Gaussian random variables, with mean Υ_{nom} and standard deviation 50 ppm. The initial conditions for clock counters $\boldsymbol{T}(0)$ are chosen as independent Gaussian random variables with zero mean and unitary variance. Finally, the time interval between two successive iterations of the algorithm is set to 10 seconds.

7.5.1 Convergence Rate and Noise Resilience

Through an analysis of the synchronization error (7.16) and of the mean squared errors (7.15) and (7.12), in this section the convergence speed and the

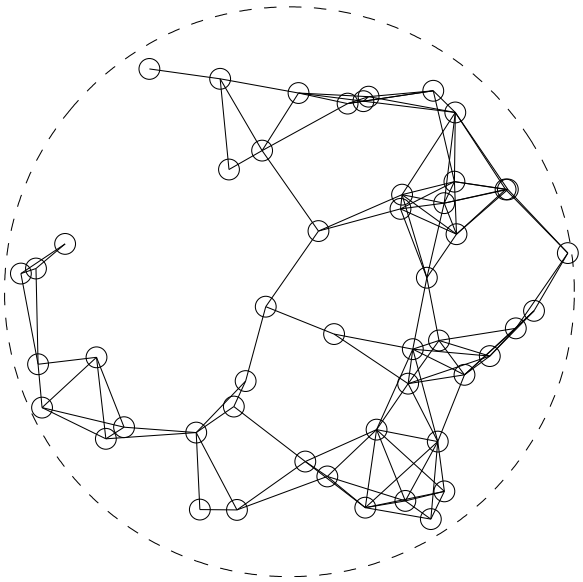

Figure 7.1: Network communication graph.

noise resilience of the proposed ADMM-based algorithm ((5.8a) and (5.8b)) will be evaluated.

The phase errors at five nodes is showed in Fig. 7.2. It can be seen how the synchronization error decreases as time goes by. Moreover, a snapshot around $t = 560$ shows that, from this time on, the network remains synchronized, that is, the absolute value of the synchronization error stays below 1. Note that in the network setup of Fig. 7.1, nodes are pretty sparse, so that the algorithm takes a while to get the network synchronized. On the other hand, it will be shown next that other consensus-based approaches have worse performance.

The characteristic sawtooth shape of Fig. 7.2 is a consequence of the fact that, at the discontinuity points, the correction signals ((7.14) and (7.10)) are applied, and until the next iteration, the frequency skew remains constant; thus, in such intervals the synchronization error increases linearly. However, the slope of the error decreases as time goes by, denoting a progressive improvement in the clock periods agreement across the network.

In Fig. 7.3 the considered metrics are the MSEs (7.15) and (7.12), namely $\mathrm{MSE}_T(t)$ and $\mathrm{MSE}_\Upsilon(t)$ respectively. Consensus on clock phases comes obviously after agreement on periods has been achieved. Moreover, it can be

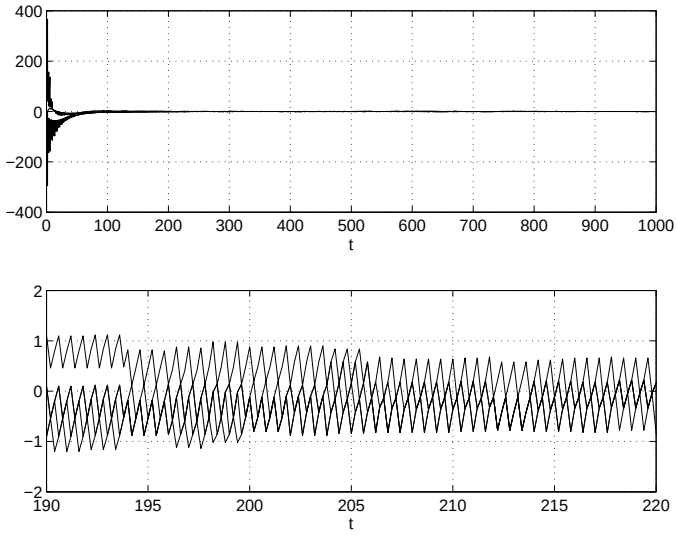

Figure 7.2: Synchronization phase error $T_i(t)$ for five nodes.

noted that the initial disagreement on the clock periods makes the clock phase error to increase. Such trend is rightafter inverted as consensus on clock periods is pursued.

Noise bound (7.13) for the estimation of $\bar{\Upsilon}$ is also showed in Fig. 7.3, confirming its tightness: in fact,the simulative error curve overlaps the bound at steady state.

7.5.2 Performance Comparison with Standard Consensus

The ADMM-based clock synchronization algorithm provides better performance with respect to synchronization algorithms based on plain consensus, even with optimized settings. In fact, Fig. 7.4 shows $\mathrm{MSE}_T(t)$ and $\mathrm{MSE}_\Upsilon(t)$ for ADMM+AC as well as the performance coming from the double application of plain consensus, both to clock phases and to clock periods. The (common) matrix used for the double consensus synchronization algorithm is the averaged consensus matrix $\boldsymbol{S}_{\mathrm{AC}}$ (AC). As a benchmark, (since its high computational cost prevents its practical implementation) the Boyd's optimum solution $\boldsymbol{S}_{\mathrm{boyd}}$ (see Section 5.3.2) is also used (BO). The great performance improvement brought by the ADMM-based consensus is well visible: for clock periods, the convergence speed of the algorithm introduced in Chapter 6 proves to be approximately more than three times the one achieved by plain consensus. This excellent performance also brings relevant bene-

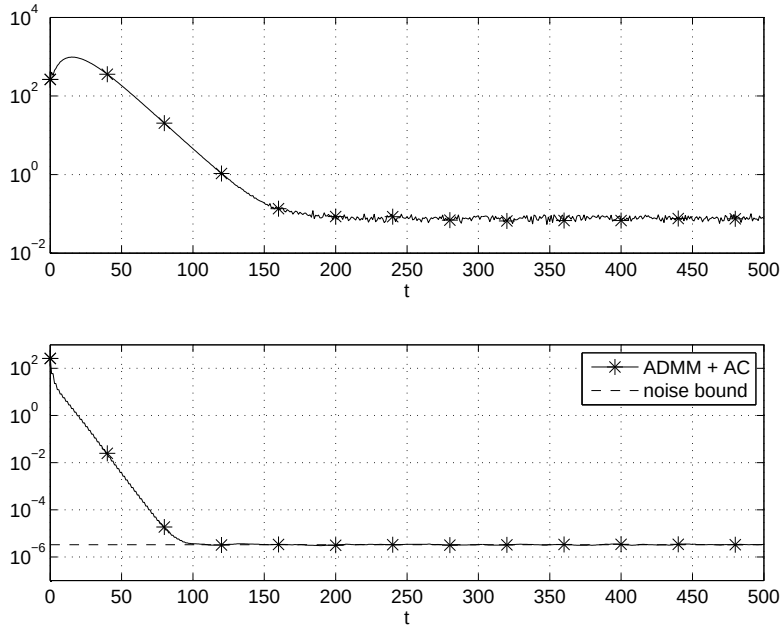

Figure 7.3: MSE for clock phases and clock periods with ADMM+AC.

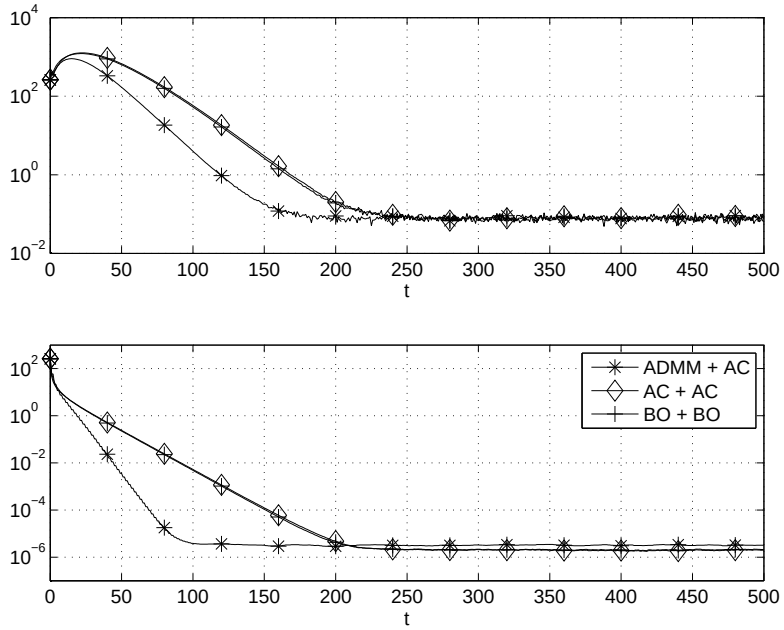

Figure 7.4: MSE of ADMM+AC compared to standard and optimized consensus approaches.

fits for reaching full synchronization: in fact, while for standard consensus more than 340 iterations are needed in order to get the network synchronized (i.e., $\text{MSE}_T < 1$), the same performance is obtained by waiting almost 270 iterations if using the ADMM based consensus.

Finally, as it can be observed in figure, the noise resilience of the proposed algorithm is comparable to that of the standard consensus approaches.

7.6 Conclusions

While in Chapter 6 a new distributed consensus algorithm based on the ADMM has been introduced, here it was shown how its application to the problem of clock synchronization in WSNs leads to better results, compared to the state-of-the-art of consensus-based clock synchronization.

ADMM-based consensus was here applied for achieving agreement on clock periods, while plain consensus was used for clock phases, since a direct application of the method proposed in Chapter 6 would have lead to an undesired and not negligible bias on the correction signal $\boldsymbol{u}(t)$.

Even if ADMM-based consensus could not be straightforwardly applied to clock phases synchronization, its use for agreement on clock periods had extremely positive consequences on the convergence speed of the overall algorithm. In fact, it allowed to reach full synchronization well in advance with respect to existing consensus-based approaches.

Appendix A

Efficient Base Station Selection in Uplink Cellular Networks

The results presented in this chapter refer to a collaboration with Prof. Vangelista and Prof. Stefano Tomasin[1] and have been published in [86, 87]. While falling out of the main topic of this thesis, because of its relevance in my Ph.D. program this work has been included as an appendix chapter. Being self-standing, the present work can also be appreciated by the interested reader independently from the rest of the thesis.

A.1 Introduction

Wireless cellular systems are characterized by the communication between a mobile terminal (MT) and one or more base stations (BSs). Recent advancements in the research community thinking brought the concept of cooperation among BSs to the attention of the other researchers as well as to the industry. In fact, by adopting cooperation among BSs the network throughput can be increased via the implementation of a distributed multiple-input multiple-output (MIMO) system, also known as macro-diversity or multi-cell processing (MCP).

MCP has been first studied from an information theoretic viewpoint and later also in more realistic scenarios with correlated Rayleigh fading between antennas [88] and with possible application to 3GPP long term evolution (LTE) advanced [89]. Moreover, scheduling and power allocation have been studied for MCP in downlink [90].

[1]S. Tomasin is with the Department of Information Engineering, University of Padova, via G. Gradenigo 6/b, 35131 Padova, Italy.

In uplink, MCP encompasses the cooperation among different BSs in order to decode the packet transmitted by a MT in a joint fashion at the serving BS [91, Chapter 2]. In other words, in uplink, cooperating BSs share information on the received packet therefore they need to exploit the backhaul network in a non-negligible manner. The issue of designing the proper characteristics of the backhaul network has been faced in [92, 93], where the capacity of an uplink MCP system with limited backhaul throughput has been investigated. Also, in [94], the goal is to maximize the user throughput while having a limited backhaul capacity.

Cooperation among BSs has been studied in [95] for application to a frequency division duplexing (FDD) LTE network in which both demodulation and decoding are performed at the serving BS, thus requiring a heavy use of the backhaul network. Differently, in [96] just decoding is performed at the serving BS, therefore only soft information bits are forwarded to the serving BS. A study on the asymptotic properties of the spectral efficiency of both the optimum and the linear minimum mean squared error (MMSE) joint multi-cell receivers have been performed in [97] assuming a large number of users per cell, showing that cooperation among receivers allows a significant performance improvement.

As it can be evinced from this discussion, the main issue in the design of an uplink MCP system is the high capacity backhaul network necessary to exchange soft information on the demodulated signals among BSs. The selection of the appropriate cooperating BSs for performing decoding jointly is therefore fundamental from this point of view: on this matter, the authors of [98] select the BSs in order to maximize the network throughput for a given user deployment under backhaul capacity constraints. Handoff for uplink MCP and BS selection are investigated in [99, 100]. In [101], a dynamic greedy algorithm is proposed for selecting the cooperating BSs considering multiple MTs transmitting in a cell and using linear receive beamforming with multi antenna.

Considering error control policies leads to a higher overhead on the backhaul network, since both automatic repeat request (ARQ) and hybrid ARQ (HARQ) encompass retransmissions by the MTs, thus increasing the information sharing and the average decoding delay, while naturally allowing a more reliable communication.

This work considers an uplink MCP system in which BSs can cooperate in order to decode a packet transmitted by a single MT in the considered sector. (H)ARQ error control policies are used for maintaining a threshold level of quality of service (QoS) in terms of maximum allowed outage probability after a given number of retransmissions. The objective is to accordingly select the cooperating BSs such that the average number of cooperating BSs

is minimized (therefore reducing the backhaul network usage). The approach of this work is different from what most of the literature does since, instead of imposing a constraint on the backhaul capacity, a QoS target must be guaranteed to the user with the aim of minimizing the backhaul occupancy. Such an approach especially fits with HARQ error control since the duration of the packet transmission is not fixed: the backhaul network can therefore be statistically exploited by each MT.

Another novel aspect of this work consists on the number of considered policies: in fact, while error control is implemented either with ARQ, HARQ with chase combining or HARQ with incremental redundancy, two levels of cooperation among BSs are also assumed:

joint decoding: cooperating BSs forward soft information to the serving BS which tries to decode the packet

local decoding: cooperating BSs try to decode the packet locally and only exchange one information bit on whether they decoded the packet or not.

In this work, the number of cooperating BSs at each retransmission frame is optimized assuming to have perfect knowledge on the average channel attenuation (path loss) between the MT to each cooperating BSs. Since finding the optimum solution is computationally expensive, an iterative algorithm is introduced which proves to be both efficient and effective. Although making strong assumptions on the channel knowledge, this work provides a way to increase the system efficiency and may provide further insights for possible future implementations of MCP solutions.

A.2 System Model

The considered scenario (see Fig. A.1) involves a MT transmitting a packet to a *serving* BS, which is typically the closest BS to the terminal. It is assumed that the surrounding BSs may help the serving BS through cooperation in a macro diversity approach. Such BSs are called *cooperating* BSs. Just one transmitting MT is considered, since tit is assumed that a pre-selection phase [102] has already taken place. However, a cooperating BS may serve other MTs as well as cooperate with other serving BSs at the same time. As an alternative to the use of other BSs as cooperators, specific devices (relays) may be used, therefore implementing a distributed broadband wireless communication (BWC) system. On the other hand, this is not relevant for the upcoming analysis. In fact, the purpose of this work is to verify if using

cooperating BSs proves useful for reducing the backhaul usage in decoding a packet transmitted by a MT while meeting certain requirements on the QoS.

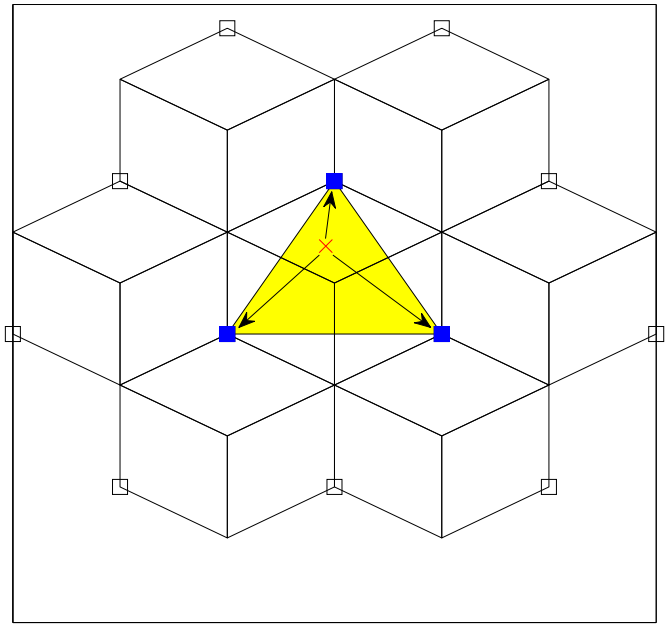

Figure A.1: The MT (red cross) is communicating with the $M = 3$ closest BSs (blu squares) in a cellular scenario.

Unsuccessful decoding is due to channel fading conditions, so that meeting QoS targets requires the adoption of error management policies. Error control is implemented via the use of retransmission policies like *plain* ARQ or *hybrid* ARQ (HARQ), the latter either with *chase combining* (HARQ-CC) or *incremental redundancy* (HARQ-IR). Considering a time division multiplexing system, retransmissions take place in different frames, each organized in two slots: the first slot is used by the MT to transmit data packets at fixed spectral efficiency R (in bit/s/Hz), while in the second slot the serving BS responds either with an acknowledgement (ACK) or with a not ACK (NACK) to the MT, in order to signal successful or failed packet decoding, respectively. The second slot is generally shorter than the first, being ACKs and NACKs shorter packets with respect to data packets transmitted by the MT. A further assumption about ACK and NACK packets is that they are always decoded successfully by the MT: this can be reasonably assumed if, for example, the transmission rate of ACKs and NACKs are low enough. (H)ARQ

frames are retransmitted until either the serving BS decodes the packet or a maximum of N frames is reached, in which case an outage event is declared.

Channel fading is assumed rapidly time-varying and not known at the BSs. The average channel conditions (i.e., path loss in the channels from the MT to the BSs) are instead assumed known at the BSs. Given the channel statistical description, the serving BS selects $x_n \in [0, \ldots, M]$ BSs (including itself) that will cooperate at frame $n \in [1, \ldots, N]$ for decoding the packet transmitted by the MT. The x_n BSs are denoted as *active* at frame n.

Two cooperation methods among BSs are envisioned in this work. On one hand, the active BSs try to decode the packet locally after each frame without exchanging any further soft information: this approach is called local decoding (LD), and the only data exchange between BSs is 1 bit indicating the outcome (either success or failure) of the local decoding process at the end of each frame. Differently from LD, in joint decoding (JD) the cooperating BSs instead forward all the received bits to the serving BS which tries to decode the packet making use of all the available data. In this latter case, the active BSs exchange soft information at the end of each frame. Because of the greater amount of available data, JD is expected to perform better than LD in terms of QoS. This comes at the cost of a larger use of the backhaul network. As a matter of comparison, also the case in which just one (the serving) BS is active is considered and denoted by the acronym NC, which stands for no cooperation.

Let Γ_m be the average signal to noise ratio (SNR) of the channel between the MT and the BSs m. It is assumed that the BSs are sorted according to decreasing values of the SNR, i.e.,

$$\Gamma_1 \geq \Gamma_2 \geq \ldots \geq \Gamma_M \; . \tag{A.1}$$

At each frame n, the serving BS selects the x_n BSs with the higher SNRs to get the best results.

In order to evaluate the performance of the various approaches, an appropriate metric must be introduced. To this end, let $p_n(\boldsymbol{x}, \boldsymbol{\Gamma}, R)$ be the probability of decoding failure at the serving BS *before* frame n, with $n = 1, \ldots, N$; let also be $p_1(\boldsymbol{x}, \boldsymbol{\Gamma}, R) = 1$. In the definition of p_n, the dependence on the number of active BSs at each frame $\boldsymbol{x} \triangleq [x_1, \ldots, x_N]$, on the SNRs of the BSs-MT links $\boldsymbol{\Gamma} \triangleq [\Gamma_1, \ldots, \Gamma_M]$ and on the required spectral efficiency R is made explicit. The fundamental metric adopted in this work is then the following.

Definition A.1 *The average number of active BSs is defined as*

$$f(\boldsymbol{x}) \triangleq \sum_{n=1}^{N} x_n p_n(\boldsymbol{x}, \boldsymbol{\Gamma}, R) \; . \tag{A.2}$$

Since f is a weighted average of x_n, it provides a measure of the backhaul network usage, even if a specific metric will be introduced in the next section for such specific purpose.

The objective here is to elaborate a strategy in order to accordingly choose the vector $\boldsymbol{x}$ so that the average number of active BSs (A.2) is minimized, under a constraint on the maximum outage probability. In other words, the goal is to solve the problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}), \tag{A.3a}$$

subject to

$$p_{N+1}(\boldsymbol{x}, \boldsymbol{\Gamma}, R) \leq \theta, \tag{A.3b}$$

$$x_n \in \{0, 1, \ldots, M\}, \qquad n = 1, 2, \ldots, N. \tag{A.3c}$$

This is essentially different from just requiring to minimize the outage probability. In fact, in that case the optimum solution would be simply setting $x_n = M$ for all n, thus overloading the backhaul network. However, in problem (A.3) the goal is to preserve the backhaul resources while at the same time guaranteeing a target QoS.

In order to attack problem (A.3), it is necessary to derive explicit expressions for the outage probability in each combination of error correction method and decoding policy. This is to be done in the following sections.

A.3 Outage Probability Expressions

The goal of this section is to provide explicit expressions for the outage probability p_n before n frames, in order to enforce the constraint (A.3b). All the cooperation policies among BSs and retransmission methods are considered, while assuming that the channels suffer of flat Rayleigh fading with path loss. To this end, new quantities and notations are introduced. Moreover, in order to evaluate the cell coverage, special focus is given to the case $\Gamma_m = \Gamma$ for all m, this case being the condition in which the MT is at the edge of all surrounding cells. The importance of this case resides on the fact that it provides the highest outage probability, thus proving useful in system design for cell dimensioning and sites position planning.

The considered combinations of decoding policies and error control methods encompass JD, LD and NC, used together with both HARQ-CC and HARQ-IR. Plain ARQ is used only in combination with LD and NC, since JD has no difference with LD, as it will be clear next. Table A.1 summarizes the aforementioned combinations.

In this section, co-user interference is not considered in order to obtain analytical results. The scenario in which only one MT is transmitting inside

	ARQ	HARQ-IR	HARQ-CC
NC	X	X	X
LD	X	X	X
JD		X	X

Table A.1: Combination of decoding policies and retransmission methods.

the sector at that specific time can be achieved by several techniques, ranging from multiple antennas or multiple receivers in each BS and time or frequency multiplexing among cells, at the price of either a higher cost or a lower spectral efficiency. Section A.5 reports simulative results in which both inter-cell and intra-cell interference are taken into account.

In combinations involving the use of JD and HARQ, there is a non negligible backhaul occupancy, since data on the received signals must be forwarded from the cooperating BSs to the serving BS. On the other hand, with LD no soft information forwarding is needed, and the backhaul occupancy (1 bit) can be considered negligible. Given this, in order to better evaluate the backhaul usage of JD, it will be assumed, for simplicity, that the BSs are connected on a ring, each BS being able to communicate directly just to the previous and the following BS in the chain.

Definition A.2 *The average number of backhaul packets exchanged among active BSs B_{P} is defined as the average sum of the lengths of the paths from each active BS to the serving BS,*

$$B_{\mathrm{P}} = \sum_{n=1}^{N} \left[\left\lfloor \frac{x_n - 1}{2} \right\rfloor \left(\left\lfloor \frac{x_n - 1}{2} \right\rfloor + 1 \right) + \chi(x_n) \left\lfloor \frac{x_n + 1}{2} \right\rfloor \right] p_n(\boldsymbol{x}, \boldsymbol{\Gamma}, R) \ , \tag{A.4}$$

with

$$\chi(x_n) = \begin{cases} 1 & x_n \ \textit{even} \\ 0 & x_n \ \textit{odd}. \end{cases}$$

The backhaul occupancy is not the only performance metric adopted in this work. In fact, it is well known that retransmissions due to decoding failures imply a delay on the packet delivery. Therefore, it is necessary to keep into account of such delay by introducing a proper metric.

Definition A.3 *The normalized (with respect to the packet duration) average decoding delay D is the average time required for decoding a packet when transmitted in uplink from a MT, i.e.*

$$D = \sum_{n} p_n(\boldsymbol{x}, \boldsymbol{\Gamma}, R) \ .$$

The delay D depends on the MT position. Its meaning does not describe a time but the number of transmissions needed to achieve correct decoding. In fact, the purpose of introducing such a quantity is to get a measure of the retransmission delay, while being transparent to delays in the retransmission requests from the BS. Moreover, the design of the backhaul network must consider the traffic of the soft information as well as the signalling packets about channel conditions that allow BS selection, in order not to impact on D. The actual performance comparison will be done on a delay-related quantity:

Definition A.4 *The normalized average payload throughput π is the ratio between the required spectral efficiency and the (normalized) average decoding delay, i.e.*

$$\pi = \frac{R}{D} \,. \tag{A.5}$$

In order to simplify the upcoming analysis, the indicator function relative to BS m at frame n

$$y_m(n) \triangleq \begin{cases} 0 & \text{no cooperation} \\ 1 & \text{cooperation} \,, \end{cases} \tag{A.6}$$

is introduced. Since the BSs are ordered with increasing path loss with respect to the MT, $y_m(n)$ are univocally determined from x_n. Moreover, it is useful to introduce the number of frames in which BS m has cooperated until frame n as

$$Y_m(n) \triangleq \sum_{j=1}^{n} y_m(j),$$

for $m = 1, 2, \ldots, M$, and for $n = 1, 2, \ldots, N$.

By inspection of (A.6), it follows that the $y_m(n)$ can be expressed in matrix form, with the row sums being equal to $Y_m(N)$ and the column sums equal to x_n, i.e.,

$$\begin{array}{cccc} \left[\begin{array}{cccc} y_1(1) & y_1(2) & \cdots & y_1(N) \\ y_2(1) & y_2(2) & \cdots & y_2(N) \\ \vdots & \vdots & \ddots & \vdots \\ y_M(1) & y_M(2) & \cdots & y_M(N) \end{array}\right] & \begin{array}{c} \overset{+}{\rightarrow} \\ \overset{+}{\rightarrow} \\ \\ \overset{+}{\rightarrow} \end{array} & \begin{array}{c} Y_1(N) \\ Y_2(N) \\ \cdots \\ Y_M(N) \end{array} \\ \begin{array}{cccc} \downarrow + & \downarrow + & & \downarrow + \\ x_1 & x_2 & \cdots & x_N \end{array} & & & \end{array} \,.$$

In the following, the explicit expressions of the outage probability p_n are derived for all the cases of interest. The NC policy is omitted since it can be obtained from either the JD or the LD case by setting $M = 1$.

A.3.1 Local Decoding with ARQ (LD-ARQ)

Assuming plain ARQ, if the first transmission by the MT fails, the terminal retransmits the same packet to the BSs. At the receiver side, the BSs discard the packet each time a decoding failure happens. This means that there is no information reuse neither from different frames nor from different BSs, so that the only meaningful policy for ARQ is LD. In this case, the expression of the outage probability after n frames, p_{n+1}, is given by the product – over the active BSs and frames – of the probabilities of decoding failure for each BS m during each frame n. The spectral efficiency at BS m for a given channel realization is given by the random variable

$$\Phi_m^{\mathrm{LD-A}}(\boldsymbol{x}, \Gamma_m) \triangleq \log_2\left(1 + y_m(j)\Gamma_m |h_m(j)|^2\right),$$

where $h_m(j)$ are i.i.d. normalized complex Gaussian random variables modeling the Rayleigh fading in the channel from the MT to BS m at frame j. Path loss is taken into account by Γ_m. The outage probability is therefore given by

$$\begin{aligned} p_{n+1}^{\mathrm{LD-A}}(\boldsymbol{x}, \boldsymbol{\Gamma}, R) &= \prod_{j=1}^{n} \prod_{m=1}^{x_j} \mathbb{P}[\Phi_m^{\mathrm{LD-A}}(\boldsymbol{x}, \Gamma_m) < R] = \\ &= \prod_{j=1}^{n} \prod_{m=1}^{x_j} a_m, \end{aligned} \tag{A.7}$$

with

$$\begin{aligned} a_m &\triangleq \mathbb{P}[\Gamma_m |h_m(j)|^2 < 2^R - 1] = \\ &= 1 - e^{-\frac{2^R - 1}{\Gamma_m}}. \end{aligned}$$

Given the ordering (A.1) assumed for the BSs, it holds that

$$0 \leq a_1 \leq \ldots \leq a_M \leq 1\,. \tag{A.8}$$

A.3.2 Local Decoding with HARQ-CC (LD-HARQ-CC)

In LD-HARQ-CC the outage probability is the product of the partial outage probabilities at each BS. The spectral efficiency at BS m is the random variable

$$\Phi_m^{\mathrm{LD-CC}}(\boldsymbol{x}, \Gamma_m) \triangleq \log_2\left(1 + \sum_{j=1}^{n} y_m(j)\Gamma_m |h_m(j)|^2\right),$$

while the probability of being in outage after n frames is

$$p_{n+1}^{\mathrm{LD-CC}}(\boldsymbol{x}, \boldsymbol{\Gamma}, R) = \prod_{m=1}^{M} \mathbb{P}\left[\Phi_m^{\mathrm{LD-CC}}(\boldsymbol{x}, \Gamma_m) < R\right]. \tag{A.9}$$

Probability (A.9) can be rearranged as

$$\prod_{m=1}^{M} \mathbb{P}\left[\sum_{j=1}^{n} y_m(j)\Gamma_m |h_m(j)|^2 < 2^R - 1\right] = \prod_{m=1}^{M} F_C\left(Y_m(n), \frac{2^R-1}{\Gamma_m}\right)$$

where $F_C(k, z)$ is the cumulative distribution function (cdf) of the Gamma random variable $\sum_{\ell=1}^{k} |X_\ell|^2$ (with X_ℓ i.i.d. normalized complex Gaussian random variables) with scale 1 and shape k, being defined as

$$F_C(k, z) = \frac{1}{(k-1)!}\gamma(k, z), \qquad z \geq 0\,,$$

where

$$\gamma(k, z) = \sum_{l=0}^{\infty} \frac{(-1)^l}{l!} \frac{z^{k+l}}{k+l}, \qquad z \geq 0\,.$$

For the scenario of multi-relay networks, similar results have been obtained in [103].

A.3.3 Joint Decoding with HARQ-CC (JD-HARQ-CC)

In JD-HARQ-CC the spectral efficiency is given by

$$\Phi^{\mathrm{JD-CC}}(\boldsymbol{x}, \boldsymbol{\Gamma}) \triangleq \log_2\left(1 + \sum_{j=1}^{n}\sum_{m=1}^{x_j} \Gamma_m |h_m(j)|^2\right)$$

with outage probability after n frames

$$p_{n+1}^{\mathrm{JD-CC}}(\boldsymbol{x}, \boldsymbol{\Gamma}, R) = \mathbb{P}\left[\Phi^{\mathrm{JD-CC}}(\boldsymbol{x}, \boldsymbol{\Gamma}) < R\right]. \tag{A.10}$$

Even if in Appendix A.A.2 a closed-form expression of probability (A.10) is derived, it results computationally intractable. To simplify the treatment, the worst case scenario is assumed in the absence of interference, that is the MT is assumed at the edge of all surrounding sectors, and $\Gamma_m = \Gamma$ for $m = 1, 2, \ldots, M$. In this case, probability (A.10) is given by the cdf of a Gamma random variable with scale 1 and shape k_n, where $k_n \triangleq \sum_{j=1}^{n} x_j$, i.e.,

$$\begin{aligned} p_{n+1}^{\mathrm{JD-CC}}(\boldsymbol{x}, \boldsymbol{\Gamma}, R) &= \mathbb{P}[\Gamma \|\mathbf{h}_n\|^2 < 2^R - 1] \\ &= F_C\left(k_n, \tfrac{2^R-1}{\Gamma}\right). \end{aligned}$$

In Section A.5, the system performance is studied with MTs dropped randomly in the sector, thus extending the special case $\Gamma_m = \Gamma$ through numerical simulation.

A.3.4 Local Decoding with HARQ-IR (LD-HARQ-IR)

For LD-HARQ-IR, information is accumulated at each BS, frame after frame. The spectral efficiency therefore results to be

$$\Phi^{\mathrm{LD-IR}}(\boldsymbol{x}, \Gamma_m) \triangleq \sum_{j=1}^{n} y_m(j) \log_2 \left(1 + \Gamma_m |h_m(j)|^2\right)$$

with the outage probability after n frames given by

$$p_{n+1}^{\mathrm{LD-IR}}(\boldsymbol{x}, \boldsymbol{\Gamma}, R) = \prod_{m=1}^{M} \mathbb{P}\left[\Phi^{\mathrm{LD-IR}}(\boldsymbol{x}, \Gamma_m) < R\right].$$

An alternative view is that BS m fails to decode the packet if the sum of $Y_m(N)$ random variables of the kind $\log_2(1 + \Gamma_m Z)$ is less than R. Following the approach of [104], $\Phi^{\mathrm{LD-IR}}(\boldsymbol{x}, \Gamma_m)$ is approximated with a Gaussian random variable with mean $\mu_{\mathrm{LD-tot}}(m) \triangleq Y_m(N)\mu(\Gamma_m)$ and variance $\sigma^2_{\mathrm{LD-tot}}(m) \triangleq Y_m(N)\sigma^2(\Gamma_m)$, where

$$\mu(\Gamma_m) \triangleq \mathbb{E}[\log_2(1 + \Gamma_m Z)] = \log_2(e) e^{1/\Gamma_m} E_1(1/\Gamma_m)$$

and

$$\begin{aligned} \sigma^2(\Gamma_m) &\triangleq \mathbb{E}[(\log_2(1 + \Gamma_m Z) - \mu(\Gamma_m))^2] \\ &= \tfrac{2}{\Gamma_m} \log_2^2(e) e^{1/\Gamma_m} G_{3,4}^{4,0}\left(1/\Gamma_m \big|_{0,-1,-1,-1}^{0,0,0}\right) - \mu^2(\Gamma_m), \end{aligned}$$

where $E_1(x) \triangleq \int_1^\infty t^{-1} e^{-xt} dt$ and $G_{p,q}^{m,n}\left(z\big|_{b_1,\ldots,b_q}^{a_1,\ldots,a_p}\right)$ is the Meijer G function. Therefore the approximation

$$p_{n+1}^{\mathrm{LD-IR}}(\boldsymbol{x}, \boldsymbol{\Gamma}, R) \approx \prod_{m=1}^{M} \mathrm{Q}\left[\frac{\mu_{\mathrm{LD-tot}}(m) - R}{\sigma_{\mathrm{LD-tot}}(m)}\right]$$

holds, where $\mathrm{Q}[\cdot]$ is the tail probability of the standard Gaussian distribution.

A.3.5 Joint Decoding with HARQ-IR (JD-HARQ-IR)

For JD-HARQ-IR, the serving BS keeps accumulating information coming from the cooperating BSs at each frame, so that the spectral efficiency is given by

$$\Phi^{\mathrm{JD-IR}}(\boldsymbol{x}, \boldsymbol{\Gamma}) \triangleq \sum_{j=1}^{n} \log_2 \left(1 + \sum_{m=1}^{M} y_m(j) \Gamma_m |h_m(j)|^2 \right) . \tag{A.11}$$

The expression of the outage probability after n frames is therefore

$$p_{n+1}^{\mathrm{JD-IR}}(\boldsymbol{x}, \boldsymbol{\Gamma}, R) = \mathbb{P}\left[\Phi^{\mathrm{JD-IR}}(\boldsymbol{x}, \boldsymbol{\Gamma}) < R \right] . \tag{A.12}$$

A closed-form expression of (A.12) cannot be derived, even for the simple case of large n and $\Gamma_m = \Gamma$ for all m. Indeed, in Appendix A.A.1 it is discussed how even a Gaussian approximation of (A.11) cannot be exploited since a closed-form expression of the variance σ_Φ^2 does not exist.

A.4 Base Station Selection

Solving problem (A.3) is not at easy task at all. Being an integer programming problem, its solution requires an exhaustive search (ES) over the $(M+1)^N$ configurations of active BSs at each frame. Indeed, as the maximum number of active BSs M and the maximum number of (H)ARQ frames N increase, the ES becomes computationally unfeasible. By adopting a continuous relaxation of the problem, when the outage probability has a closed-form tractable expression a solution can be found, as shown in Section A.4.2 for LD-ARQ and MT positioned at the same distance from all the M surrounding BSs. In the general case, there is the need of a heuristic algorithm for recursively computing the number of active BSs at each frame. The approach being introduced in the next section provides a low complexity suboptimal solution to problem (A.3), whose performance results extremely close to ES, thus justifying its adoption in real scenarios.

A.4.1 The Recursive Search (RS)

The recursive search (RS) algorithm iteratively computes the number of active BSs at each frame n, by starting from a static feasible solution and then modifying the x_n of a single unit, one at a time in a greedy fashion. In fact, the recursive procedure generally decreases the number of active BSs at early frames and decreases it at later frames, trying to reduce the average number

of active BSs (A.2). The rationale behind this procedure is that if correct decoding occurs at an early stage, the BSs scheduled for cooperation in the next frames are not active. On the other hand, to get an early decoding, a high number of active BSs is necessary at the earliest frames. Therefore, a trade-off must be designed between probability of early decoding and number of active BSs at early frames. With such assumptions, RS looks for the best trade-off by simultaneously ensuring that the QoS constraint on the outage probability is met. Although RS is generally suboptimal, in Section A.4.2 it is shown that in a particular case (LD-ARQ and $\Gamma_m = \Gamma$ for all m) it leads to the optimum solution.

The starting configuration of RS is a static cooperation (SC), which is a condition in which the number of active BSs is the same at each frame, i.e.,

$$x_n^{(0)} = x_{\min} \triangleq \min\{1 \le M' \le M : p_{N+1}([M', \ldots, M'], \mathbf{\Gamma}, R) \le \theta\}. \quad \text{(A.13)}$$

The value of $x_n^{(0)}$ is obtained via a search procedure.

At iteration $i+1$, with $i = 0, 1, \ldots$, the BS configuration is modified by selecting two frame indices s and d, with $s < d$. Then the number of active BSs at the early frame s is decreased by one and the number of active BSs at the later frame d is instead increased by the same quantity, i.e.,

$$x_s^{(i+1)} = x_s^{(i)} - 1\,, \quad x_d^{(i+1)} = x_d^{(i)} + 1\,.$$

Then if

$$f(\boldsymbol{x}^{(i+1)}) \le f(\boldsymbol{x}^{(i)})$$

and condition (A.3b) is still satisfied, then the new BS configuration $x^{(i+1)}$ is saved as the tentative optimum configuration of active BSs ($\bar{\boldsymbol{x}}$). This means that at each iteration the total average number of cooperating BSs is decreased while meeting constraints (A.3b) and (A.3c). The choice of s and d satisfies

$$s, d \in \{1, 2, \ldots, N\} \text{ and } s < d\,, \quad \text{(A.14)}$$

starting from $s = 1$. Moreover, for a given value of s, d is first set to N and then reduced, thus first activating BSs at the later stages, to have the sharpest decrease of $f(\boldsymbol{x}^{i+1})$. The pseudocode of the RS algorithm is reported in Table A.2.

Note that with RS the total number of active BSs is kept fixed, i.e.,

$$\sum_{n=1}^{N} x_n^{(i)} = x_{\min} N\,. \quad \text{(A.15)}$$

If the MT is at the same distance from all the surrounding BSs and ARQ is used, RS assures that the outage constraint is always satisfied without the

Table A.2: Recursive Search Algorithm.

Set $x_n^{(0)} = x_{\min}$ from (A.13), for all $n = 1, \ldots, N$;
Set $i = 0$
for $s = 1$ to $\lceil \frac{N}{2} \rceil$
  for $d = N$ down to $s + 1$
    while $x_s^{(i)} > 0$
      $x_s^{(i+1)} = x_s^{(i)} - 1$;
      if $x_d^{(i)} < M$
        then $x_d^{(i+1)} \leftarrow x_d^{(i)} + 1$
        otherwise break;
      end if
      if $p_{N+1}(\boldsymbol{x}^*, \boldsymbol{\Gamma}, R) \leq \theta$ and if $f(\boldsymbol{x}^{(i+1)}) < f(\boldsymbol{x}^{(i)})$
        then $\bar{\boldsymbol{x}} \leftarrow \boldsymbol{x}^{(i+1)}$;
      end if
    $i \leftarrow i + 1$
    end
  end
end
Return $\bar{\boldsymbol{x}}$ as the set of active BSs.

need of checking it explicitly. In fact, (A.7) assures that in this case the outage probability only depends on the total number of active BSs across all frames. However, in general this is not the case, therefore the outage constraint (A.3b) must be checked at each iteration. Simulation results reported in Section A.5 show that the performance of RS is really close to the optimum solution provided by ES, in all considered policies. Moreover, RS represents a low complex solution since it can be easily shown that it converges in $O(MN^2)$ iterations, i.e. in polynomial time. On the other hand, the ES requires the evaluation of (A.2) for each possible configuration of active BSs, therefore requiring $O\left((M+1)^N\right)$ operations, i.e., an exponential number on the parameter N. Genetic approaches as well as sphere decoding algorithms have been also proposed in the literature [98]: they lead to good results but at the cost of a higher computational burden.

A.4.2 A closed-form expression for LD-ARQ

Considering LD-ARQ leads to a closed-form solution of the continuous relaxation of problem (A.3) in a specific case. However, before proceeding with the specific solution, it is worth to notice why the rationale behind the RS algorithm is justified.

Theorem A.1 *Vector* $\boldsymbol{x}$ *solving the minimization problem (A.3) for LD-ARQ satisfies*

$$x_n \leq x_{n+1}, \qquad n = 1, 2, \ldots, N-1.$$

Proof: See Appendix A.A.3 ■

If the MT is positioned at the same distance from the M surrounding BSs, i.e. $\Gamma_m = \Gamma$ for $m = 1, \ldots, M$, a closed-form solution can be derived, as it is shown in Appendix (A.A.4). The solution provides

$$x_n = \begin{cases} -\log_a(1 - x_{n+1} \log a) & 1 \leq n \leq N-1 \\ -\log_a \frac{\delta}{\theta} & n = N\,, \end{cases} \tag{A.16}$$

where δ is a solution of the implicit equation

$$\mathscr{E}_N(\delta) = 1\,, \tag{A.17}$$

with

$$\begin{array}{rcll} \mathscr{E}_1(\delta) & \triangleq & \delta & \\ \mathscr{E}_2(\delta) & \triangleq & \delta\left[1 + \log \frac{\delta}{\theta}\right] & \\ \mathscr{E}_n(\delta) & = & \mathscr{E}_{n-1}(\delta)\left[1 + \log \frac{\mathscr{E}_{n-1}(\delta)}{\mathscr{E}_{n-2}(\delta)}\right] & n = 3, \ldots, N\,. \end{array} \tag{A.18}$$

Eq. (A.17) is trascendent, therefore it has to be solved by the use of numerical methods. However, it has a unique solution, as it is proved in Appendix A.A.4.

The last step is to find a solution in $\mathbb{Z}^N$, which is done by approximating the solution (A.16) with the vertex in $\mathbb{Z}^N$ of the hypercube around $\boldsymbol{x}$ providing the least average number of active BSs and satisfying the outage constraint. Such discretization process is a source of sub-optimality for the analytical solution, but in Section A.5 it is shown that it actually provides performance close to the optimum.

A.5 Numerical Results

The considered simulative scenario consists of a cellular system in which a single MT is present in the studied sector. It is surrounded by $M = 3$ BSs arranged on three vertices of an hexagon of unitary side, dividing a cell in three sectors. Antenna attenuation for out of sector signals is assumed to be 20 dB [105], while the maximum number of retransmission frames is set to $N = 3$. Rayleigh fading affects the channel while path loss is assumed with

coefficient 3.4, therefore the SNR of the link between the MT and BS m at distance d is given by

$$\Gamma_m = \Gamma_0 \, (d)^{-3.4} \, |\xi|^2 \, ,$$

where Γ_0 is the average SNR at unitary distance and ξ is a complex zero-mean Gaussian random variable with unitary variance modelling the Rayleigh fading.

Interference is kept into account by considering the presence of MTs in the neighboring sectors with *interference probability* q. The condition $q = 0$ means total absence of interferers, while $q = 1$ corresponds to the case in which all the interfering MTs are transmitting. The positions of interfering MTs is random within their sectors. Both intra-cell and inter-cell interference is considered by including a ring of surrounding cells around the studied sector, assuming a MT (active with probability q) for each sector of the ring.

In the following subsections, a number of communication policies are compared. Within each policy, two macro diversity approaches are considered: 1) flexible cooperation (FC), in which the active BSs are selected according to the methods proposed in this work, and 2) static cooperation (SC), where BSs are activated according to the rule provided by (A.13). For FC, results for both RS and ES (as a benchmark) are reported. The maximum allowed outage probability is set at $\theta = 10^{-2}$.

A.5.1 Outage Probability in the Sector

To provide examples of the distribution of the outage probability in the studied sector, this section reports spatial plots for two specific cases, assuming $R = 2$ bit/s/Hz and activating al the $M = 3$ BSs. Supposing that the serving BS is located at the point with coordinates $(0, 0)$, Fig. A.2 focuses on the policy LD-HARQ-IR with $q = 0$. It can be observed that the outage probability increases with the distance from the $M = 3$ BSs, revealing that the position at the edge of the cell sectors is the worst case. Similar conclusions can be drawn looking at Fig. A.3 which reports the spatial distribution of the average (with respect to the interfering nodes' positions) outage probability for LD-ARQ and $q = 1$. The point at the maximum distance from all the BSs results to be the worst case for the outage probability also using the other policies listed in Section A.3, the corresponding plots being omitted.

A.5.2 Coverage improvement

In order to evaluate the cell coverage improvement allowed by the innovative FC policies introduced so far in a macro diversity approach, in this section

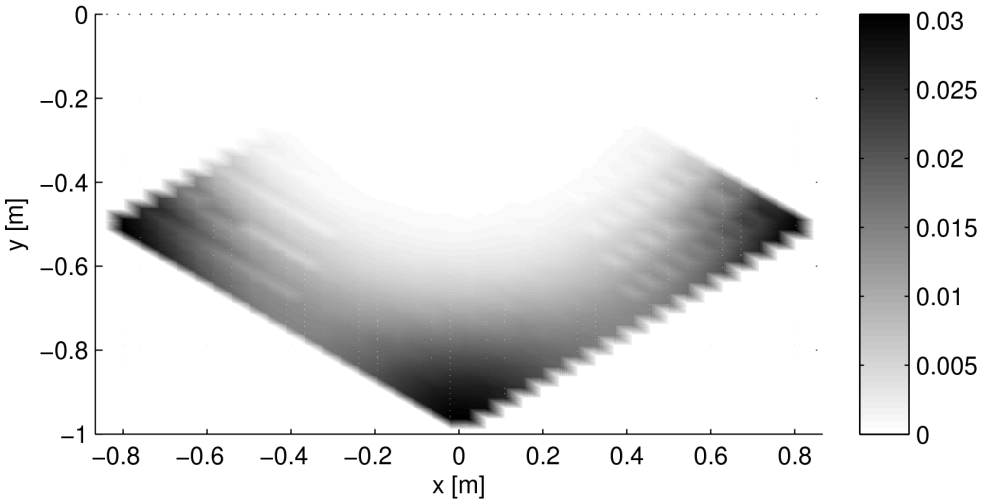

Figure A.2: Final outage probability p_{N+1} as a function of the MT position in a sector with LD-HARQ-IR and in the absence of interference ($q = 0$). The serving BS is located at coordinates (0,0) in the plot.

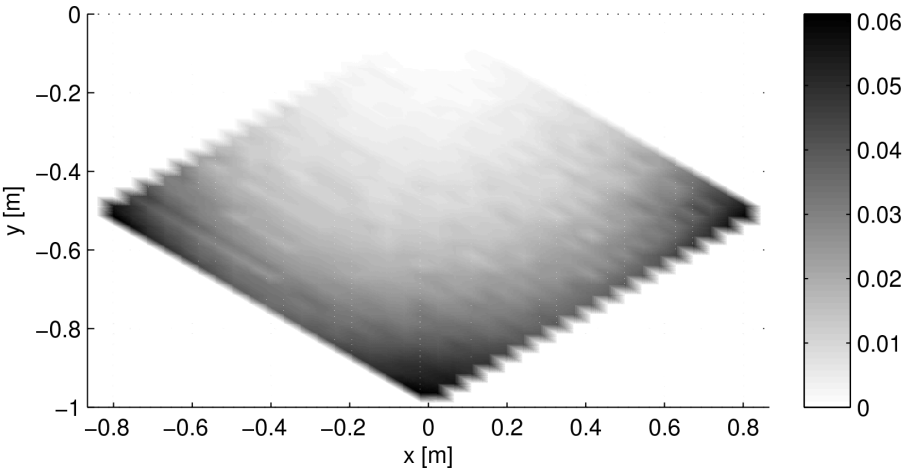

Figure A.3: Final outage probability p_{N+1} as a function of the MT position in a sector with LD-ARQ and in the presence of interference ($q = 1$). The serving BS is located at coordinates (0,0) in the plot.

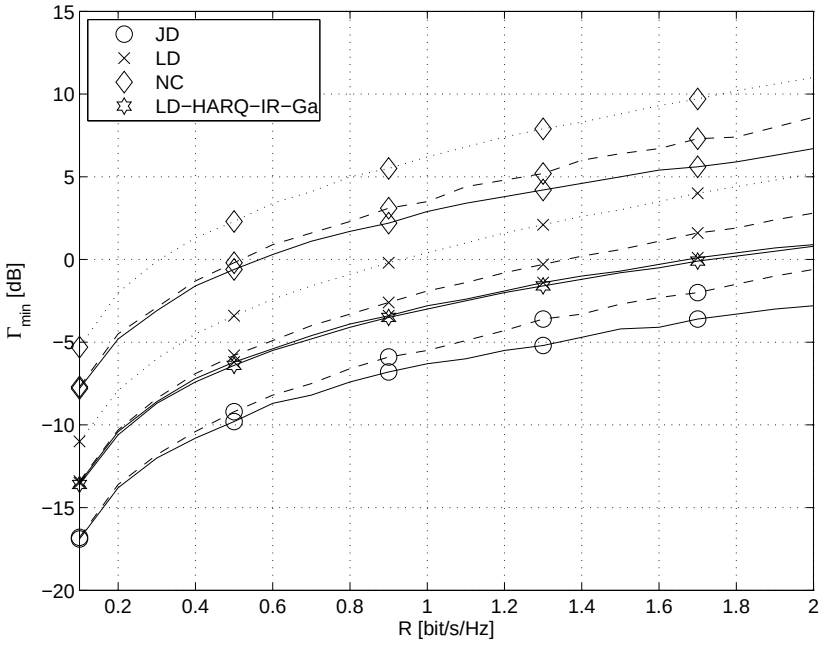

Figure A.4: Minimum value of the SNR at the unitary distance vs the spectral efficiency of the first transmission for various decoding schemes. Solid line: HARQ-IR; dashed line: HARQ-CC; dotted line: ARQ.

the MT is placed at the edge of three sectors, equally distant from the $M = 3$ surrounding BSs, that is in the worst case in the sector in terms of outage probability. It is assumed that all the $M = 3$ BSs are activated during each frame, i.e., $\boldsymbol{x} = [M, M, M]$.

Interference-free Scenario

When interferers are switched off ($q = 0$), the minimum acceptable value of the SNR Γ_0 can be evaluated by letting the MT transmit while imposing that the outage constraint (A.3b) is fulfilled, i.e., that means finding

$$\Gamma_{\min} = \min_{\Gamma_0} \{\Gamma_0 : p_{N+1}(\boldsymbol{x}, \boldsymbol{\Gamma}, R) \leq \theta\} .$$

Fig. A.4 shows the value of $\Gamma_{\min}$ as a function of the normalized spectral efficiency R using a) macro diversity with JD, b) macro diversity with LD and c) no cooperation (NC), i.e. only the serving BS is active. Since all BSs are active in all frames, a unique line is reported for all cooperation and error protection scheme, since distinction between RS and ES has not meaning at this point. JD significantly outperforms both macro diversity with LD and

NC by reducing $\Gamma_{\min}$ by about 3.5 dB and 9 dB, respectively, for $R = 0.4$ bit/s/Hz. Moreover, also the Gaussian approximation discussed in Section A.3.4 (LD-HARQ-IR-Ga) is shown in Fig. A.4, so that it can be observed that it provides a satisfying approximation of the simulated performance.

Fig. A.4 can be read also from other different viewpoints: in fact, results of $\Gamma_{\min}$ can be translated in terms of coverage radius. As an example of the increased coverage properties yielded by FC schemes, it can be observed that NC achieves a spectral efficiency of 0.4 bit/s/Hz at the edge of a hexagonal cell with unitary side, while the same spectral efficiency is achieved by JD and LD on cells with a side of 1.3 and 1.8, respectively, thus substantially extending the coverage radius of the BSs.

Another reading of Fig. A.4 consists of looking at the achieved spectral efficiency R as a function of the SNR, therefore obtaining information about diversity and multiplexing gains [106]. In particular, note that HARQ-CC (dashed lines) has a higher multiplexing gain with respect to ARQ (dotted lines), while HARQ-IR (solid lines) exhibits also a diversity gain, as can be seen from the slope at high SNRs.

Interference Scenario

If the interferers are switched on, their transmissions produce interference to the studied MT. The outage probability $p_{N+1}(\boldsymbol{x}, \boldsymbol{\Gamma}(q), R)$ is therefore also a function of the interference probability q which determines the signal to interference plus noise ratio (SINR) vector $\boldsymbol{\Gamma}(q)$, so that a threshold for the maximum value of q can be found for having sustainable communications in terms of outage probability, for each cooperation policy. Fig. A.5 shows the maximum value of the interference probability, i.e.,

$$q_{\max} = \max_{q} \{ q : p_{N+1}(\boldsymbol{x}, \boldsymbol{\Gamma}(q), R) \leq \theta \} ,$$

as a function of the spectral efficiency R of the first transmission that can be reached by the MT of interest, despite interference. Also in this case JD provides the highest immunity to interference for a fixed spectral efficiency or, equivalently, letting BSs cooperate increases the achievable spectral efficiency for a given level of interference.

A.5.3 Randomly Dropped Users

In this section, MTs are randomly dropped in the sector. In each case, the cooperation and error protection combinations of Table A.1 are considered in order to determine the number of active BSs for each frame. The cases in

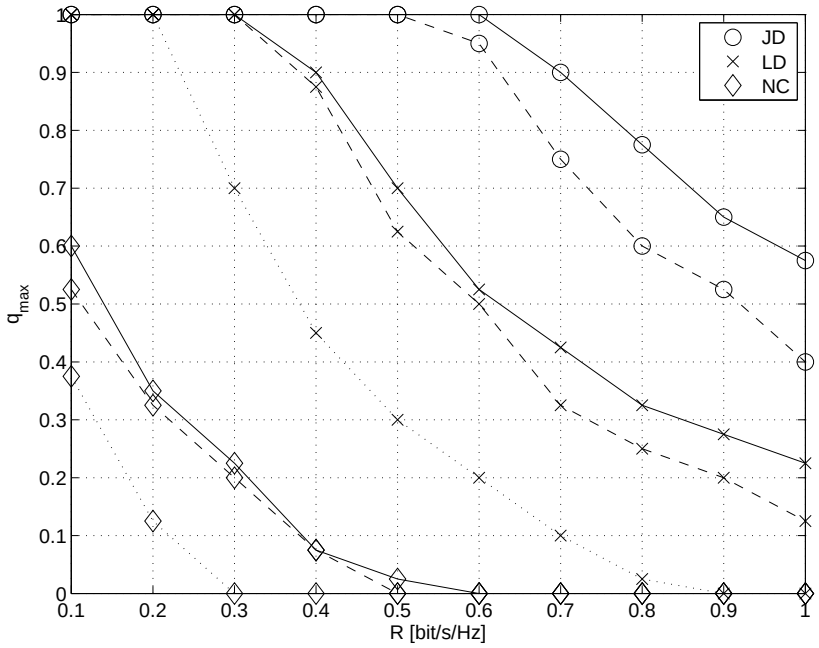

Figure A.5: Maximum allowed interference probability $q_{\max}$ vs the spectral efficiency of the first transmission for various decoding schemes. Solid line: HARQ-IR; dashed line: HARQ-CC; dotted line: ARQ.

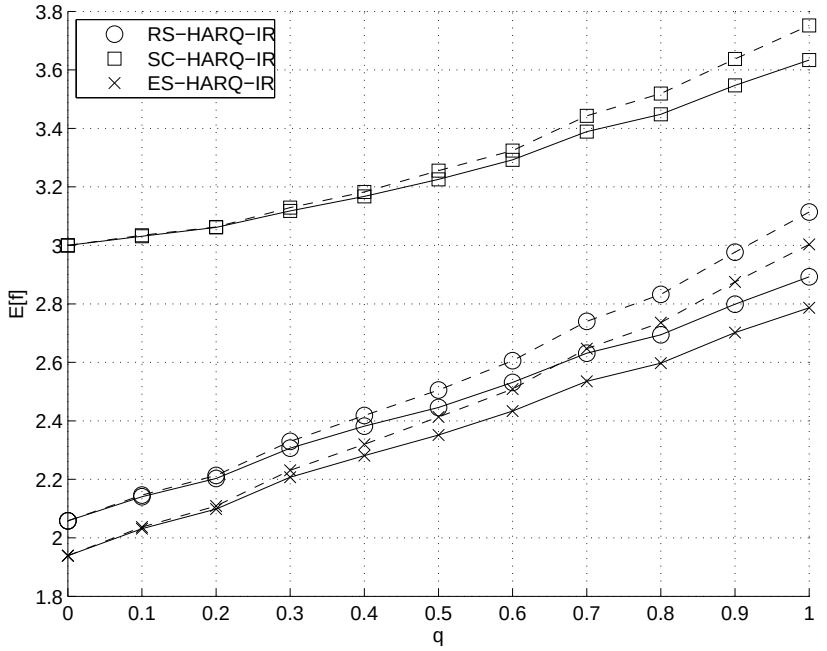

Figure A.6: Total average number of active BSs as a function of q for HARQ-IR. Solid line: JD; dashed line: LD.

which the outage probability constraint is not statistically met are dropped and the MT transmission re-scheduled. From data on the number of active BSs, the performance metrics (A.2), (A.4) and (A.5) are computed. In the following, the spectral efficiency R is set to 0.4 bit/s/Hz.

Total Average Number of Active BSs

In analogy with the well known soft handover overhead, the function $f(\boldsymbol{x})-1$ can be seen as a *macro diversity overhead*, which is proportional to the additional hardware resources required in the cellular system to implement uplink BS cooperation. Indeed, active BSs must be able to manage additional uplink signal processing in order to cooperate with the serving BS. If no extra hardware is provided in the active BSs, cooperation translates into a spectral efficiency reduction. Moreover, the backhaul network will be partially used for the required exchange of information between cooperating BSs, determining an additional overhead, especially for JD, which will be considered in more details in the following.

For HARQ-IR, Fig. A.6 shows $\mathbb{E}[f(\boldsymbol{x})]$ as a function of the interference probability q, where the average is taken over the random MT position and $\Gamma_0 = 0$ dB. JD outperforms LD in the entire range of q (especially for values

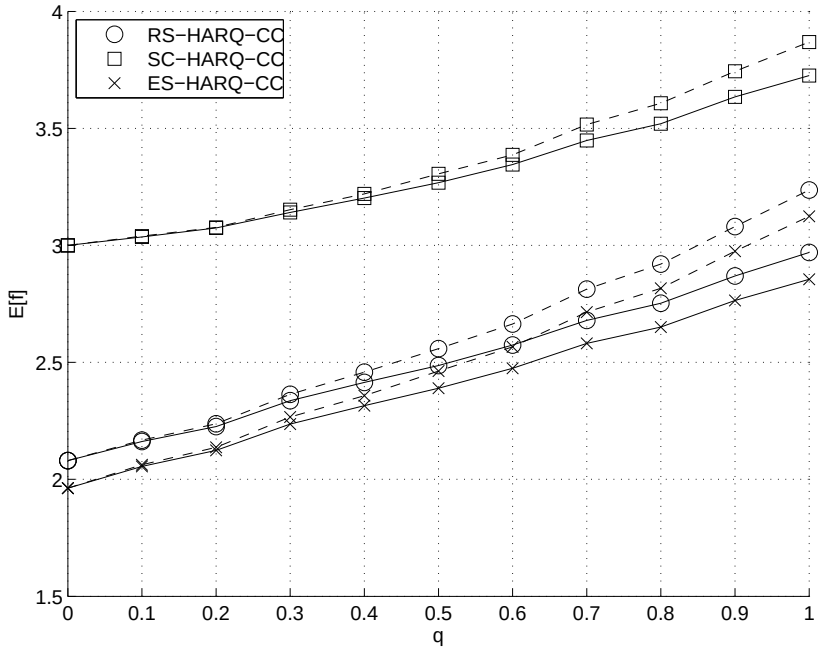

Figure A.7: Total average number of active BSs as a function of q for HARQ-CC. Solid line: JD; dashed line: LD.

of q close to 1), thus it can be concluded that information sharing among the BSs significantly reduces the average number of active slots. For both JD and LD, RS always outperforms SC while at the same time providing performance really close to the optimum solution: in fact, for $q = 0.5$, more than 25% of active BSs are saved in the JD case.

Fig. A.7 shows $\mathbb{E}[f(\boldsymbol{x})]$ for HARQ-CC as a function of q. Also in this case RS has only slightly suboptimal performance with respect to the best configuration (ES) of active BSs at each frame; in particular, for $q = 0.5$, ES outperforms RS only by 4% in the JD case, with a consistent reduction in computational complexity: for example, in the performed simulations, for RS the search is completed in less than 5 iterations in all cases, instead of $(M + 1)^N = 64$ iterations for ES. Moreover, RS significantly reduces the backhaul usage with respect to SC, since for $q = 0.5$, it requires 24% less total active BSs on average for the JD case.

Fig. A.8 shows $\mathbb{E}[f(\boldsymbol{x})]$ for LD-ARQ as a function of q. Even in this case, RS significantly outperforms SC while only slightly under-performing ES. For example, the resource saving with respect to SC at $q = 0.5$ is of about 20%.

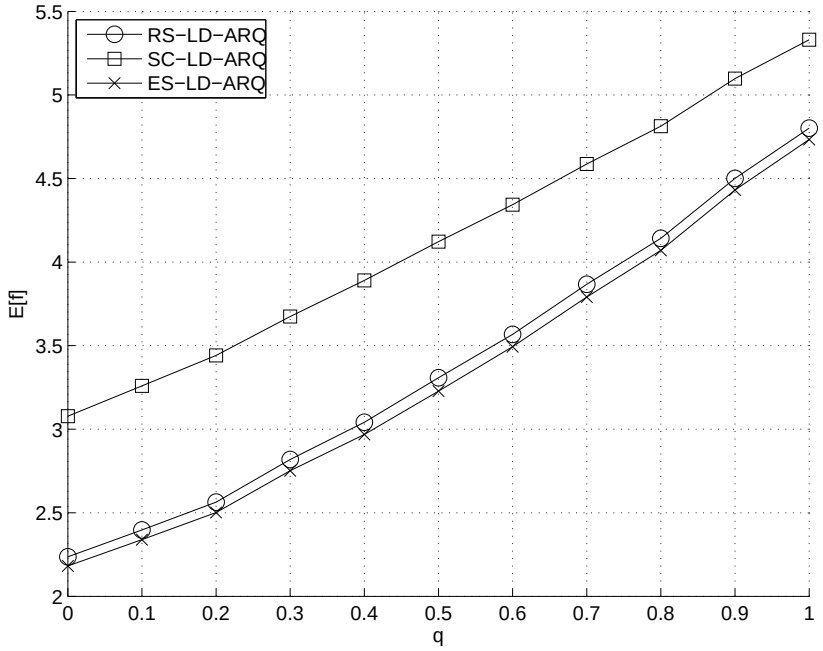

Figure A.8: Total average number of active BSs as a function of q for ARQ.

Backhaul Occupation

Fig. A.9 shows $\mathbb{E}[B_P]$ as a function of q by remarking that RS allows a strong saving in the resource usage with respect to SC. As an example, for $q = 0.5$ and JD-HARQ-IR, RS requires 52% less backhaul links usage on average, while the saving increases up to 64% for JD-HARQ-CC for the same value of q.

Payload Throughput

Fig. A.10 shows $\mathbb{E}[\pi]$ as a function of q for a system using HARQ-IR. It can be observed how SC slightly outperforms RS. This result was expected since RS tends to reduce the number of active BSs during earlier frames while increasing it in the latter frames. As a consequence, while the average number of active BSs is reduced, the average decoding delay is increased, since there is a higher probability of late decoding. On the other hand, to confirm the close similarity between ES and RS, they have a very close performance also in terms of payload throughput.

Fig. A.11 shows $\mathbb{E}[\pi]$ as a function of q for HARQ-CC. Also in this case it can be observed that SC outperforms RS. By comparing Fig.s A.6 and A.7 with Fig.s A.10 and A.11 it can be observed that while requiring a

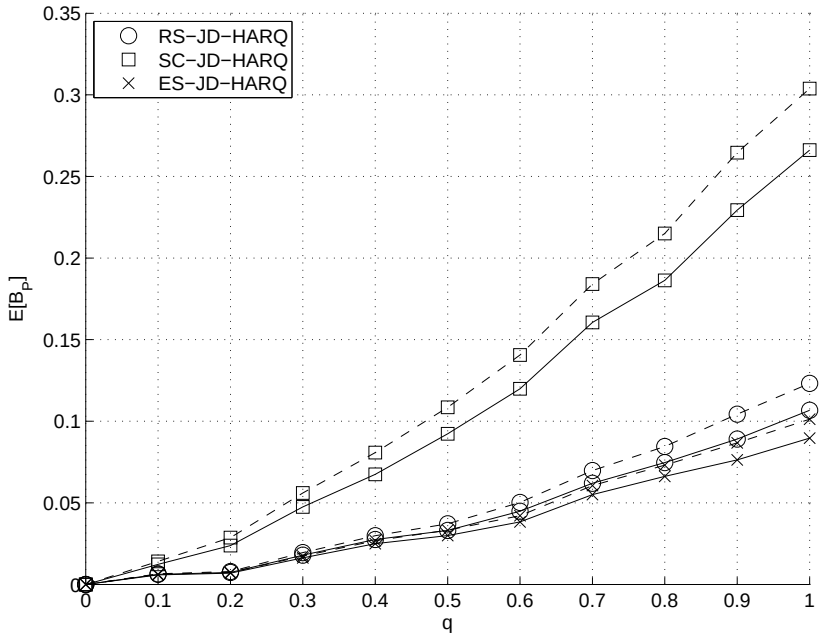

Figure A.9: Total average of number backhaul transmissions as a function of q for JD-HARQ. Solid line: HARQ-IR; dashed line: HARQ-CC.

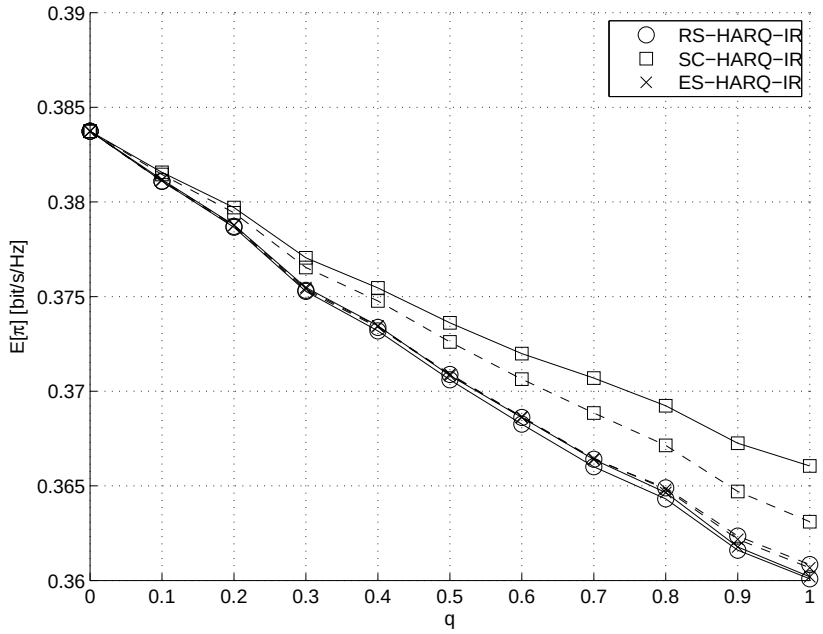

Figure A.10: Average payload throughput as a function of q for HARQ-IR. Solid line: JD; dashed line: LD.

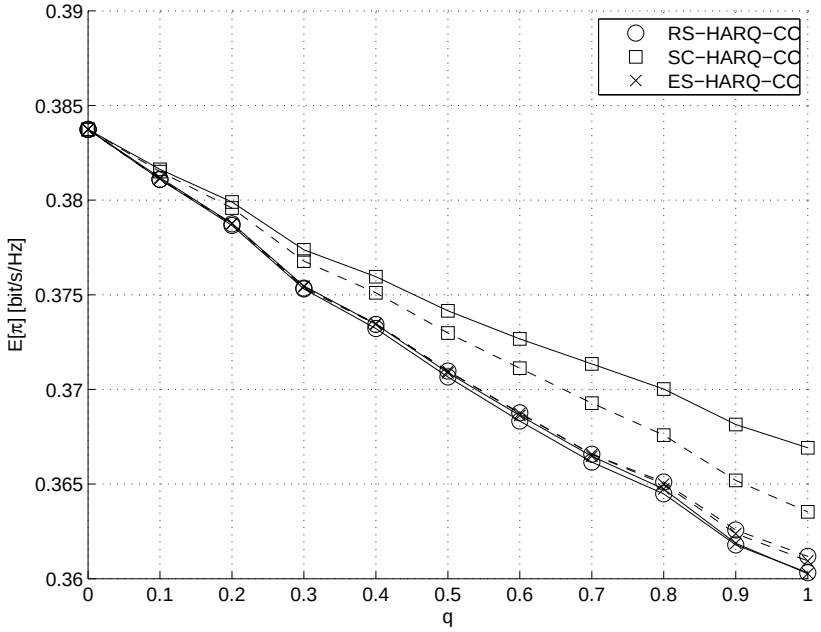

Figure A.11: Average payload throughput as a function of q for HARQ-CC. Solid line: JD; dashed line: LD.

very close payload throughput, the various techniques lead to a substantially different number of active BSs, i.e. different backhaul occupancies. Hence, $1/f(\boldsymbol{x})$ from Fig.s A.6 and A.7 can be read as the energy efficiency of the transmission.

A.5.4 LD-ARQ and $\Gamma_m = \Gamma$

To compare the performance of the analytical solution (AS) derived in Section A.4.2 with the optimal solution (ES) of (A.3) for the LD-ARQ case, in Fig. A.12 the total average number of active BSs is shown in the case in which the MT is positioned at the edge of the cell and the transmission rate is set equal to $R = 2$ bit/s/Hz. It can be seen that the analytical solution is clearly optimal. Finally, it can be observed that in the case in which the SNR toward all the BSs is the same, RS is fully optimal in the LD-ARQ case.

A.6 Conclusions

In uplink cellular communications, a MT in a sector transmits to a set of surrounding BSs in a multi-cell processing scenario. Error control schemes

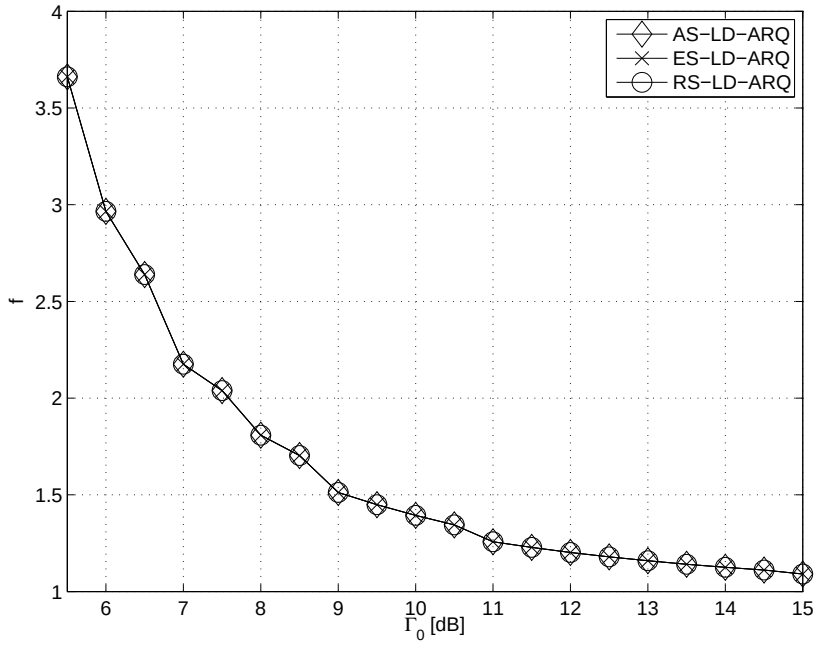

Figure A.12: Total number of active BSs when the MT is positioned at the edge of the cell with ARQ.

as well as cooperation policies among BSs are allowed in order to select which BSs should be activated in the retransmission frames with the aim of minimizing the average number of active BSs during all frames, under a constraint on the maximum allowed outage probability. The iterative proposed algorithm, RS, provides an efficient and just slightly suboptimal solution to the problem, for each selected combination of error control scheme and cooperation policy. The proposed algorithm permits a strong performance improvement (up to 64%) in the backhaul usage with respect to existing cooperative solutions (SC), as it is also confirmed by the provided simulation results. It can be therefore concluded that RS is a computationally cheap but effective solution to the problem of minimizing the backhaul usage under a QoS constraint.

A.A Appendix

A.A.1 Gaussian Approximation for JD-HARQ-IR

It can be noticed that $W_j \triangleq \sum_{m=1}^{M} y_m(j)|h_m(j)|^2$ is Gamma distributed with scale 1 and shape x_j and

$$\mathbb{E}[\log_2(1+\Gamma W_j)] = \frac{1}{(x_j-1)!}\int_0^{+\infty} \log_2(1+\Gamma w) w^{x_j-1} e^{-w} dw \ .$$

It follows [107] that

$$\begin{aligned}\mathbb{E}[\log_2(1+\Gamma W_j)] =& \frac{1}{(x_j-1)!\log 2}\times \\ &\left[\left(\frac{1}{\Gamma}\right)^{x_j}\frac{\pi}{x_j\sin(x_j\pi)}{}_1F_1\left(x_j; x_j+1; \frac{1}{\Gamma}\right) - \frac{1}{(x_j-1)!}p^{-x_j}\times\right. \\ &\left.\left\{\left[\log\frac{1}{\Gamma} - \Psi(x_j)\right] - \frac{1}{\Gamma(1-x_j)}{}_2F_2\left(1,1;2,2-x_j;\frac{1}{\Gamma}\right)\right\}\right] ,\end{aligned}$$

with

$${}_pF_q\left(r_1,\ldots,r_p; s_1,\ldots,s_q; z\right) \triangleq \sum_{k=0}^{\infty}\frac{\prod_{i=1}^{p}(r_i)_k}{\prod_{i=1}^{q}(s_i)_k}\frac{z^k}{k!}$$

$$\Psi(z) \triangleq \frac{1}{(z-1)!}\frac{\partial(z-1)!}{\partial z}$$

and $(r)_k = r(r+1)\ldots(r+k-1)$ is the rising factorial of r. Finally it is

$$\mu_\phi = \sum_{j=1}^{n}\mathbb{E}[\log_2(1+\Gamma W_j)].$$

However, a closed-form expression for $\mathbb{E}[\log_2^2(1+\Gamma W_j)]$ does not exist, therefore σ_ϕ^2 cannot be determined analytically.

A.A.2 Closed-form Expression of Probability (A.10) and ARQ

Probability (A.10) can be rewritten thanks to [108] as

$$\mathbb{P}\left[\sum_{j=1}^{n}\sum_{m=1}^{x_j}\Gamma_m|h_m(j)|^2 < 2^R-1\right] = 1- \left(\prod_{m=1}^{M}\Gamma_m^{Y_m(n)}\right)\sum_{m=1}^{M}\sum_{l=1}^{Y_m(n)}\frac{\phi_{m,l}(-\Gamma_m)}{\Gamma_m^{Y_m(n)-l+1}}\times \Psi\left(\Gamma_m(2^R-1), Y_m(n)-l\right) ,$$

where $\Psi(a,b)$ is the cdf of a Poisson random variable of parameter a, i.e.,

$$\phi_{m,l}(b) = (-1)^{l-1}\sum_{\Omega(M,m,l)}\prod_j \binom{i_j+Y_j(n)-1}{i_j}\tau_j(b) .$$

The set $\Omega(M,m,l)$ contains partitions of $l-1$ through the positive integers indices i_j, such that $\sum_{j=1,j\neq m}^{M} i_j = l-1$ and $\tau_j(b) = (\Gamma_j+b)^{-(Y_j(n)+i_j)}$.

A.A.3 Proof of Theorem A.1

The proof of theorem A.1 in the case $N=2$ is reported here. It can be easily extended to the general case. Let $x_+ \leq M$ and $x_- \leq x_+$ be two integers. First observe that both configurations of vector $\boldsymbol{x}$, $[x_-, x_+]$ and $[x_+, x_-]$, satisfy the outage probability constraint since for LD-ARQ the outage probability is only determined by the set of active BSs within the whole transmission irrespective of their order, see (A.7). The aim is to show that

$$f(x_-, x_+) \leq f(x_+, x_-) .$$

From (A.2) and (A.7) it is

$$f(x_+,x_-) - f(x_-,x_+) = x_+ + x_-\prod_{m=1}^{x_+} a_m + - \left[x_- + x_+\prod_{m=1}^{x_-} a_m\right] .$$

Using (A.8) it holds

$$f(x_+,x_-) - f(x_-,x_+) \geq (x_+ - x_-) + + \prod_{m=1}^{x_-} a_m(x_- a_{x_-}^{x_+-x_-} - x_+) .$$

Then it must be observed that for fixed $\prod_{m=1}^{x_-} a_m$, a_{x_-} is minimized when $a_1 = \ldots = a_{x_-} = a \triangleq \sqrt[x]{\prod_{m=1}^{x_-} a_m}$, therefore defining

$$g(a) \triangleq x_+ + x_- a^{x_+} - (x_- + x_+ a^{x_-})$$

it follows that

$$f(x_+, x_-) - f(x_-, x_+) \geq g(a)\,.$$

Now, note that $g(0) = x_+ - x_- \geq 0$ and $g(1) = 0$ and also $\frac{\partial g}{\partial a} = \frac{x_- x_+}{a}(a^{x_+} - a^{x_-}) \leq 0$ for $0 \leq a \leq 1$. Since $g(a)$ is a continuous function of a, the minimum of $g(a)$ is assumed for $a = 1$ and it is equal to $g(1) = 0$, hence $g(a) \geq 0$.

A.A.4 Closed-form solution of (A.3) for $\Gamma_m = \Gamma$.

The average outage probability after n frames (A.7) can be written as

$$p_{n+1}^{\mathrm{LD-A}}(\boldsymbol{x}, \boldsymbol{\Gamma}) = a^{\sum_{j=1}^{n} x_j}\,,$$

with

$$a \triangleq 1 - e^{-\frac{2^R - 1}{\Gamma}}.$$

With the application of the Lagrange multipliers method, the problem to solve turns out to be

$$\min_{\boldsymbol{x},\delta} \Lambda(\boldsymbol{x}, \delta) = \min_{\boldsymbol{x},\delta} \left[f(\boldsymbol{x}) - \delta \left(\sum_{n=1}^{N} x_n - \log_a \theta \right) \right]. \tag{A.19}$$

Note that in defining the Lagrangian (A.19) the bounds (A.3c) on x_n are not taken into account. Indeed, from Theorem A.1 if the solution of the unbounded problem (A.3a) with the constraint (A.3b) yields $x_n > M$ for some n, for sure it will be $x_i > M, n \leq i \leq N$. In such a case, the values $x_i = M, n \leq i \leq N$ will be set, therefore attacking the problem (A.3a) with the outage constraint (A.3b) on the $n-1$ unknowns $x_1, \ldots, x_{n-1}$ (all the others being equal to M). Setting to zero the derivatives of (A.19)

$$\frac{\partial \Lambda}{\partial x_n} = a^{\sum_{j=1}^{n-1} x_j} \left[1 + \left(\sum_{l=n}^{N-1} x_{l+1} a^{\sum_{j=n}^{l} x_j} \right) \log a \right] - \delta = 0, \tag{A.20a}$$

with $n = 1, 2, \ldots, N$, δ provided by (A.17) and (A.18). Therefore

$$\frac{\partial \Lambda}{\partial \delta} = -\sum_{n=1}^{N} x_n + \log_a \theta = 0. \tag{A.20b}$$

By manipulating the $N+1$ equations (A.20), (A.16) is obtained.

Lastly, notice that (A.17) has a unique solution, since a product of two monotonically increasing functions is a monotone function if either of the two functions is positive valued everywhere. This can be seen by noticing that both the feasible values of δ in $\mathscr{E}_n(\delta)$ and the positiveness of δ, from (A.16), imply $\mathscr{E}_n(\delta) > 0$ for $1 \leq n \leq N-1$. Hence, $\mathscr{E}_N(\delta)$ is a monotone function, therefore (A.17) has a unique solution.

Appendix B

List of Publications

Part of the work presented in this thesis appeared also in the following articles:

- A. Ahmad, D. Zennaro, E. Serpedin, and L. Vangelista, "A Factor Graph Approach to Clock Offset Estimation in Wireless Sensor Networks," submitted to *IEEE Trans. Inf. Theory*.

- A. Ahmad, D. Zennaro, E. Serpedin, and L. Vangelista, "Time-Varying Clock Offset Estimation in Two-Way Timing Message Exchange in Wireless Sensor Networks using Factor Graphs," accepted for publication in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012*.

- T. Erseghe, D. Zennaro, E. Dall'Anese, and L. Vangelista, "Fast Consensus by the Alternating Direction Multipliers Method," *IEEE Transactions on Signal Processing*, vol.59, no.11, pp.5523-5537, Nov. 2011.

- D. Zennaro, S. Tomasin, and L. Vangelista, "Base Station Selection in Uplink Macro Diversity Cellular Systems with Hybrid ARQ," *IEEE J. Sel. Areas Commun.*, vol.29, no.6, pp.1249-1259, June 2011.

- D. Zennaro, E. Dall'Anese, T. Erseghe, and L. Vangelista, "Fast Clock Synchronization in Wireless Sensor Networks via ADMM-based Consensus," *International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2011*, Princeton (NJ), May 9-13, 2011.

- D. Zennaro, S. Tomasin, and L. Vangelista, "Uplink Cell Selection for Cooperative Multi-Cell Networks with Hybrid ARQ," *Proc. of IEEE*

Global Telecommunications Conf. (GLOBECOM), Miami (FL), Dec. 6-10, 2010.

Bibliography

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] I. Chlamtac, M. Conti, and J. J.-N. Liu, "Mobile ad hoc networking: imperatives and challenges," *Ad Hoc Networks*, vol. 1, no. 1, pp. 13 – 64, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870503000131

[3] H. Hartenstein and K. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *Communications Magazine, IEEE*, vol. 46, no. 6, pp. 164 –171, Jun. 2008.

[4] M. Lukac, P. Davis, R. Clayton, and D. Estrin, "Recovering temporal integrity with Data Driven Time Synchronization," in *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, ser. IPSN '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 61–72. [Online]. Available: http://dl.acm.org/citation.cfm?id=1602165.1602173

[5] D. Mills, "Internet time synchronization: The network time protocol," *Communications, IEEE Transactions on*, vol. 39, no. 10, pp. 1482 – 1493, Oct. 1991.

[6] S. Gezici, Z. Tian, G. Giannakis, H. Kobayashi, A. Molisch, H. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 70 – 84, Jul. 2005.

[7] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 3, no. 3, pp. 281 – 323, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870505000144

[8] B. Sadler and A. Swami, "Synchronization in Sensor Networks: an Overview," in *Military Communications Conference, 2006. MILCOM 2006. IEEE*, Oct. 2006, pp. 1 –6.

[9] O. Simeone, U. Spagnolini, Y. Bar-Ness, and S. Strogatz, "Distributed synchronization in wireless networks," *Signal Processing Magazine, IEEE*, vol. 25, no. 5, pp. 81–97, Sept. 2008.

[10] I. K. Rhee, J. Lee, J. Kim, E. Serpedin, and Y. C. Wu, "Clock synchronization in wireless sensor networks: an overview," *Sensors*, vol. 9, no. 1, pp. 56–85, 2009.

[11] J. E. Elson, "Time synchronization in wireless sensor networks," Ph.D. dissertation, UCLA, Los Angeles, California, 2003.

[12] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *SIGOPS Oper. Syst. Rev.*, vol. 36, pp. 147–163, Dec. 2002.

[13] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi, "The flooding time synchronization protocol," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, 2004, pp. 39–49.

[14] R. Kumar, S. Ganeriwal, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *ACM Conf. Embedded Networked Sensor Systems*, 2003.

[15] K.-L. Noh, E. Serpedin, and K. Qaraqe, "A new approach for time synchronization in wireless sensor networks: Pairwise broadcast synchronization," *Wireless Communications, IEEE Transactions on*, vol. 7, no. 9, pp. 3318–3322, Sept. 2008.

[16] O. Simeone and U. Spagnolini, "Distributed time synchronization in wireless sensor networks with coupled discrete-time oscillators," *Wireless Communications and Networking, EURASIP Journal on*, 2007.

[17] L. Schenato and G. Gamba, "A distributed consensus protocol for clock synchronization in wireless sensor network," in *Decision and Control, 46th IEEE Conference on*, Dec. 2007, pp. 2289–2294.

[18] R. Carli, A. Chiuso, S. Zampieri, and L. Schenato, "A PI consensus controller for networked clocks synchronization," in *Automatic Control, IFAC World Congress on*, 2008.

[19] A. A. Syed and J. Heidemann, "Time Synchronization for High Latency Acoustic Networks," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, Apr. 2006, pp. 1 –12.

[20] N. Chirdchoo, W. S. Soh, and K. C. Chua, "MU-Sync: A Time Synchronization Protocol for Undewater Mobile Networks," in *Proc. of ACM WUWNet*, San Francisco, CA, 2008.

[21] J. Liu, Z. Zhou, Z. Peng, and J. H. Cui, "Mobi-Sync: Efficient Time Synchronization for Mobile Underwater Sensor Networks," in *Proc. of ACM WUWNet*, Berkeley, CA, Nov. 2009.

[22] F. Lu, D. Mirza, and C. Schurgers, "D-sync: Doppler-based time synchronization for mobile underwater sensor networks," in *Proceedings of the Fifth ACM International Workshop on UnderWater Networks*, ser. WUWNet '10. New York, NY, USA: ACM, 2010, pp. 3:1–3:8. [Online]. Available: http://doi.acm.org/10.1145/1868812.1868815

[23] H. Abdel-Ghaffar, "Analysis of synchronization algorithms with time-out control over networks with exponentially symmetric delays," *Communications, IEEE Transactions on*, vol. 50, no. 10, pp. 1652 – 1661, Oct. 2002.

[24] D. Jeske, "On maximum-likelihood estimation of clock offset," *Communications, IEEE Transactions on*, vol. 53, no. 1, pp. 53 – 54, Jan. 2005.

[25] Q. Chaudhari, E. Serpedin, and K. Qaraqe, "On Minimum Variance Unbiased Estimation of Clock Offset in a Two-Way Message Exchange Mechanism," *Information Theory, IEEE Transactions on*, vol. 56, no. 6, pp. 2893 –2904, Jun. 2010.

[26] K.-L. Noh, Q. Chaudhari, E. Serpedin, and B. Suter, "Novel Clock Phase Offset and Skew Estimation Using Two-Way Timing Message Exchanges for Wireless Sensor Networks," *Communications, IEEE Transactions on*, vol. 55, no. 4, pp. 766 –777, Apr. 2007.

[27] Q. Chaudhari, E. Serpedin, and K. Qaraqe, "On Maximum Likelihood Estimation of Clock Offset and Skew in Networks With Exponential Delays," *Signal Processing, IEEE Transactions on*, vol. 56, no. 4, pp. 1685 –1697, Apr. 2008.

[28] A. Ahmad, A. Noor, E. Serpedin, H. Nounou, and M. Nounou, "On Clock Offset Estimation in Wireless Sensor Networks with Weibull Distributed Network Delays," in *Pattern Recognition (ICPR), 2010 20th International Conference on*, Aug. 2010, pp. 2322 –2325.

[29] J.-S. Kim, J. Lee, E. Serpedin, and K. Qaraqe, "A robust estimation scheme for clock phase offsets in wireless sensor networks in the presence of non-gaussian random delays," *Signal Processing*, vol. 89, no. 6, pp. 1155 – 1161, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168409000024

[30] ——, "Robust Clock Synchronization in Wireless Sensor Networks Through Noise Density Estimation," *Signal Processing, IEEE Transactions on*, vol. 59, no. 7, pp. 3035 –3047, Jul. 2011.

[31] M. Leng and Y.-C. Wu, "On clock synchronization algorithms for wireless sensor networks under unknown delay," *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 1, pp. 182 –190, Jan. 2010.

[32] "MSP430 F1611 data sheet," Texas Instruments.

[33] J. Hill and D. Culler, "A Wireless Embedded Sensor Architecture for System-Level Optimization," Tech. Rep., 2001.

[34] H. Meyr and G. Ascheid, *Synchronization in Digital Communications, vol.1.* Wiley, 1990.

[35] S. Moon, P. Skelly, and D. Towsley, "Estimation and removal of clock skew from network delay measurements," in *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1, Mar. 1999, pp. 227 –234.

[36] "Definitions and terminology for synchronization networks," ITU-T, Ed., City, State of Publication, 1996, ch. G.810, pp. 400–402.

[37] O. Simeone, U. Spagnolini, G. Scutari, and Y. Bar-Ness, "Physical-layer distributed synchronization in wireless networks and applications," *Physical Communication*, vol. 1, no. 1, pp. 67 – 83, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1874490708000049

[38] Q. Chaudhari and E. Serpedin, "Maximum Likelihood Estimation of clock parameters for synchronization of wireless sensor networks," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, Apr. 2008, pp. 2517 –2520.

[39] Q. Chaudhari, E. Serpedin, and K. Qaraqe, "Minimal cost clock synchronization using a sender-receiver protocol in Wireless Sensornets," in *Performance Evaluation of Computer and Telecommunication Systems, 2008. SPECTS 2008. International Symposium on*, Jun. 2008, pp. 159 –164.

[40] ——, "Some improved and generalized estimation schemes for clock synchronization of listening nodes in wireless sensor networks," *Communications, IEEE Transactions on*, vol. 58, no. 1, pp. 63 –67, Jan. 2010.

[41] Q. Chaudhari and E. Serpedin, "A new scheme for synchronization of inactive nodes in a sender-receiver protocol," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, Apr. 2009, pp. 2905 –2908.

[42] Q. Chaudhari, E. Serpedin, and J.-S. Kim, "Energy-Efficient Estimation of Clock Offset for Inactive Nodes in Wireless Sensor Networks," *Information Theory, IEEE Transactions on*, vol. 56, no. 1, pp. 582 –596, Jan. 2010.

[43] B. Sadler, "Fundamentals of energy-constrained sensor network systems," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 20, no. 8, pp. 17–35, Aug. 2005.

[44] [Online]. Available: http://mail.millennium.berkeley.edu/pipermail/tinyos-help/2008-October/036812.html

[45] [Online]. Available: http://www.tla.co.nz/xtal1.html

[46] C. J. Bovy, H. T. Mertodimedjo, G. Hooghiemstra, H. Uijterwaal, and P. V. Mieghem, "Analysis of end-to-end delay measurements in internet," in *Proc. Passive Active Meas. Workshop*, Fort Collins, CO, Mar. 2002, pp. 26–33.

[47] S. M. Kay, *Fundamentals of statistical signal processing. Estimation theory.* Prentice-Hall, 1993.

[48] A. Ahmad, D. Zennaro, E. Serpedin, and L. Vangelista, "A Factor Graph Approach to Clock Offset Estimation in Wireless Sensor Networks," *submitted to Information Theory, IEEE Transactions on.*

[49] M. Wainwright, T. Jaakkola, and A. Willsky, "A new class of upper bounds on the log partition function," *Information Theory, IEEE Transactions on*, vol. 51, no. 7, pp. 2313 – 2335, Jul. 2005.

[50] M. J. Wainwright and M. I. Jordan, "Graphical Models, Exponential Families, and Variational Inference," *Found. Trends Mach. Learn.*, vol. 1, pp. 1–305, Jan. 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1498840.1498841

[51] D. Chapman and H. Robbins, "Minimum variance estimation without regularity assumptions," *Annals of Math. Statistics*, vol. 22, no. 4, pp. 581–586, Dec. 1951.

[52] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2003.

[53] A. Ahmad, D. Zennaro, E. Serpedin, and L. Vangelista, "Time-varying clock offset estimation in two-way timing message exchange in wireless sensor networks using factor graphs," in *accepted at Acoustics, Speech and Signal Processing, 2012. ICASSP 2012. IEEE International Conference on*, 2012.

[54] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *Information Theory, IEEE Transactions on*, vol. 47, no. 2, pp. 498 –519, Feb. 2001.

[55] J. Dauwels and H.-A. Loeliger, "Phase estimation by message passing," in *Communications, 2004 IEEE International Conference on*, vol. 1, Jun. 2004, pp. 523 – 527 Vol.1.

[56] H. V. Trees and K. Bell, *Bayesian Bounds for Parameter Estimation and Nonlinear Filtering/Tracking*. Wiley, 2007.

[57] P. Tichavsky, C. Muravchik, and A. Nehorai, "Posterior Cramer-Rao bounds for discrete-time nonlinear filtering," *Signal Processing, IEEE Transactions on*, vol. 46, no. 5, pp. 1386 –1396, May 1998.

[58] 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. 215–233, Jan. 2007.

[59] G. Scutari, S. Barbarossa, and L. Pescosolido, "Distributed decision through self-synchronizing sensor networks in the presence of propagation delays and asymmetric channels," *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1667–1684, Apr. 2008.

[60] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *Automatic Control, IEEE Transactions on*, vol. 49, no. 9, pp. 1520–1533, Sept. 2004.

[61] S. Sardellitti, M. Giona, and S. Barbarossa, "Fast Distributed Average Consensus Algorithms Based on Advection-Diffusion Processes," *IEEE Transactions on Signal Processing*, vol. 58, no. 2, pp. 826 –842, Feb. 2010.

[62] R. Olfati-Saber and J. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, Dec. 2005, pp. 6698–6703.

[63] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *Information Theory, IEEE Transactions on*, vol. 52, no. 6, pp. 2508 – 2530, Jun. 2006.

[64] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast Gossip Algorithms for Consensus," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2748 –2761, Jul. 2009.

[65] D. P. Bersekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Athena Scientific, 1997.

[66] I. Schizas, A. Ribeiro, and G. Giannakis, "Consensus in ad hoc WSNs with noisy links - part I: Distributed estimation of deterministic signals," *Signal Processing, IEEE Transactions on*, vol. 56, no. 1, pp. 350–364, Jan. 2008.

[67] I. D. Schizas, G. B. Giannakis, S. I. Roumeliotis, and A. Ribeiro, "Consensus in Ad Hoc WSNs With Noisy Links - Part II: Distributed Estimation and Smoothing of Random Signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1650 –1666, Apr. 2008.

[68] H. Zhu, G. B. Giannakis, and A. Cano, "Distributed In-Network Channel Decoding," *IEEE Transactions on Signal Processing.*, vol. 57, no. 10, pp. 3970–3983, Oct. 2009.

[69] S.-J. Kim, E. Dall'Anese, and G. B. Giannakis, "Cooperative Spectrum Sensing for Cognitive Radios Using Kriged Kalman Filtering," *IEEE Journal on Selected Topics in Signal Processing*, vol. 5, no. 1, pp. 24 –36, Feb. 2011.

[70] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing Markov chain on a graph," *SIAM Rev.*, vol. 46, no. 4, pp. 667 – 689, 2004.

[71] E. Kokiopoulou and P. Frossard, "Polynomial Filtering for Fast Convergence in Distributed Consensus," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 342 –354, Jan. 2009.

[72] M. Cao, D. A. Spielman, and E. M. Yeh, "Accelerated gossip algorithms for distributed computation," in *44th Annual Allerton Conference on Communication, Control, and Computation*, Sept. 2006.

[73] B. Oreshkin, M. Coates, and M. Rabbat, "Optimization and analysis of distributed averaging with short node memory," *IEEE Transactions on Signal Processing*, vol. 58, no. 5, pp. 2850 –2865, May 2010.

[74] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65 – 78, Sept. 2004.

[75] D. Jakovetic and, J. Xavier, and J. Moura, "Weight Optimization for Consensus Algorithms With Correlated Switching Topology," *IEEE Transactions on Signal Processing*, vol. 58, no. 7, pp. 3788 –3801, Jul. 2010.

[76] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 5, Dec. 2003, pp. 4997 – 5002.

[77] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of Parallel and Distributed Computing*, vol. 67, no. 1, pp. 33 – 46, 2007.

[78] T. Erseghe, D. Zennaro, E. Dall'Anese, and L. Vangelista, "Fast Consensus by the Alternating Direction Multipliers Method," *Signal Processing, IEEE Transactions on*, vol. 59, no. 11, pp. 5523 –5537, Nov. 2011.

[79] F. R. Gantmacher, *The Theory of Matrices, Vol. 2.* Chelsea Publishing Company, 2000.

[80] J. S. Rosenthal, "Convergence rates of markov chains," *SIAM Review*, vol. 37, pp. 387–405, 1995.

[81] S. Kar and J. Moura, "Distributed Consensus Algorithms in Sensor Networks With Imperfect Communication: Link Failures and Channel Noise," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 355 –369, Jan. 2009.

[82] R. Horn and C. Johnson, *Matrix analysis*. Cambridge University Press, 1985.

[83] D. Zennaro, E. Dall'Anese, T. Erseghe, and L. Vangelista, "Fast clock synchronization in wireless sensor networks via ADMM-based consensus," in *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2011 International Symposium on*, May 2011, pp. 148 –153.

[84] W. Lindsey, F. Ghazvinian, W. Hagmann, and K. Dessouky, "Network synchronization," *Proceedings of the IEEE*, vol. 73, no. 10, pp. 1445–1467, Oct. 1985.

[85] T. Schmidl and D. Cox, "Low-overhead, low-complexity [burst] synchronization for ofdm," in *Communications, IEEE International Conference on*, vol. 3, Jun. 1996, pp. 1301 –1306.

[86] D. Zennaro, S. Tomasin, and L. Vangelista, "Uplink Cell Selection for Cooperative Multi-Cell Networks with Hybrid ARQ," in *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference*, Dec. 2010, pp. 1 –5.

[87] ——, "Base Station Selection in Uplink Macro Diversity Cellular Systems with Hybrid ARQ," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 6, pp. 1249 –1259, Jun. 2011.

[88] S. Chatzinotas, M. Imran, and R. Hoshyar, "On the multicell processing capacity of the cellular MIMO uplink channel in correlated rayleigh fading environment," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 7, pp. 3704 –3715, Jul. 2009.

[89] S. W. Peters, A. Y. Panah, K. T. Truong, and R. W. Heath, "Relay architectures for 3GPP LTE-advanced," *EURASIP J. Wirel. Commun. Netw.*, vol. 2009, pp. 1:1–1:14, Mar. 2009. [Online]. Available: http://dx.doi.org/10.1155/2009/618787

[90] L. Venturino, N. Prasad, and X. Wang, "Coordinated Scheduling and Power Allocation in Downlink Multicell OFDMA Networks," *Vehicular Technology, IEEE Transactions on*, vol. 58, no. 6, pp. 2835 –2848, Jul. 2009.

[91] H. Hu, Y. Zhang, and J. Luo, *Distributed Antenna Systems: Open Architecture for Future Wireless Communications*, 2007.

[92] A. Sanderovich, O. Somekh, and S. Shamai, "Uplink Macro Diversity with Limited Backhaul Capacity," in *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, Jun. 2007, pp. 11 –15.

[93] A. Sanderovich, O. Somekh, H. Poor, and S. Shamai, "Uplink Macro Diversity of Limited Backhaul Cellular Network," *Information Theory, IEEE Transactions on*, vol. 55, no. 8, pp. 3457 –3478, Aug. 2009.

[94] P. Marsch and G. Fettweis, "A Framework for Optimizing the Uplink Performance of Distributed Antenna Systems under a Constrained Backhaul," in *Communications, 2007. ICC '07. IEEE International Conference on*, Jun. 2007, pp. 975 –979.

[95] C. Hoymann, L. Falconetti, and R. Gupta, "Distributed Uplink Signal Processing of Cooperating Base Stations Based on IQ Sample Exchange," in *Communications, 2009. ICC '09. IEEE International Conference on*, Jun. 2009, pp. 1 –5.

[96] L. Falconetti, C. Hoymann, and R. Gupta, "Distributed Uplink Macro Diversity for Cooperating Base Stations," in *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on*, Jun. 2009, pp. 1 –5.

[97] O. Somekh, B. Zaidel, and S. Shamai, "Spectral Efficiency of Joint Multiple Cell-Site Processors for Randomly Spread DS-CDMA Systems," *Information Theory, IEEE Transactions on*, vol. 53, no. 7, pp. 2625 –2637, Jul. 2007.

[98] M. Kamoun and L. Mazet, "Base-station selection in cooperative single frequency cellular network," in *Signal Processing Advances in Wireless Communications, 2007. SPAWC 2007. IEEE 8th Workshop on*, Jun. 2007, pp. 1 –5.

[99] Y. Wang, X. Shen, and P. Zhang, "Uplink performance analysis in group cell systems," in *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, vol. 4, Jun. 2005, pp. 2405 – 2409.

[100] A. Del Coso and S. Simoens, "Distributed compression for MIMO coordinated networks with a backhaul constraint," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 9, pp. 4698 –4709, Sept. 2009.

[101] A. Papadogiannis, D. Gesbert, and E. Hardouin, "A Dynamic Clustering Approach in Wireless Networks with Multi-Cell Cooperative Processing," in *Communications, 2008. ICC '08. IEEE International Conference on*, May 2008, pp. 4033 –4037.

[102] S. Kiani and D. Gesbert, "Optimal and Distributed Scheduling for Multicell Capacity Maximization," *Wireless Communications, IEEE Transactions on*, vol. 7, no. 1, pp. 288 –297, Jan. 2008.

[103] I. Stanojev, O. Simeone, Y. Bar-Ness, and C. You, "Performance of multi-relay collaborative hybrid-ARQ protocols over fading channels," *Communications Letters, IEEE*, vol. 10, no. 7, pp. 522 –524, Jul. 2006.

[104] P. Wu and N. Jindal, "Coding Versus ARQ in Fading Channels: How Reliable Should the PHY Be?" in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, Dec. 2009, pp. 1 –6.

[105] F. Calabrese, M. Anas, C. Rosa, P. Mogensen, and K. Pedersen, "Performance of a Radio Resource Allocation Algorithm for UTRAN LTE Uplink," in *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, Apr. 2007, pp. 2895 –2899.

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

[107] A. Prudnikov, Y. A. Brychkov, and O. Marichev, *Integrals and Series, vol. 1.* Gordon and Breach Science Publishers, 1986.

[108] S. Amari and R. Misra, "Closed-form expressions for distribution of sum of exponential random variables," *Reliability, IEEE Transactions on*, vol. 46, no. 4, pp. 519 –522, Dec. 1997.