Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Methods and applications in networked control
Feedback control design for quantum systems

Ph.D. candidate
Saverio Bolognani

Advisor
prof. Sandro Zampieri

Ph.D. School in
Information Engineering
2011

Summary

Two quite different topics are covered in the thesis.

Methods and Applications in Networked Control

The first topic is networked control, identification, and optimization, with particular interest in the design of individual control laws for multi-agent networked systems. These systems consist in a large, sometimes unknown and time-varying, number of agents. These agents can communicate, they interact with an underlying physical system by sensing and actuating it, and none of them has a complete knowledge of the system state and parameters. The main challenges in designing control, identification, and optimization algorithms for these systems are the need for scalability, the constraints in the communication capabilities of the agents, the need for robustness in case of agent failure, and the adaptivity to changes in the system (node appearance and disappearance, communication failures, change in the agents' placement, etc.).

Some mathematical tools and methods from the literature are reviewed, and their application to the problem of networked control are explored. A series of original mathematical methods and algorithms have then be derived:

- analysis results on the stability and scalability of consensus algorithms for unstable dynamics;
- distributed parametric identification via least-square estimation;
- distributed quasi-Newton methods for convex optimization;
- randomized iterative methods for quadratic, linearly constrained optimization.

These methods are presented via their application to some motivating example of networked control systems. All of them are also of great interest *per se*, and can be considered challenging open problems in the field of networked control. They are the following:

- leaderless, distributed, clock synchronization in networks of agents with extremely limited computational, communication, and energy resources;
- distributed, online estimation of the wireless channel parameters for localization of mobile nodes via triangulation;
- management architectures and distributed networked control laws for reactive power compensation (and other ancillary services) in the so called smart grids, with a particular focus on microgrids at the power distribution level.

These open problems will be explored both as a motivating testbed for the proposed methods and as an outlook for future directions of investigation.

Feedback Control Design for Quantum Systems

The second topic is the problem of control of quantum dynamical systems, via the design of stabilizing feedback control law when quantum measurements are available.

Discrete time models for quantum dynamics are reviewed, with a particular attention to those control problems that are typical key tasks in quantum information processing:

- preparation of states of maximal information;
- engineering of protected realization of quantum information, i.e. the realization of information encodings that preserve the fragile quantum states from the action of noise.

These tasks can be casted into the more general problem of engineering stable quantum dynamical subspaces.

In this part of the thesis the asymptotic behavior of discrete-time, Markovian quantum systems with respect to a subspace of interest, is analyzed. The results of this analysis, based on a Ljapunov approach, provide necessary and sufficient conditions on the dynamical model that ensure global asymptotic stability of a certain quantum subspace.

It is then introduced a control scheme that allows modifying the underlying dynamics by indirectly measuring it and by applying unitary control actions conditioned on the outcome of the measurement. For this discrete-time feedback control scheme, an original design algorithm capable of stabilizing a target subspace is introduced. It is guaranteed that if the control problem is feasible, then the algorithm returns an effective control choice. In order to prove this result, a new technical tool is derived, namely a canonical QR matrix decomposition, which is also used to establish the control scheme potential for the simulation of open-system dynamics.

Sommario

Nella tesi vengono trattati due argomenti distinti.

Metodi e applicazioni nel controllo distribuito

Il primo argomento trattato riguarda il controllo, l'identificazione e l'ottimizzazione distribuiti (*networked*), con particolare interesse per la progettazione di leggi di controllo per sistemi multiagenti distribuiti. Questi sistemi consistono di un numero di agenti elevato, talvolta sconosciuto e variabile nel tempo. Questi agenti sono in grado di comunicare, di interagire con il sistema fisico nel quale si trovano tramite operazioni di misura e attuazione, e nessuno di essi ha una conoscenza completa dello stato e dei parametri del sistema. Le sfide maggiori per la progettazione di algoritmi di controllo, identificazione e ottimizzazione per questi sistemi sono i requisiti di scalabilità, i vincoli nelle comunicazioni tra gli agenti, i requisiti di robustezza nei confronti di guasti degli agenti, e la capacità di adattarsi a modifiche del sistema (apparizione e scomparsa di nodi, errori di comunicazione, spostamento e riconfigurazione degli agenti, ecc.).

Sono stati riportati alcuni strumenti e metodi matematici disponibili in letteratura, illustrando la loro applicazione al problema del controllo distribuito. È stata poi ricavata un serie di metodi e algoritmi originali:

- risultati sull'analisi di stabilità e di scalabilità degli algoritmi di consenso per sistemi dinamici instabili;
- identificazione parametrica distribuita tramite stima ai minimi quadrati;
- metodi quasi-Newton distribuiti per problemi di ottimizzazione convessa;
- metodi iterativi randomizzati per l'ottimizzazione quadratica con vincoli lineari.

Questi metodi vengono presentati illustrandone l'applicazione ad alcuni esempi di sistemi di controllo distribuiti. Questi esempi di applicazione sono interessanti di per sé, in quando possono essere considerati problemi tuttora aperti nel campo del controllo distribuito. Essi sono:

- sincronizzazione temporale completamente decentralizzata in reti di agenti dotati di limitate capacità di calcolo, comunicazione e autonomia;
- stima online distribuita dei parametri del canale di comunicazione wireless per la localizzazione di nodi mobili tramite triangolazione;
- architetture e leggi di controllo distribuite per la compensazione della potenza reattiva nelle *smart grids*, ed in particolare nelle micro-reti di distribuzione.

Queste applicazioni vengono illustrati sia come banchi di prova per i metodi proposti, sia come possibili futuri sviluppi della ricerca in questo campo.

Controllo a retroazione per sistemi quantistici

Il secondo argomento riguarda il problema del controllo di sistemi dinamici quantistici tramite la progettazione di leggi di retroazione stabilizzanti, in presenza di misure quantistiche.

Innanzitutto vengono richiamati modelli a tempo discreto per sistemi quantistici, con particolare interesse per i problemi del controllo quantistico che sono di maggiore rilevanza per la teoria dell'informazione quantistica:

- preparazione di stati di massima informazione;
- protezione dell'informazione quantistica, ovvero realizzazione di codifiche dell'informazione che preservino gli stati quantistici dall'azione del rumore.

Questi compiti possono essere interpretati all'interno del problema più ampio di stabilizzazione di sottospazi nei sistemi quantistici.

In questa parte della tesi viene analizzato il comportamento asintotico di sistemi quantistici Markoviani a tempo discreto rispetto ad un sottospazio di interesse. Il risultato di questa analisi, basato su un approccio alla Ljapunov, fornisce condizioni necessarie e sufficienti sul modello dinamico per garantire la stabilità asintotica di un certo sottospazio.

Viene poi introdotto uno schema di controllo che permette di modificare la dinamica del sistema tramite misure indirette e tramite l'applicazione di azioni di controllo coerenti, condizionate dal risultato della misura. Viene proposto un algoritmo originale per la progettazione della legge di controllo capace di stabilizzare un dato sottospazio. È garantito che, se il problema di controllo ha soluzione, allora l'algoritmo fornisce una legge di controllo stabilizzante.

Per dimostrare questi risultati è stato necessario definire un nuovo strumento: una decomposizione QR canonica, che viene anche utilizzata per studiare le potenzialità dello schema di controllo per la simulazione di dinamiche diverse da quelle del sistema.

Contents

Part I

Methods and application in networked control

1 Introduction

An extremely synthetic definition of networked control systems (NCS) is provided in Special issue on technology of networked control systems (2007):

> "Networked control systems are control systems comprised of the system to be controlled and of actuators, sensors, and controllers, the operation of which is coordinated via a communication network. These systems are typically spatially distributed, may operate in an asynchronous manner, but have their operation coordinated to achieve desired overall objectives."

It is clear from this definition that the class of control problems that can be casted into this framework is extremely large.

An overview on the historical background of this field, on the available methodologies, and on the many applications of NCS, can be found in Special issue on sensor networks and applications (2003) and in Special issue on technology of networked control systems (2007). The very seminal works on these topics have been inspired by some examples of

multi-agent cooperation observed in nature, ranging from coordinated school of fish to the synchronized behavior of fireflies (see Strogatz 2004 for an inspiring lecture). The analysis of such behaviors in nature has soon motivated the design of similar algorithms for the control of large numbers of autonomous vehicles, robots, unmanned aircrafts, and underwater devices. The widespread and cheap availability of wireless communication and the falling cost of extremely small computing and communicating units has also driven the research towards networks made of a huge numbers of sensors for environmental monitoring, process control, and dispersed automation. Lately, extremely diverse examples of NCS have entered into the picture: social networks, financial and market economy, biotechnology devices.

It should be clear from this (non exhaustive) list of applications that NCS are better characterized by the issues and the challenges that they present, rather than by the specific scenario that they represent.

A list of these issues is reported hereafter, and will be referenced later in the thesis, when these issues are addressed. Then, in Chapter 2, a series of methodological tools that find frequent application in NCS is reviewed. These tools are presented with a consistent approach, notation and terminology, which will be adopted also in the rest of the thesis. Proper references to the existing literature are also provided in the same chapter. Chapter 3 contains the original contributions of this work. The proposed methods and solutions are presented via their application to specific scenarios.

The reason of this choice is twofold.

While these methods are quite general and they are not strictly application-specific, they are however motivated by some specific case studies, and they exploit some of their features. The opportunity of adapting the same methods to different applications will become evident in their presentation.

At the same time, the modeling of the specific applications is interesting *per se*, as the chosen applications are some of the most motivating and promising examples of NCS. *Wireless sensor networks* have already proved their effectiveness in many situations, their implementation has been translated into technological and engineering problems, and experimental testbed are widely available. On the other hand, the application of NCS methodologies to the *power distribution networks* is still novel and unexplored, but extremely promising; smart grids are likely to become a valuable testbed and a field of application for many of these methods, because of the extremely diverse issues that they present, because of their novelty, and because they require a holistic approach that embraces all field of ICT: communications, control, computer science, information theory, together of course with power electronics and electrical engineering.

1.1 Issues and challenges in NCS

Multi agent architectures

According to the definition given before, NCS are control systems in which the controller elements are not centralized but distributed over the entire system via parallel and distributed processing.

NCS are therefore agent-based system. Multi-agent systems comprise multiple intelligent entities (agents) working together to solve the complex problem, which is difficult or impossible to an individual agent.

The decomposition of the whole system into agents is generally inherent in the system itself, and it is not part of the design. The abstract concept of agent can correspond to extremely different entities: parts of a software, pieces of hardware, sensors and actuators, computers in a data network, customers, companies, people with their opinion, moving robots, vehicles, and so on. In all these scenarios the multi-agent aspect of the problem is critical.

Scalability

The rationale behind the design of most of NCS is that a large number of agents with basic computational, sensing, and actuating capabilities, if properly commanded, can achieve some complex global behavior. In this sense, the fact of having an arbitrary large number of agents is a feature that has to be necessarily exploited to achieved the desired result.

At the same time, this requires that scalability of the control laws and of the distributed algorithms is taken into consideration at the design stage: the computational effort (in terms of both memory space and computing time) must remain constant or almost constant for an arbitrary number of replicas of the same agent (i.e. for an arbitrary size of the system).

Distributed information

In the analysis and in the design of a NCS, one of the critical aspects of its multi-agent structure is understanding and deciding what information is available to each agent.

Indeed, both the systems structure (the number of agents, their configuration, the communication graph, the underlying physical system) and the system state are unknown in their full extent to any agent. Every agent has only a partial knowledge of this information, and there is no central unit that has the global view of the system. Moreover, the structure of the systems may change in time, due to external events, reconfiguration, node appearance and disappearance.

Communication constraints

Communication between agents in a NCS is constrained in many ways. Almost always, agents willing to coordinate their behavior and exchange data are forced to interact with a smaller subset of neighbors. In many cases, they also experience transmission noise, quantization, data rate constraints, delay and unreliable communication.

Interaction with an underlying physical system

The agents of a NCS are generally deployed in an environment, and interact with it by sensing or actuating. This interaction poses some challenges, in particular when the underlying environment is partially unknown and exhibits its own dynamics.

In some cases the presence of an underlying physical system governed by its own laws poses some extra challenges. Consider for example the problem of robot coordination when, because of the physics of the wireless transmission channel, the communication graph depends on the state of the system (the nodes' distances). In this case, like in many others, the underlying physical system couples different aspects (communication constraints and system state in this example) and makes the analysis and the design of a distributed control law much more complex.

In some other cases the underlying physical system can be exploited by the agents for *signaling*. The idea that controllers actuating and sensing the same system can use it to implement some exchange of information dates back to the Witsenhausen counterexample (see Mitter and Sahai 1999 for an interpretation in this sense). This possibility is still an open problem, especially in the case in which a large number of networked controllers are involved. In some cases, however, something similar has been achieved: by sensing locally some outputs of the underlying system, the agents of a NCS may be able to infer some functions of the whole system state. This is of course very application specific, but can be recognized in a remarkable number of examples.

Robustness to systems changes

When a system is constituted by a large number of agents, the design of a distributed algorithm or control law must take into account the possibility that some agents enter into the system, some disconnect, some are reconfigured. This is especially true when these agents are inexpensive and therefore unreliable.

Distributed algorithms designed for a NCS must guarantee that their performance degrades in a reasonable way when some agents misbehave. The global behavior of the system

has to be quite insensitive with respect to heterogeneity of its parts, faulty behaviors and also malicious attacks.

Sometimes guaranteeing this feature in a multi-agent system is easier than in a centralized setup. It is indeed one of the motivations for the transition towards NCS in some applications, for example in critical infrastructures where the reliability of the whole system cannot depend on few central devices, although very reliable.

Performances

The issue of performances of the algorithms implemented in NCS is quite different from the analysis of the performances in parallel computing (which has been object of wide research and share some similarities). While in parallel computing, the algorithms are decomposed into sub-algorithms for the different processing units to obtain improvements in the performances (speed and/or memory requirements), in NCS the multi-agent structure of the algorithms is given by the architecture of the system and cannot, in general, be avoided (it may also be part of the designer choice to accept a slight degradation of the performances when this guarantees extra robustness of the system).

Moreover, many algorithms in NCS are asynchronous and randomized. In these cases analyzing their worst case behavior is almost always too conservative and it underestimates the algorithm potentials. Probabilistic methods are then needed, together with proper models for the strategies of execution of the individual nodes' sub-algorithms.

2

Methods and mathematical tools

2.1 Multiagent architectures and communication models

Graphs

Let $\mathcal{V}$ be a set of *nodes* corresponding to the agents of a networked control system:

$$\mathcal{V} = \{v_1, v_2, \ldots, v_N\}, \quad |\mathcal{V}| = N.$$

A *graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, s(\cdot), t(\cdot))$ can be defined on the set of nodes $\mathcal{V}$, where

- $\mathcal{E}$ is the set of *edges*
- $s(\cdot) : \mathcal{E} \to \mathcal{V}$ is a function that maps every edge into its *source*
- $t(\cdot) : \mathcal{E} \to \mathcal{V}$ is a function that maps every edge into its *sink*.

In the case in which multiple edges (edges with the same source and the same sink) are not allowed, it is possible to replace the set of nodes with the set of node indices, obtaining $\mathcal{V} = \{1, \dots, N\}$, and the set of edges with a set of ordered pairs of nodes: $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. As the source and the sink of an edge (i, j) are implicitely defined as $s\big((i, j)\big) = i$ and $t\big((i, j)\big) = j$, a graph is completely determined by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is called *undirected* when

$$(i, j) \in \mathcal{E} \quad \Longrightarrow \quad (j, i) \in \mathcal{E}.$$

In some application it is however useful to consider a more general definition of an undirected graph, in which *hyperedges* $e \in \mathcal{E}$ are defined as subsets of the set of nodes $\mathcal{V}$. The set of nodes $\mathcal{V}$ and the set of hyperedges $\mathcal{E}$ constitute an *hypergraph*. Regular undirected graphs are hypergraphs with $|e| \leq 2$ for all $e \in \mathcal{E}$. To avoid extra notational burden, hypergraphs will not be considered in the rest of this section.

The following definitions will be adopted for the rest of the thesis.

Let the set of *out-neighbors* of the node $v \in \mathcal{V}$ be defined as

$$\mathcal{N}(v) = \{u \in \mathcal{V} \,|\, \exists e \in \mathcal{E}, s(e) = v, t(e) = u\} \subseteq \mathcal{V}$$

or equivalently, when nodes are identified by their indices,

$$\mathcal{N}(i) = \big\{j \in \mathcal{V} \,|\, (i, j) \in \mathcal{E}\big\} \subseteq \mathcal{V}.$$

Let instead $\overline{\mathcal{N}}(i)$ be the set of in-neighbors, defined as

$$\overline{\mathcal{N}}(i) = \big\{j \in \mathcal{V} \,|\, (j, i) \in \mathcal{E}\big\} \subseteq \mathcal{V}.$$

Let the *out-degree* of a node v be defined as $d(v) = |\mathcal{N}(v)|$, and its *in-degree* as $\bar{d}(v) = |\overline{\mathcal{N}}(v)|$. In the sequel the out-neighbors will be just called *neighbors* and the out-degree will be called *degree*.

Let the *adjacency matrix* $A \in \{0, 1\}^{N \times N}$ of a graph be defined by its elements:

$$[A]_{ij} = \begin{cases} 1 & \text{if } (j, i) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$$

Let the *Laplacian matrix* $L \in \mathbb{R}^{N\times N}$ of a graph be defined by its elements:

$$[L]_{ij} = \begin{cases} -1, & \text{if } i \in \mathcal{N}(j) \\ d(i) & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

It is straightforward to show that $L = D - A$, where $D = \text{diag}(d(1), \ldots, d(N))$.

According to the definition, all row-sums of L are zero, and therefore L always has a zero eigenvalue $\lambda_1 = 0$. The corresponding eigenvector is $\mathbf{1} = [1 \ldots 1]^T$. Moreover, if L is symmetric (undirected graph), its eigenvalues satisfy

$$0 = \lambda_1 \leq \lambda_1 \leq \cdots \leq \lambda_N \leq 2 \max_{i\in\mathcal{V}} d(i),$$

while if L is not symmetric, the real part of its eigenvalues satisfy the same condition.

In a directed graph, a *path* of length ℓ from nodes u to v is a sequence of edges $\{e_1, \ldots, e_\ell\}$ such that $t(e_i) = s(e_{i+1})$ for $i = 1, \ldots, \ell - 1$, $s(e_1) = u$, $t(e_\ell) = v$. The graph $\mathcal{G}$ is *connected* if every pair of nodes u, v is connected by a path. A directed graph is *weakly connected* if the undirected graph obtained by adding the inverse edge (j, i) every time that (i, j) exists, is connected. For an undirected, connected graph $\mathcal{G}$, the second smallest eigenvalue of the Laplacian is strictly positive ($\lambda_2 > 0$). For this reason λ_2 is called *algebraic connectivity*.

Unless stated differently, it is assumed in the following that any graph $\mathcal{G}$ includes all self-arcs, i.e. $(i, i) \in \mathcal{E}, \forall i \in \mathcal{N}$.

A graph is *rooted in* k if there exists a path in the graph from node $k \in \mathcal{V}$ to any other node in $\mathcal{V}$, and *strongly rooted in* k if node k is directly connected to all other nodes, i.e. $(k, j) \in \mathcal{E}, \forall j \in \mathcal{V}$. Clearly a strongly connected graph implies that it is also a rooted graph for any node. A graph which has at least one root is called *rooted*; finally, a *strongly rooted* graph is a graph which has at least one vertex at which it is strongly rooted.

The *concatenation* of two graphs $\mathcal{G}_1 = (\mathcal{V}, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}, \mathcal{E}_2)$ is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}) = \mathcal{G}_2 \circ \mathcal{G}_1$ such that $(i, j) \in \mathcal{E}$ if there exists $k \in \mathcal{V}$ such that $(k, j) \in \mathcal{E}_1, (i, k) \in \mathcal{E}_2$. Similarly, the *union* of two graphs is a graph $\mathcal{G} = \mathcal{G}_1 \cup \mathcal{G}_2$ for which $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_1$. Clearly $\mathcal{G}_1 \cup \mathcal{G}_2 = \mathcal{G}_2 \cup \mathcal{G}_1$ and $\mathcal{G}_2 \circ \mathcal{G}_1 \neq \mathcal{G}_1 \circ \mathcal{G}_2$.

In many cases it is convenient to introduce *weights* on the edges, defined as a scalar real-valued function $w : \mathcal{E} \to \mathbb{R}$. From this definition, the concepts of *weighted adjacency*

matrix A^w, and *weighted Laplacian* L^w are derived, where

$$[A^w]_{ij} = \begin{cases} w\big((j,i)\big) & \text{if } (j,i) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}$$

and

$$[L^w]_{ij} = \begin{cases} -w\big((j,i)\big), & \text{if } i \in \mathcal{N}(j) \\ \sum_{k \in \mathcal{N}(i)} w\big((i,k)\big) & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

Communication models

In many NCS applications, graphs are defined on the set of agents to model the communication constraints in the systems. In these cases, an edge e belongs to $\mathcal{E}$ if the agent $s(e)$ is allowed to send some information to the node $t(e)$ via the given communication channel.

Different communication technologies correspond to very different graphs. Just to mention some examples, they include *complete graphs* when every node can communicate with any other node (for example in multihop data networks), *regular graphs* when agents are devices arranged in some predefined structure (for example multiprocessor machines), *random geometric graphs* in wireless sensor networks (see Figure 2.1 for a brief definition), *trees* when communication happens on the electrical power grid via power-line communication.

A communication graph can also be constructed from experimental data, as shown in Section 3.2, where edges between wireless sensors represent wireless links that exhibited a sufficiently high probability of success in a series of transmission tests (or, conversely, a packet drop probability lower than a given threshold).

This simple model for agent-to-agent communication does not take into account many of the phenomena that characterize the communication channels that are available in real-life NCS: communication noise, packet losses and data errors, quantization, just to mention few. In some specific cases (for example when agents move in the space and communication is limited to a given maximum range) the communication graph can also depend on the nodes' state. The analysis of these phenomena has been done in the literature for some specific algorithms and applications (see Section 2.2 for the relative discussion in the case of consensus algorithms).

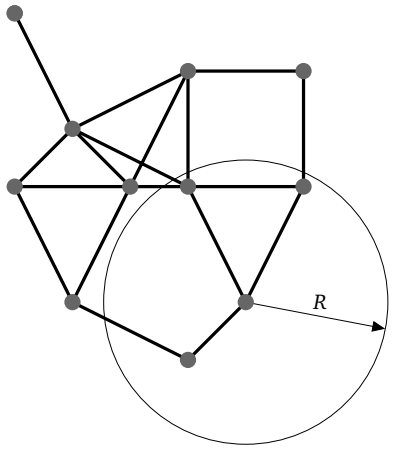


Figure 2.1: Example of a random geometric graph. Nodes are deployed in the space (in this case in $\mathbb{R}^2$) according to a given probability distribution. Pair of nodes whose distance is less than a given radius R are connected by an edge.

Communication and algorithm execution strategies

The access of the agents to the communication media can be regulated according to different strategies. Generally there is a correspondence between the action of communicating and the action of processing the acquired data, and therefore the following communication strategies also induce different iterative forms of the algorithms. In many cases, the agents involved in the communication are also required to take action, perform some sensing, actuate themselves or the system; just to mention some examples, in Bullo, Cortés, and Martínez (2009); Frasca, Carli, and Bullo (2009a) the agents (moving robots) that perform a communication are typically required to move in a new computed position, and in Bolognani and Zampieri (2011) the agents (power inverters) are also required to modify the amount of power injected in the power grid.

The following strategies have then to be intended as strategies for the activation of the agents, and not only for the communication:

- in *gossip communication* one agent receives the information from another agent, while the rest of the system in unaware of the exchange of data; if the graph is undirected, it is also possible to enforce that the exchange of information is always bidirectional (symmetric gossip); symmetry in the data transmission can be required by some algorithms to guarantee some particular features (see for example average consensus in Section 2.2); note however that ensuring symmetry of operation when the channel suffers from non zero packet drop rate or error probability is a challenging – if not impossible – goal to achieve (see the *Two Generals' Problem*);

- in *broadcast communication*, agent v sends the same information to all its neighbors $\mathcal{N}(v)$, receiving nothing in response; according to the communication technology, this strategy can require more resources (energy, bandwidth, time) than the gossip strategy, because a larger number of agents is involved; however this is not always true: one notable example is wireless communication, which is inherently a broadcast communication as the same information in received by all the nodes in the communication range of the transmitting node.

Agents can be triggered to communicate according to different modalities.

In *synchronous algorithms*, all agents communicate at the same time. This approach usually returns global behaviors that are easier to analyze, but can be unpractical to achieve, as it requires a great level of coordination among the agents and may not be acceptable if the communication channel has to be shared among agents.

In *asynchronous algorithms*, on the other hand, agents are triggered one at a time. The resulting sequence of nodes can be *deterministic* or *randomized*. Building a deterministic sequence on the set of agents can require some supervision in the system, but it can simplify the algorithm analysis. The study of synchronous algorithms and asynchronous deterministic algorithms inherits many methodologies from the literature on parallel computation, as in Bertsekas and Tsitsiklis (1997). In randomized algorithms, instead, the sequence of activation of the nodes is the realization of a random process. While many stochastic processes can be adopted, it is interesting to consider the case in which the sequence is obtained as the result of individual random processes on each agent.

As an example, consider the remarkable cases of a *Poisson process* in continuous time and of a *Bernoulli process* in discrete time.

In continuous time, suppose that each agent, independently from the others, waits for an exponentially distributed amount of time between consecutive communications, with average waiting time of $1/\lambda$. The number of times that a single agent has executed its local algorithm up to a given time is described by a Poisson distribution of intensity λ; therefore in the entire system nodes are triggered according to a Poisson distribution of intensity $N\lambda$, exhibiting also the property that the event of simultaneous communication of two different nodes has zero probability (which is highly desirable in the case in which nodes have to share a communication channel).

The analogous strategy in discrete time consists in having all the agents "flipping a coin" at each time instant, therefore deciding whether to communicate or not with a fixed probability p. At a system level, pN agents will be transmitting, in average, at every time instant.

Note that these two randomized strategy have the remarkable feature that they need no

coordination to be implemented. The design and analysis of algorithms based on randomized asynchronous operation is generally more challenging. Their study is sometimes simplified by assuming that a $T \in \mathbb{R}$ exists such that every agent is triggered at least once in a fixed with time window $[t, t+T]$, for all $t \in \mathbb{R}$. These algorithms are called *pseudo-synchronous* in Bertsekas and Tsitsiklis (1997).

2.2 Distributed information and consensus

System state

Let the state of each agent v in a NCS be defined by $x_v \in \mathbb{R}^p$. x_v can also be intended as the value of a function $x : \mathcal{V} \to \mathbb{R}^p$ evaluated at node v of the graph.

By indexing nodes as $\{1, \dots, N\}$, a scalar function ($p = 1$) defined on the graph nodes becomes a function defined on the first N natural numbers, and can therefore be interpreted as a vector $x \in \mathbb{R}^N$. In the more general case of a multidimensional state ($p \neq 1$), the system state $x \in \mathbb{R}^{pN}$ is obtained by stacking all the individual agents' states.

In this notation for the system state, the correspondance between nodes and their state is lost: when designing any control law for the individual agent i, one must consider the fact that only a subset of the system state components can be accessed by the node, namely the states of its in-neighbors $\overline{\mathcal{N}}(i)$.

Consider for example the following linear update equation for the system state:

$$x(t+1) = Q(t)x(t) \tag{2.1}$$

where $x \in \mathbb{R}^N$ and $Q \in \mathbb{R}^{N \times N}$. Let $\mathcal{G}_Q = (\mathcal{V}, \mathcal{E}_Q)$ be the graph associated with Q, where $\mathcal{E}_Q = \{(j,i) \,|\, [Q]_{ij} > 0\}$. A matrix Q is *compatible* with the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, denoted with $Q \sim \mathcal{G}$, if its associated graph $\mathcal{G}_Q = (\mathcal{V}, \mathcal{E}_Q)$ is such that $\mathcal{E}_Q \subseteq \mathcal{E}$.

Consensus algorithms

The *consensus problem* can be briefly defined as the problem of reaching an agreement among nodes in a multiagent system regarding a certain quantity of interest that depends on the state of all agents.

A *consensus algorithm* is an interaction rule that specifies the information exchange between an agent $v \in \mathcal{V}$ and its neighbors $\mathcal{N}(v)$ in the communication graph.

The theoretical framework for posing and solving consensus problems for networked dynamic systems was introduced in Fax and Murray (2004) and Olfati-Saber and Murray (2004).

Consensus algorithms have been successfully applied to NCS (Olfati-Saber, Fax, and Murray, 2007), coordinated robotics (Jadbabaie, Lin, and Morse, 2003), time synchronization (Solis, Borkar, and Kumar, 2006; Schenato and Gamba, 2007), and distributed estimation (Spanos, Olfati-Saber, and Murray, 2005; Speranzon, Fischione, Johansson, and Sangiovanni-Vincentelli, 2008). While most of the consensus algorithms assume Euclidean spaces, they have also been defined on manifolds, as in Sarlette and Sepulchre (2009), al-

lowing their application to specific fields that cannot be modeled otherwise (e.g. orientation control as in Sarlette, Sepulchre, and Leonard 2009).

Consensus has also been studied in the case in which communication between agents is subject to some non-idealities, like quantization and time delays (Olfati-Saber and Murray, 2004; Moreau, 2005; Carli, Fagnani, Speranzon, and Zampieri, 2008a; Frasca, Carli, Fagnani, and Zampieri, 2009b).

These algorithms share some similarities with the asynchronous asymptotic agreement problem for distributed decision making systems and parallel computing as defined in Bertsekas and Tsitsiklis (1997). They also resemble some of the distributed asynchronous iterative algorithms for solving, or at least approximately solving in least-square sense, linear equations $Az = b$, where $z \in \mathbb{R}^r$ is the vector of unknown parameters, and the matrix $A \in \mathbb{R}^{s\times r}$ and the vector $b \in \mathbb{R}^s$ are known (Bertsekas and Tsitsiklis, 1997; Strikwerda, 2002; Frommer and Szyld, 2000). These latter algorithms are particularly suitable for solving sparse equations where the number of equations is equal to the number of unknowns parameters, i.e. when A is square and has many zero-entries. Although, in principle these algorithms can be applied also to a non-square matrix A, since the least square solution of $Az = b$ is equivalent to $A^T Az = A^T b$, it might happen that $A^T A \in \mathbb{R}^{r\times r}$, which is now square, is dense even if A is sparse, thus loosing many of their advantages. Moreover, these algorithms naturally lead to parallel implementation since each element of vector z can be computed by a distinct agent. However, in many applications, such as in sensor networks, the number of physical agents, i.e. the sensor nodes, coincides with the size s of the vector b rather than the size r of the vector z on which they have to agree, thus making the distributed implementation not feasible. Differently, consensus algorithms can be effective also in these contexts.

Some consensus algorithms are recalled hereafter, together with some definitions.

Definition 2.2.1 (Stochastic and doubly stochastic matrices)**.** $Q \in \mathbb{R}^{N\times N}$ is a *stochastic matrix* if $[Q]_{ij} \geq 0$ and $\sum_{j=1}^{N} [Q]_{ij} = 1$, $\forall j$, i.e. each row sums to unity. A stochastic matrix Q is said *doubly stochastic* if also $\sum_{i=1}^{N} [Q]_{ij} = 1$, i.e. each column sums to unity. Clearly if a stochastic matrix is symmetric then it is also doubly stochastic.

Definition 2.2.2 (Consensus problem)**.** Consider the algorithm (2.1). $Q(t)$ solves the *consensus problem* if $\lim_{t\to\infty} x_i(t) = \alpha$, $\forall i = 1,\dots,N$, where $x_i(t)$ is the i-th component of the vector $x(t)$, and where α is a function of the system initial state $x(0)$. $Q(t)$ solves the *average consensus problem* if $\alpha = \frac{1}{N}\sum_{i=1}^{N} x_i(0)$. If $Q(t)$ is a random variable, (2.1) is said to solve the *probabilistic (average) consensus problem* if the limit above exists almost surely.

This definitions includes a wide class of consensus strategies: strategies with a time–invariant matrix $Q(t) = Q$, deterministic time-varying strategies $Q(t)$, and randomized strategies where $Q(t)$ is drawn from some distributions on a set of stochastic matrices.

The next theorems describe some sufficient conditions which guarantee deterministic and probabilistic (average) consensus.

Theorem 2.2.3. *Consider a sequence of constant matrices $Q(t) = Q$. If the graph $\mathcal{G}_Q$ is rooted, then Q solves the consensus problem. If in addition Q is doubly stochastic, then $\mathcal{G}_Q$ is strongly connected and Q solves the average consensus problem. Moreover, the convergence rate in both cases is exponential and it is given by second largest eigenvalue in absolute value of the matrix Q.*

Theorem 2.2.4. *Consider a deterministic sequence of stochastic matrices $\{Q(t)\}_{t\geq 0}$ and the corresponding associated graphs $\mathcal{G}(t) = \mathcal{G}_{Q(t)}$. Suppose there exists a finite positive integer number T such that the graphs $\mathcal{G}^{'}(\cdot)$ obtained from the composition of the graphs $\mathcal{G}(\cdot)$ in the following way: $\mathcal{G}^{'}(\tau) = \mathcal{G}(\tau \cdot T) \circ \mathcal{G}(\tau \cdot T + 1) \circ \ldots \circ \mathcal{G}(\tau \cdot T + T - 1)$ with $\tau = 0, 1, \ldots$ are all rooted. Then the sequence $Q(t)$ solves the consensus problem. If the matrices $Q(t)$ are all doubly stochastic, then they solve the average consensus problem.*

Theorem 2.2.5. *Consider a random i.i.d. sequence of stochastic matrices $\{Q(t)\}_{t\geq 0}$ drawn according to some distribution from the set of all stochastic matrices, and the stochastic matrix $\overline{Q} = \mathbb{E}[Q(t)]$. If $\mathcal{G}_{\overline{Q}}$ is rooted, and the probability $P\left([Q]_i i > 0\right)$ is greater than zero for any i, then the sequence $Q(t)$ solves the probabilistic consensus problem. If in addition $Q(t)$ are all doubly stochastic, then they solve the probabilistic average consensus problem.*

The first theorem is concerned with constant consensus matrix and shows how convergence conditions can be reframed as a graph problem which is easy to verify (Costa, Fragoso, and Marques, 2004). Xiao and Boyd (2004) proposed an algorithm based on convex optimizations that, given a symmetric graph, finds a symmetric doubly stochastic matrix Q compatible with the graph which maximize the rate of convergence. The second theorem focuses on deterministic time-varying consensus algorithms and shows that it is not necessary for the communication graph to be connected at any iteration but over a fixed time window (Cao, Morse, and Anderson, 2008; Moreau, 2005). The last theorem addresses the consensus problem in a probabilistic context that arises from randomized communication strategies, as in Boyd, Ghosh, Prabhakar, and Shah (2006); Fagnani and Zampieri (2008), or networks subject to random external disturbances, such as link or node failure, as in Fagnani and Zampieri (2009).

According to the communication strategies presented in the previous section, different consensus algorithms can be implemented, resulting in different sequences $\{Q(t)\}_{t\geq 0}$.

In the broadcast strategy the consensus matrix Q_i^B (the superscript "B" stands for broad-

cast), when a node i transmits, is given by:

$$\left[Q_i^B\right]_{mn} = \begin{cases} 1 & \text{if } m = n \notin \mathcal{N}(i)\backslash\{i\} \\ 1-w & \text{if } m = n \in \mathcal{N}(i)\backslash\{i\} \\ w & \text{if } m \in \mathcal{N}(i)\backslash\{i\},\ n = i \\ 0 & \text{otherwise} \end{cases}$$

where $w \in (0,1)$ is a tuning parameter and often $w = \frac{1}{2}$. In the symmetric gossip, when the edge (i,j) is selected, the consensus matrix Q_{ij}^G (the superscript "G" stands for gossip) is given by:

$$\left[Q_{ij}^G\right]_{mn} = \begin{cases} 1 & \text{if } m = n \neq j \text{ and } m = n \neq i \\ 1-w & \text{if } m = n = j \text{ or } m = n = i \\ w & \text{if } (m,n) = (i,j) \text{ or } (m,n) = (j,i) \\ 0 & \text{otherwise} \end{cases}$$

The consensus matrices defined above are based on the assumption that there is no link failure during the communication. In many cases, however, this is not true. When a link (i,j) fails in broadcast communication, the matrix Q_i^B needs to be modified with $[Q_i^B]_{jj} = 1, [Q_i^B]_{ji} = 0$. Instead, when it happens in symmetric gossip, there is no communication at all and then no update is performed, i.e. $Q_{ij}^G = I$ is the identity matrix. Note that in broadcast $Q_{ij}^B = I$ if all links fail. If the reliability of the communication is modeled via an independent, identically distributed, *transmission success probability* c,then it is possible to show that the expected consensus matrix $\overline{Q^B} = \mathbb{E}\left[Q^B(t)\right]$ generated for the broadcast strategy is given by:

$$\left[\overline{Q}^B\right]_{mn} = \begin{cases} 1 - \frac{c \cdot w \cdot d(n)}{N} & \text{if } m = n \\ \frac{c \cdot w}{N} & \text{if } m \in \mathcal{N}(n) \\ 0 & \text{otherwise} \end{cases}$$

Note that if the graph is undirected, $\overline{Q}^B = (\overline{Q}^B)^T$ is symmetric and hence doubly stochastic, although the matrices Q_i^B are never symmetric. Moreover $\mathcal{G}_{\overline{Q}^B} = \mathcal{G}$, i.e. the graph associated with the expected consensus matrix $\overline{Q}^B$ coincides with the underlying communication graph $\mathcal{G}$. Therefore, if $\mathcal{G}$ is strongly connected, then this implies that the randomized broadcast guarantees probabilistic consensus although it does not guarantee average consensus for all possible realizations of $Q^B(t)$. Even if the gossip matrices are not doubly stochastic, the *expected* consensus matrix $\overline{Q}^B$ is doubly stochastic, therefore the elements converge to the

average of the initial conditions in mean sense.

One might also wonder if $\overline{Q}^B$ provides some information about convergence rate for the randomized strategy. In Fagnani and Zampieri (2008) there is an extensive analysis of rates of convergence of the mean square dispersion of final consensus value w.r.t. the average of initial conditions. The main message being that the second largest eigenvalue of $\overline{Q}^B$ provides only an optimistic rate of convergence, and that the dispersion of the final consensus value from the average of the initial conditions decreases as the number of nodes increases.

Similarly, the expected consensus matrix $\overline{Q}^G$ for the symmetric gossip is given by:

$$\left[\overline{Q}^G\right]_{mn} = \begin{cases} 1 - \sum_{i \in \overline{\mathcal{N}}(n)} \frac{2c \cdot w}{N\left(\bar{d}(n)+\bar{d}(i)\right)} & \text{if } m = n \\ \frac{2c \cdot w}{N\left(\bar{d}(m)+\bar{d}(n)\right)} & \text{if } (m,n) \in \mathcal{E},\ m \neq n \\ 0 & \text{otherwise.} \end{cases}$$

Obviously $\overline{Q}^G = \left(\overline{Q}^G\right)^T$ since all the gossip matrices Q_{ij}^G from which the distribution is drawn are symmetric by construction. Similarly to the broadcast, $\mathcal{G}_{\overline{Q}^G} = \mathcal{G}$. Therefore, if $\mathcal{G}$ is strongly connected, then the randomized symmetric gossip guarantees probabilistic average consensus. Compared to the randomized broadcast, the randomized symmetric gossip guarantees average consensus for all realizations. The rate of converge is however much slower, as can be guessed by the observation that the off-diagonal elements of the matrix $\overline{Q}^G$ are smaller than their counterparts in $\overline{Q}^B$, i.e. there is slower information propagation.

These results on the convergence, the properties, and the performances of consensus algorithms will be used many times in Chapter 3, when some specific applications and methods will be discussed.

2.3 Algorithm distribution via optimization decomposition

Distributed decision making

Optimization algorithms are among the first algorithms that have been object of research for a distributed implementation, because of the need of solving large-scale problems via multiple parallel processors. In other words, distributed optimization has historically been intended as the problem of dispatching part of a large scale optimization algorithm to different computational units (see the seminal work Tsitsiklis, Bertsekas, and Athans 1986, and Bertsekas and Tsitsiklis 1997).

The methods that have been developed in this sense can be considered as a foundation for the problems of *distribution decision making*.

While the two problems (distributed optimization and distributed decision making) share many common aspects, they have different rationales. In distributed decision making, the way in which decision variables are assigned to different agents is not part of the designer degrees of freedom, nor is the communication graph. Agents have to decide on their local variables in order to maximize some system-wide value function (which may or may not be the sum of individual utilities functions) while satisfying some *constraints* that couple their choices.

More recently, distributed optimization (or, better, distributed decision making) has been applied to complex, large-scale systems and networked control systems, introducing some other issues in the picture. In this scenario, which is the object of this thesis, the agents are allowed to communicate via a given communication channel, but also to interact with an underlying physical system by sensing and actuating it. The behavior of each agent, and therefore the optimization algorithm that they implement, must necessarily depend on local data and on the information that each node can gather from the nodes that are in its communication neighborhood, but also on the local measurements that they can perform.

After a mathematical description of the optimization problems that networked control systems are required to solve, some *decomposition methods* will be presented, showing how these tools can help in the design of the individual optimization laws.

Let $x_i \in \mathbb{R}^p$ be the state of agent i, and $x \in \mathbb{R}^{pN}$ be the state of the entire system, as defined in the previous section.

Let $f_0(x_1, \dots, x_N)$ (or equivalently, with a minor abuse of notation, $f_0(x)$) be a scalar

convex function of the nodes' states, and consider the convex optimization problem

$$\begin{aligned} \min_{x} \quad & f_0(x) \\ \text{subject to} \quad & h_i(x) \leq 0, \quad i = 1, \dots, r \\ & a_i^T x = b_i, \quad i = 1, \dots, s \end{aligned} \tag{2.2}$$

where h_i, $i = 1, \dots, r$ are convex scalar functions, a_i^T are the rows of a matrix $A \in \mathbb{R}^{s \times pN}$, and b_i are the elements of a vector $b \in \mathbb{R}^s$. It is easy to show that the r inequality constraints and the s equality constraints define a convex feasible set X for x.

Plenty of algorithms have been designed for convex optimization problems, and a strong and useful theory has been derived (see Boyd and Vandenberghe 2008 and the many references therein). Many different problems can be casted into this framework, as the following examples shows.

Example 2.3.1 (Average consensus / distributed least squares)**.** The problem of average consensus introduced in Section 2.2 can be expressed as the problem of computing (and making all nodes agree on) x^* that solves

$$\min_{x} \quad \sum_{i \in \mathcal{V}} (x - y_i)^2,$$

where y_i is the measurement available to node i. This is not, however, expressed as a distributed optimization problem, as x is a system-wide scalar quantity. The same problem in the framework of distributed decision making takes the form

$$\begin{aligned} \min_{x_i, i \in \mathcal{V}} \quad & \sum_{i \in \mathcal{V}} (x_i - y_i)^2 \\ \text{subject to} \quad & x_i = x_j \quad \forall (i,j) \in \mathcal{E} \\ & \mathcal{G} = (\mathcal{V}, \mathcal{E}) \quad \text{connected.} \end{aligned}$$

The following result characterize optimality for convex optimization problems.

Lemma 2.3.2. *Consider the convex optimization problem* (2.2)*. Assume f_0 is differentiable; then x^* is optimal if and only if*

- *$x^* \in X$ (i.e. $h_i(x^*) \leq 0$ for $i = 1, \dots, r$, and $Ax^* = b$;*
- *the gradient ∇f_0 satisfies $\nabla f_0(x^*)^T (y - x) \geq 0$ for all $y \in X$.*

Note that, if $r = 0$ (no inequality constraint are defined), then the optimality condition

stated in Lemma 2.3.2 correspond to the condition

$$\nabla f_0(x^*) \in \mathrm{Im} A^T.$$

Many algorithms exist for the centralized solution of convex optimization problems: gradient descent, Newton, interior-point methods, among others. Decentralized algorithms are generally based on gradient-descent methods, in the form

$$x(t+1) = P_X \left[x(t) - \alpha \nabla f_0(x(t)) \right], \tag{2.3}$$

where P_X is the projection operator on the feasible set X, and α is a suitable step length.

To implement (2.3), nodes have to know the gradient of f_0 in the current system state $x(t)$, and to be able to implement the projection P_X.

If both the cost function f_0 and the feasible set X are separable, i.e.

$$f_0(x) = \sum_{i \in \mathcal{V}} f_i(x_i) \quad \text{and} \quad X = \{x | x_i \in X_i \ \forall i \in V\},$$

then (2.3) can be implemented independently by the nodes, in the form

$$x_i(t+1) = P_{X_i} \left[x_i(t) - \alpha \nabla f_i(x_i(t)) \right],$$

provided that they can compute $\nabla f_i(x_i(t))$. However, this corresponds to the trivial case in which the individual decision of the agents are decoupled.

In Nedic and Ozdaglar (2009) the authors considered the more interesting case in which nodes can compute their individual gradient $\nabla f_i(x)$, and they have to agree on the same minimizer x^*:

$$\begin{aligned} \min_{x_i,\, i \in \mathcal{V}} \quad & \sum_{i \in \mathcal{V}} f_i(x_i) \\ \text{subject to} \quad & x_i = x_j \quad \forall (i,j) \in \mathcal{E} \\ & \mathcal{G} = (\mathcal{V}, \mathcal{E}) \quad \text{connected.} \end{aligned}$$

In their *distributed subgradient method*, agents agree on the optimal value (and therefore satisfy the constraints of the optimization problem) only asymptotically. This is often accepted in distributed optimization, but may not be acceptable when applied to NCS.

If feasibility has to be enforced at any time, then each iterative step must be followed by the projection of the state on the feasible set X, as in (2.3). This is typically difficult to achieve in a distributed manner, and requires coordination and information exchange

between nodes.

The choice of the step length α is another critical issue in the implementation of (2.3). Indeed, it can be shown that convergence of gradient driven algorithms generally require that α is smaller than some threshold, which depends on global quantities, typically some bounds on the Hessian of f_0. It is quite common in NCS that none of the nodes have a global knowledge of the system (which can also be subject to changes in time). Also, line search methods, which are quite common in convex optimization and prescribe that the cost function is evaluated for different α's before deciding for the optimal step length, are in general impractical in NCS, because in many cases driving the system into a new state corresponds to actuating a physical underlying system, and because it is possible that none of the nodes is able to compute the value of f_0, which depends on the whole system state x.

It should be clear that solving optimization problems defined on large scale NCS poses some challenges, whose solution strongly depends on the specific application (as shown in the Chapter 3). Some methodologies however come into help quite frequently, namely *decomposition methods*. They will be presented hereafter, showing how they can help in distributing decision problems among nodes.

Decomposition methods

In distributed optimization, the basic idea of decomposition is to divide the original large optimization problem into smaller subproblems, which are then coordinated by a master problem by means of signalling. When applied to NCS, not every decomposition is possible, because the division into subproblems is usually inherited from the multiagent architecture of the system; however, these methodologies are useful in understanding which are the *signals* that agents should exchange.

Most of the existing decomposition techniques can be classified into primal decomposition and dual decomposition methods.

Primal decomposition

Consider the following optimization problem:

$$\begin{aligned} \min_{x_i', x_i'', i\in\mathcal{V}} \quad & \sum_{i\in\mathcal{V}} f_i(x_i', x_i'') \\ \text{subject to} \quad & x_i'' = x_j'' \quad \forall (i,j)\in\mathcal{E} \\ & x_i' \in X_i' \quad \forall i \in \mathcal{V} \\ & x_i'' \in X'' \quad \forall i \in \mathcal{V} \\ & \mathcal{G} = (\mathcal{V}, \mathcal{E}) \quad \text{connected.} \end{aligned} \tag{2.4}$$

The decision variables of the individual agent i have been divided into two different sets: some *private variables* x_i', and some *public variables* x_i''. Public variables are coupled by one or more constraint (which may be more general than a simple equality constraint). Private variables, instead, can be manipulated independently by the nodes they belong to.

Suppose that each agent can solve the subproblem

$$\begin{aligned} \phi_i(\bar{x}'') = \min_{x_i'} \quad & f_i(x_i', \bar{x}'') \\ \text{subject to} \quad & x_i' \in X_i' \end{aligned}$$

for a given, fixed, $\bar{x}''$. A *master problem* can then be defined as

$$\begin{aligned} \min_{x''} \quad & \sum_{i\in\mathcal{V}} \phi_i(x'') \\ \text{subject to} \quad & x'' \in X''. \end{aligned}$$

A supervisioning node (or one of the agents, elected for this role) can then solve the master problem, if the agents provide their gradient

$$\nabla\phi_i(x'') = \frac{\partial\phi_i(x'')}{\partial x''}.$$

Iteratively, the master node decides on the public variable x'' and communicates its choice to the agents. The agents solve the individual subproblems and return the gradient $\nabla\phi_i$ (which is often available as a side-product of the optimization). The master node then updates x'' according to the global gradient $\sum_{i\in\mathcal{V}} \nabla\phi_i(x'')$.

This approach becomes interesting when the private variables have high dimension while the public variables have low dimension. In this case, communication is reduced to a minimum, as only public variables (and the gradients $\nabla\phi_i$, which have low dimension too)

have to be transmitted. Moreover, the master node do not have to know the private variables, nor the details of the optimization subproblems. Agents could be heterogeneous units with different sets of private variables, or they could have some privacy and security concerns about sharing their operating strategies.

One possible interpretation of this decomposition technique is that the master problem directly gives each subproblem an amount of resources that it can use; the individual agents operate in the best possible way, given those resources; the role of the master problem is then to properly allocate the existing resources, receiving from the agents only a quantitative information about "how better" they could do, if provided with extra resources. Note that the constraints are always satisfied, while optimality of the allocation is achieved only asymptotically.

Dual decomposition

To introduce the tool of dual decomposition, consider the same problem defined in (2.4), but in the simpler case of two agents.

$$\begin{aligned} \min_{x_i', x_i'',\, i=1,2} \quad & f_1(x_1', x_1'') + f_2(x_2', x_2'') \\ \text{subject to} \quad & x_1'' = x_2'' \\ & x_i' \in X_i' \quad \text{for } i = 1,2 \\ & x_i'' \in X'' \quad \text{for } i = 1,2. \end{aligned} \tag{2.5}$$

The extension to an arbitrary number of agents do not pose any additional difficulties but requires a more involved notation. A multi-agent application of dual decomposition will be presented later in this section.

To form the *dual problem* of (2.5), consider the *Lagrangian*

$$L(x_1', x_1'', x_2', x_2'', \eta) = f_1(x_1', x_1'') + f_2(x_2', x_2'') + \eta(x_1'' - x_2'').$$

The dual function $g(\eta)$ is defined as

$$g(\eta) = \inf_{x_i' \in X_i', x_i'' \in X''} L.$$

Note that L is separable into individual terms for the agents:

$$\begin{aligned} L_1(x_1', x_1'', \eta) &= f_1(x_1', x_1'') + \eta x_1'' \\ L_2(x_2', x_2'', \eta) &= f_2(x_2', x_2'') - \eta x_2'', \end{aligned}$$

and therefore $g(\eta)$ can be decomposed into $g_1(\eta) + g_2(\eta)$, where

$$g_1(\eta) = \inf_{x_1' \in X_1', x_1'' \in X''} L_1, \qquad g_2(\eta) = \inf_{x_2' \in X_2', x_2'' \in X''} L_2.$$

Agents can compute $g_1(\eta)$ and $g_2(\eta)$ independently, when they are provided with η. The *dual problem* is defined as

$$\max_{\eta} g(\eta) = g_1(\eta) + g_2(\eta),$$

and it can be solved via any gradient driven algorithm. Evaluating the gradient of g_1 (or g_2) for a given value $\bar{\eta}$ of the dual variable is indeed easy. Consider first $g_1(\eta)$. Let $\bar{x}_1'$ and $\bar{x}_1''$ be the arguments that solve

$$\inf_{x_1', x_1''} L_1(x_1', x_1'', \bar{\eta}).$$

The following holds:

$$\begin{aligned} g_1(\eta) &= \inf_{x_1' \in X_1', x_1'' \in X''} L_1(x_1', x_1'', \eta) \\ &= \inf_{x_1' \in X_1', x_1'' \in X''} f_1(x_1', x_1'') + \eta x_1'' \\ &\leq f_1(\bar{x}_1', \bar{x}_1'') + \eta \bar{x}_1'' \\ &= f_1(\bar{x}_1', \bar{x}_1'') + \bar{\eta} \bar{x}_1'' - (\eta - \bar{\eta}) \bar{x}_1'' \\ &= g_1(\bar{\eta}) - (\eta - \bar{\eta}) \bar{x}_1''. \end{aligned}$$

As this holds for any η and $g_1(\eta)$ is concave, $-\bar{x}_1''$ must be the gradient (or, better, *a subgradient* in the case of non-smooth functions) of g in $\bar{\eta}$.

Similarly, a subgradient of $g_2(\eta)$ is given by $\bar{x}_2''$. Therefore a subgradient of the dual function $g(\eta)$ is given by $\bar{x}_2'' - \bar{x}_1''$, which is nothing more than the consistency constraint residual.

The resulting maximization method for $g(\eta)$ will then consist in the alternate execution of an algorithm for the update of η according to the gradient $\bar{x}_2'' - \bar{x}_1''$, and two individual algorithms at the agents that compute $\bar{x}_i'$ and $\bar{x}_i''$ by minimizing the Lagrangian L_i for the given dual variable η.

Duality theory guarantees that the solution of the dual problem is a lower bound for the original optimization problem. That is

$$\max_{\eta} g(\eta) \leq \min_{x_i', x_i'', i=1,2} f_1(x_1', x_1'') + f_2(x_2', x_2'').$$

For some classes of problems, however, the duality gap (the difference between the solution

of the primal and the dual problem) is zero, and therefore solving the maximization of $g(\eta)$ returns the solution of the original problem (*strong duality* is said to hold). This is the case, for example, of convex problems where dual decomposition is applied to equality constraints. For other characterizations of *strong duality*, see Boyd and Vandenberghe (2008).

In the case of two agents, the dual problem is a scalar problem. In the more general case of many agents, the size of the dual problem corresponds to the number of constraints that have to be satisfied. In particular, if agents correspond to nodes of a graph, and consistency constraints correspond to the edges of the graph, the dual problem will have the form

$$g(\eta_e,\, e\in\mathcal{E}) = \inf_{x_i',x_i'',i\in\mathcal{V}} \left(\sum_{i\in\mathcal{V}} f_i(x_i', x_i'') + \sum_{e\in\mathcal{E}} \eta_e \left(x''_{s(e)} - x''_{t(e)} \right) \right).$$

The decision variables of the dual problem can therefore be assigned to the edges of the graph, and can be updated according to the subgradients

$$\frac{\partial\, g(\eta_e,\, e\in\mathcal{E})}{\partial\, \eta_e} = \bar{x}''_{s(e)} - \bar{x}''_{t(e)}.$$

In some applications there is some intelligence associated with the edges, and it can take care of solving the dual problems; in other cases, one of the two nodes connected by the edge can be assigned to the dual problem.

One possible interpretation for the dual decomposition is that the dual algorithm sets the price for the resources to each subproblem, which has to decide the amount of resources to be used depending on the price; the role of the dual problem is then to obtain the best pricing strategy, on the basis of the difference between demand and offer.

Note that, differently for the primal decomposition algorithm, feasibility of the solution is not guaranteed at every iteration.

An example: traffic congestion control in data networks

A notable "success story" for the application of distributed optimization methods to NCS is the traffic congestion control in data networks: since the work of Kelly, Maulloo, and Tan (1998), large-scale data networks have been probably the preferred testbed for these algorithms. The problem will now be presented briefly, following Chiang, Low, Calderbank, and Doyle (2007).

Consider a data network described by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Let each edge $e \in \mathcal{E}$ have a data transfer capacity $c_e > 0$. Suppose that every node is interested in transmitting some data to another point in the newtork. This can be modeled by associating to each node v a subset $\mathcal{E}_v$ of the edges, corresponding to the path from v to the destination of the transmission.

Conversely, let $\mathcal{V}_e$ be the set of nodes that use the edge e for their transmission. The interest of v in transmitting the data is described by a strictly concave utility function $U_v(x_v)$ of the rate of transmission x_v.

The following utility maximization problem is then well defined:

$$\begin{aligned} \max_{x_v,\, v\in\mathcal{V}} \quad & \sum_{v\in\mathcal{V}} U_v(x_v) \\ \text{subject to} \quad & \sum_{v\in\mathcal{V}_e} x_v \le c_e, \quad \forall\, e\in\mathcal{E}. \end{aligned}$$

The associated Lagrangian is

$$L(x_v, v\in\mathcal{V}, \eta_e, e\in\mathcal{E}) = \sum_{v\in\mathcal{V}} U_v(x_v) + \sum_{e\in\mathcal{E}} \eta_e \left(c_e - \sum_{v\in\mathcal{V}_e} x_v \right).$$

This yields the following dual function (where η is the vector of all η_e's)

$$\begin{aligned} g(\eta) &= \max_{x_v,\, v\in\mathcal{V}} \left[\sum_{v\in\mathcal{V}} U_v(x_v) + \sum_{e\in\mathcal{E}} \eta_e \left(c_e - \sum_{v\in\mathcal{V}_e} x_v \right) \right] \\ &= \sum_{v\in\mathcal{V}} \max_{x_v} \left(U_v(x_v) - x_v \sum_{e\in\mathcal{E}_v} \eta_e \right) + \sum_{e\in\mathcal{E}} \eta_e c_e. \end{aligned}$$

The subproblem associated to each node is then

$$\max_{x_v} \left(U_v(x_v) - x_v \sum_{e\in\mathcal{E}_v} \eta_e \right), \tag{2.6}$$

which requires that node v knows the sum of the dual variables η_e associated to all the edges that it is using. Before discussing how this can be achieved, consider the dual problem that edges have to solve to update the dual variable η. As shown before, a subgradient for the problem of minimizing $g(\eta)$ is returned by the solution of the nodes' subproblem. The component of $\nabla g(\eta)$ corresponding to edge e is given by:

$$\frac{\partial\, g(\eta)}{\partial\, \eta_e} = c_e - \sum_{v\in\mathcal{V}_e} \bar{x}_v, \tag{2.7}$$

where $\bar{x}_v$ is the solution of the Lagrangian maximization (2.6). It therefore depends on the cumulative data transmission rate of all the nodes that are using that edge.

It is then needed that every edge implement (2.7) and that every node v compute $\sum_{e\in\mathcal{E}_v}\eta_e$, which seems to be unpractical, as nodes are not aware of which edges belong to $\mathcal{E}_v$ (i. e. which edges they are using). Moreover, collecting the data η_e from all the edges in the path via the same communication network that is being controlled would hardly be acceptable, because of the resulting communication overhead.

What happens in real data network (under a very rough approximation that is commented and motivated in far more details in Chiang et al. 2007) is that the packets that travel through a congestioned link experience time delays and non-zero packet loss probability. This phenomena can be caused by intentional strategies of the routers (the devices that manage the traffic on the links), but can also be inherent in the communication channel and in the packet queueing mechanisms.

Consider for simplicity the case in which packets travelling through link e are delayed by a time τ_e, where τ_e changes in time according to

$$\tau_e(t+1)=\tau_e(t)-\alpha_\tau\left(c_e-\sum_{v\in\mathcal{V}_e}\bar{x}_v\right), \tag{2.8}$$

where α_τ is a positive design parameter, and where $c_e-\sum_{v\in\mathcal{V}_e}\bar{x}_v$ is the residual transmission capacity of the link e. The update (2.8), executed by all the edges, is then implementing a gradient descent algorithm for the minimization of $g(\tau)$, there τ is the vector of the time delays of all edges, which plays exactly the role of the dual variable η.

The generic node v that has to solve the maximization of the Lagrangian (2.6), can therefore implement a gradient ascent algorithm in the form

$$\begin{aligned}x_v(t+1)&=x_v(t)+\alpha_x\frac{\partial}{\partial x_v}\left(U_v(x_v)-x_v\sum_{e\in\mathcal{E}_v}\tau_e\right)\\&=x_v(t)+\alpha_x\left(U_v'(x_v)-\sum_{e\in\mathcal{E}_v}\tau_e\right)\\&=x_v(t)+\alpha_x\left(U_v'(x_v)-T_v\right),\end{aligned}$$

where α_x is a positive design parameter, and T_v is the total, point to point, delay in the packet transmission that node v is experiencing.

In real data network the agents indeed control their transmission rate according to some protocols (TCP) that resemble this behavior.

This example is extremely interesting because via dual decomposition techniques, a

complex global optimization problem has been divided into smaller and tractable problems; however, what is really worth noticing is that part of the decomposition, namely the dual problem and the communication of the dual variables, is not solved by any controller, but is inherently solved by the underlying system. At a first analysis the nodes are not using the communication channels to exchange any dual variable for the solution of this optimization problem; however, the specific behavior of the system is exploited as a *signaling* mechanism for the satisfaction of the given capacity constraints. Notice that this approach requires that the system is actuated at any iteration, because edges obtain the primal variables x_v by measuring the congestion of the link (and therefore the transmission rate must have been updated) and the nodes obtaine the dual variables η_e by measuring the time delay of the sent packets (and therefore the edges must indeed delay the packets travelling through them).

This kind of approach is the ultimate goal of the application of distributed optimization methods to NCS. It is worth noticing that traffic congestion control algorithms have been designed without this approach in mind, but on an empirical basis. A similar approach can be recognized in the algorithms proposed in Section 3.3 for a completely different application.

3

Algorithms and applications

3.1 Clock synchronization in sensor networks

The field of sensor networks has been witnessing a large attention in the scientific community during the last decade. This is due in no small part to the remarkable advances which have been made in recent years in the development of small, relatively inexpensive sensor nodes with networking capabilities, ultimately intended for a wide variety of purposes such as search and rescue, environmental monitoring and surveillance just to mention few. Wireless sensor networks (WSNs) have the potential to truly revolutionize the way we monitor and control our environment. Several WSN applications either benefit from, or require, time synchronization (see Sundararaman, Buy, and Kshemkalyani, 2005, and references therein). These applications include, for example, mobile target tracking using a large number of motion detection devices as in Oh, Schenato, and Sastry (2005), habitat monitoring as in Szewczyk, Osterweil, Polastre, Hamilton, Mainwaring, and Estrin (2004), power scheduling

and TDMA communication schemes as in Hohlt, Doherty, and Brewer (2004), and rapid synchronized coordination of powerlines nodes in electric power distribution networks for catastrophic power-outage prevention as in Massoud Amin and Schewe (2007).

Also energy management and power control for battery powered sensors can greatly benefit from precise time synchronization. Indeed, to extend their battery life, sensors may decide to turn their communication circuitries off for most of the time, and turn them on only when communication is needed (see Pantazis and Vergados, 2007, for a survey on these methods). Because of the multihop nature of WSN communications, it is required that nodes belonging to the same routing path coordinate their duty cycles precisely. This specific application will be adopted as a testbed for the proposed algorithm later in this section.

One possible approach to the synchronization problem is electing a reference node and allowing every node to communicate with it. To deal with the problem that communication in a sensor network usually happens through multi-hop paths, these algorithms, for example the Time-synchronization Protocol for Sensor Networks (TPSN) by Ganeriwal, Kumar, and Srivastava (2003) and the Flooding Time Synchronization Protocol (FTSP) by Maròti, Kusy, Simon, and Àkos Lédeczi (2004), require that first a spanning tree rooted in the clock reference is built, and then the offset of any node with the respect to the root is obtained simply by adding the offset of the edges in the unique path from each node to the root.

Although these strategies can be easily implemented and Maròti et al. (2004) showed that they exhibit remarkable performances, it suffers from two main problems. The first problem is robustness. In fact, if a node dies or a new node is added to the network, then it is necessary to rebuild the tree, at the price of additional implementation overhead and possibly long periods in which the network or part of it is poorly synchronized. The second problem is that, depending on how the tree is built, it might happen that two clocks, which are physically close and can communicate with each other, belong to two different branches of the tree or two different clusters, thus possibly having large clock differences.

Fully distributed algorithms for clocks synchronization have recently appeared. For example in Werner-Allen, Tewari, Patel, Welsh, and Nagpal (2005) the Reachback Firefly Algorithm (RFA) is proposed, a protocol inspired by the fireflies integrate-and-fire synchronization mechanism, capable of compensating for different clock offsets but not for different clock skews. On the opposite, the algorithm proposed by Simeone and Spagnolini (2007), adopting a P-controller, is able to compensate for the clock skews but not for the offsets, obtaining constant time differences between the clocks.

Distributed protocols that can compensate for both clock skews and offsets are the Tiny-Sync Protocol by Yoon, Veerarittiphan, and Sichitiu (2007), the Distributed Time-Sync Protocol by Solis et al. (2006) and the Average Time-Sync Protocols by Schenato and Gamba

(2007). The first one is based on a type of robust linear regression, the second on distributed least-square estimator, and the last on a cascade of two consensus algorithms. They are all proved to synchronize a network of clocks in the absence of noise and delivery time-delay and they also show good performance in experimental testbeds. However, in these protocols it is difficult to predict the effect of noise on the steady-state performance.

Differently, Carli, Chiuso, Schenato, and Zampieri (2008b) proposed a synchronization algorithm that can be formally analyzed not only in the noiseless scenario in terms of rate of convergence but also in a noisy setting in terms of the steady-state synchronization error. This algorithm compensates for both initial offsets and differences in internal clock speeds and is based on a Proportional-Integral (PI) controller. Both convergence guarantees as well optimal design using standard optimization tools when the underlying communication graph is known are provided. It is important to remark that the time-synchronization algorithm proposed in Carli et al. (2008b) require each node to perform all the operations related to the k-th iteration of the algorithm, including transmitting messages, receiving messages and updating estimates, within a short time window. This quasi-synchronous implementation might be very sensitive to packet losses, node and link failure.

A far more practical version of the PI synchronization algorithm, based on an asynchronous gossip protocol for communication, has been developed and analyzed in Bolognani, Carli, and Zampieri (2009) and will be described hereafter. In this algorithm, at each iteration one node can estabilish a bi-directional communication with only one of its neighbors. This applies very well to real sensor networks, and drastically reduces the network requirements in terms of reliability, bandwidth, and synchronization. When the underlying communication topology is given by the complete graph, it is possible to derive some analytical convergence results, while more general families of graphs can be considered by means of simulations.

Problem formulation

Consider N clocks. Each clock has its own speed, denoted by d_i and its own initial offset, denoted by o_i. Let us denote with x_i the *local time* estimate of node i, $i \in \{1, \dots, N\}$. The objective is to make all $x_i(t)$ as close as possible to each other for all times t. This can be mathematically formulated by requiring that

$$\bar{x}_i(t) := x_i(t) - \frac{1}{N} \sum_{j=1}^{N} x_j(t)$$

goes to zero as t goes to infinity. The variable $\bar{x}_i(t)$ is called the *synchronization error* of the i-th node. Note that it is not needed that all clocks follow an *absolute time* but rather that

they all agree on their estimate of time. If the nodes cannot communicate, then the local time $x_i(t)$ can be modeled as a discrete time integrator

$$x_i(t+1) = x_i(t) + d_i.$$

Clearly, different initial offset, $x_i(0)$, and different speeds d_i cause synchronization errors.

In Carli et al. (2008b) the authors propose a distributed clock synchronization strategy based on a Proportional-Integral (PI) controller that treats the different clock speeds as unknown constant disturbances and the different clock offsets as different initial conditions for the system dynamics. They assume that it is possible to control each clock by a local input $u_i(t)$ as follows

$$x_i(t+1) = x_i(t) + d_i + u_i(t).$$

To make the algorithm distributed, the control action $u_i(t)$ is only allowed to use *local* information. Precisely, at time instants $t \in \mathbb{N}$ nodes can exchange their local time according to a communication graph $\mathcal{G}$ having $\mathcal{V} = \{1, \dots, N\}$ as the set of vertices and in which there is an edge from i to j whenever the node i can send $x_i(t)$ to the node j. Moreover each node has in memory, besides the estimate $x_i(t)$, another variable denoted by $w_i(t)$; this variable plays an important role in compensating the different speeds d_i. By defining the N-dimensional vectors $x(t), u(t)$ and d as the vectors with components $x_i(t), u_i(t)$ and d_i respectively, the PI consensus controller proposed in Carli et al. (2008b) can be expressed as

$$\begin{aligned} w(t+1) &= w(t) - \alpha K x(t), \qquad w(0) = 0 \\ u(t) &= w(t) - K x(t) \end{aligned}$$

where α is a suitable positive real number in $[0,1]$ and where $K \in \mathbb{R}^{N \times N}$ is such that

i) $I - K$ is an aperiodic and irreducible stochastic matrix (*consensus matrix*),

ii) $K_{ij} \neq 0$ only if (j,i) is an edge of the graph $\mathcal{G}$.

The overall system then becomes

$$\begin{bmatrix} x(t+1) \\ w(t+1) \end{bmatrix} = \begin{bmatrix} I - K & I \\ -\alpha K & I \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \end{bmatrix} + \begin{bmatrix} d \\ 0 \end{bmatrix}. \tag{3.1}$$

The analysis in Carli et al. (2008b) is restricted to symmetric matrices K and shows that, under certain conditions on the parameter α and on the eigenvalues of K, asymptotic synchronization is achieved.

Note that this algorithm requires reliable and synchronized communication along all the

edges of a given graph $\mathcal{G}$. Clearly this is an impracticable condition, as nodes do not have access to the *absolute time* t. To overcome these limitations the authors in Carli and Zampieri (2010) provide a realistic quasi-synchronous implementation of the previous algorithm where each clock carries out all its operations related to the k-th iteration, including transmitting messages, receiving messages and updating its estimate, within a short time window. The effectiveness of this realistic implementation has been confirmed by simulations. However, certain conditions on the size of the time window and on the maximum distance between any pair $x_i(t), x_j(t)$ for any instant t are required. These conditions might be seriously affected by packet losses, node and link failures.

A PI consensus controller with gossip communication

The communication requirements of (3.1) can be reduced by considering a different and more realistic, communication protocol between the clocks. Precisely, assume that the communication model is that of gossiping clocks, i.e. a model in which only a pair of clocks can communicate at any time, as stated in the following assumption.

Assumption 3.1.1. The communication graph $\mathcal{G} = (V, \mathcal{E})$ is an undirected connected graph without any self-loop and, at every time instant t, each edge $(i, j) \in \mathcal{E}$ can be selected with a strictly positive probability $W_{(i,j)}$.

It is then possible to define the matrix $W \in \mathbb{R}^{N\times N}$ as the matrix having W_{ij} as element in the i-th row and j-th column and in the j-th row and i-th column, namely

$$[W]_{ij} = [W]_{ji} := W_{(i,j)}. \tag{3.2}$$

Similarly to (3.1), each node has in memory the two variables x_i and w_i. Assume that, at time t the edge (i, j) is selected, and that x and w are updated according to

$$\begin{aligned}
x_i(t+1) &= \frac{1}{2}\big(x_i(t) + x_j(t)\big) + w_i(t) + d_i(t) \\
x_j(t+1) &= \frac{1}{2}\big(x_i(t) + x_j(t)\big) + w_j(t) + d_j(t) \\
x_h(t+1) &= x_h(t) + w_h(t) + d_h(t) \qquad\qquad \text{for all } h \neq i, j
\end{aligned}$$

and

$$\begin{aligned} w_i(t+1) &= \frac{\alpha}{2}\big(-x_i(t)+x_j(t)\big)+w_i(t) \\ w_j(t+1) &= \frac{\alpha}{2}\big(-x_j(t)+x_i(t)\big)+w_j(t) \\ w_h(t+1) &= w_h(t) \qquad\qquad \text{for all } h\neq i,j \end{aligned}$$

These equations can be rewritten as

$$\begin{aligned} \begin{bmatrix} x(t+1) \\ w(t+1) \end{bmatrix} &= \begin{bmatrix} I-e_ie_i^T-e_je_j^T+\left(e_i+e_j\right)\left(e_i+e_j\right)^T & I \\ -\frac{\alpha}{2}(e_i+e_j)(e_i+e_j)^T-\alpha\left(e_ie_i^T+e_je_j^T\right) & I \end{bmatrix}\begin{bmatrix} x(t) \\ w(t) \end{bmatrix}+\begin{bmatrix} d(t) \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} I-\frac{1}{2}E^{(i,j)} & I \\ -\frac{\alpha}{2}E^{(i,j)} & I \end{bmatrix}\begin{bmatrix} x(t) \\ w(t) \end{bmatrix}+\begin{bmatrix} d(t) \\ 0 \end{bmatrix} \end{aligned}$$

where $E^{(i,j)}=\left(e_i-e_j\right)\left(e_i-e_j\right)^T$. Note that $I-E^{(i,j)}$ is a doubly stochastic matrix. Considering the variable $v(t)=w(t)+d(t)$, the above system can be rewritten as

$$\begin{bmatrix} x(t+1) \\ v(t+1) \end{bmatrix} = \begin{bmatrix} I-\frac{1}{2}E^{(i,j)} & I \\ -\frac{\alpha}{2}E^{(i,j)} & I \end{bmatrix}\begin{bmatrix} x(t) \\ v(t) \end{bmatrix} := F^{(i,j)}\begin{bmatrix} x(t) \\ v(t) \end{bmatrix}.$$

It is convenient to consider the variables

$$\bar{x}(t)=\Omega x(t) \text{ and } \bar{v}(t)=\Omega v(t)$$

where

$$\Omega = I-\frac{1}{N}\mathbf{1}\mathbf{1}^T.$$

Clearly the synchronization is asymptotically reached if and only if $\bar{x}(t)\to 0$. Straightforward calculations show that

$$\begin{bmatrix} \bar{x}(t+1) \\ \bar{v}(t+1) \end{bmatrix} = F^{(i,j)}\begin{bmatrix} \bar{x}(t) \\ \bar{v}(t) \end{bmatrix}, \tag{3.3}$$

i.e. $[\bar{x}\ \bar{w}]^T$ satisfies the same recursive equation of $[x\ w]^T$.

Mean-square analysis of the algorithm

To provide a mean-square analysis of (3.3), it is convenient to introduce the matrix

$$P(t) = \mathbb{E}\left[\begin{bmatrix}\bar{x}(t)\\ \bar{v}(t)\end{bmatrix}\begin{bmatrix}\bar{x}(t)^T & \bar{v}(t)^T\end{bmatrix}\right] = \begin{bmatrix}P_{11} & P_{12}\\ P_{12}^T & P_{22}\end{bmatrix}$$

where

$$\begin{aligned}P_{11}(t) &= \mathbb{E}\left[\bar{x}(t)\bar{x}^T(t)\right]\\ P_{12}(t) &= \mathbb{E}\left[\bar{x}(t)\bar{v}^T(t)\right]\\ P_{22}(t) &= \mathbb{E}\left[\bar{v}(t)\bar{v}^T(t)\right].\end{aligned}$$

The covariance matrix $P(t)$ evolves according to[1]

$$\begin{bmatrix} P_{11} & P_{12}\\ P_{12}^T & P_{22}\end{bmatrix}^+ = \mathbb{E}\left[F^{(i,j)}\begin{bmatrix} P_{11} & P_{12}\\ P_{12}^T & P_{22}\end{bmatrix}F^{(i,j)^T}\right],$$

which yields the following recursive equations

$$\begin{aligned}P_{11}^+ &= P_{11} - \frac{1}{2}\mathbb{E}\left[E^{(i,j)}P_{11}\right] + \frac{1}{4}\mathbb{E}\left[E^{(i,j)}P_{11}E^{(i,j)}\right] - \frac{1}{2}\mathbb{E}\left[P_{11}E^{(i,j)}\right] + P_{12}^T - \frac{1}{2}\mathbb{E}\left[P_{12}^TE^{(i,j)}\right]\\ &\quad + P_{12} - \frac{1}{2}\mathbb{E}\left[E^{(i,j)}P_{12}\right] + P_{22}\\ P_{12}^+ &= \frac{\alpha}{2}\mathbb{E}\left[P_{11}E^{(i,j)}\right] - \frac{\alpha}{4}\mathbb{E}\left[E^{(i,j)}P_{11}E^{(i,j)}\right] + \frac{\alpha}{2}\mathbb{E}\left[P_{12}^TE^{(i,j)}\right] + P_{12} - \frac{1}{2}\mathbb{E}\left[E^{(i,j)}P_{12}\right] + P_{22}\\ P_{22}^+ &= \frac{\alpha^2}{4}\mathbb{E}\left[E^{(i,j)}P_{11}E^{(i,j)}\right] + \frac{\alpha}{2}\mathbb{E}\left[P_{12}^TE^{(i,j)}\right] + \frac{\alpha}{2}\mathbb{E}\left[E^{(i,j)}P_{12}\right] + P_{22}\end{aligned}$$

The covariance matrix P then updates according to a linear transformation

$$P(t+1) = \mathcal{L}\left[P(t)\right]$$

defined by the recursive equations that have just been computed, and whose initial conditions can be obtained once the following assumption on $x(0)$ and $d(0)$ has been stated.

Assumption 3.1.2. The initial conditions $x(0)$ and $d(0)$ are random vectors such that $\mathbb{E}\left[x(0)\right] = 0$, $\mathbb{E}\left[x(0)x^T(0)\right] = \sigma_x^2 I$ and $\mathbb{E}\left[d(0)\right] = \mathbf{1}$, $\mathbb{E}\left[d(0)d^T(0)\right] = (\sigma_d^2+1)I$ for some $\sigma_x^2 > 0$ and $\sigma_d^2 > 0$. Moreover $w(0) = 0$.

[1] Time dependance has been omitted in these recursive equations. The superscript plus sign indicates the value of the variables at time $t+1$.

It then follows that

$$P = \begin{bmatrix} P_{11}(0) & P_{12}(0) \\ P_{12}^T(0) & P_{22}(0) \end{bmatrix} = \begin{bmatrix} \sigma_x^2\Omega & 0 \\ 0 & (\sigma_d^2+1)\Omega. \end{bmatrix} \tag{3.4}$$

The analysis of the convergence properties of $\mathcal{L}$ is a challenging problem when $\mathcal{G}$ is an arbitrary graph. In the next section, a theoretical analysis for the complete graph is provided. Later a more realistic family of graphs will be considered via numerical simulations.

Complete communication graph

Assume that the graph $\mathcal{G}$ describing the feasible communications between nodes is the complete graph. Moreover, assume that each edge has the same probability $\frac{2}{N(N-1)}$ of being selected. Hence

$$W = \frac{2}{N(N-1)}(\mathbf{1}\mathbf{1}^T - I). \tag{3.5}$$

The following technical lemma holds.

Lemma 3.1.3. *Let W be defined as in (3.5). Then*

$$\mathbb{E}\left[E^{(i,j)}\Omega\right] = \mathbb{E}\left[\Omega E^{(i,j)}\right] = \frac{2}{N-1}\Omega$$
$$\mathbb{E}\left[E^{(i,j)}\Omega E^{(i,j)}\right] = \frac{4}{N-1}\Omega$$

Proof. The given expressions follow just by direct computation. ∎

The two following results characterize the evolution of the matrices P_{11}, P_{12} and P_{22} when $\mathcal{G}$ is the complete graph and the matrix W has the expression given in (3.5).

Proposition 3.1.4. *In the case of a complete graph and of equiprobability of edge selection, the set*

$$\mathcal{J} = \left\{ P | P = \begin{bmatrix} a & b \\ b & c \end{bmatrix} \otimes \Omega \right\}$$

is invariant under the transformation (3.1).

Proof. Let $P(t) = P$, with $P \in \mathcal{J}$. Then

$$\begin{aligned} P_{11}(t+1) &= \frac{N-2}{N-1} a\Omega + 2\frac{N-2}{N-1} b\Omega + c\Omega \\ P_{12}(t+1) &= \left(1 - \frac{\alpha+1}{N-1}\right) b\Omega + c\Omega \\ P_{22}(t+1) &= \frac{\alpha^2}{N-1} a\Omega - \frac{2\alpha}{N-1} b\Omega + c\Omega \end{aligned}$$

and therefore $P(t+1) \in \mathcal{J}$. ∎

Proposition 3.1.5. *In the case of a complete graph and equiprobability of edge selection, with Assumption 3.1.2 holding, $P(t)$ can be expressed as*

$$P_{11}(t) = p_{11}(t)\Omega, \qquad P_{12}(t) = p_{12}(t)\Omega, \qquad P_{22}(t) = p_{22}(t)\Omega,$$

where

$$\begin{bmatrix} p_{11}(t+1) \\ p_{12}(t+1) \\ p_{22}(t+1) \end{bmatrix} = \begin{bmatrix} \frac{N-2}{N-1} & 2\frac{N-2}{N-1} & 1 \\ 0 & 1 - \frac{\alpha+1}{N-1} & 1 \\ \frac{\alpha^2}{N-1} & -\frac{2\alpha}{N-1} & 1 \end{bmatrix} \begin{bmatrix} p_{11}(t) \\ p_{12}(t) \\ p_{22}(t) \end{bmatrix}$$

with initial conditions

$$\begin{bmatrix} p_{11}(0) \\ p_{12}(0) \\ p_{22}(0) \end{bmatrix} = \begin{bmatrix} \sigma_x^2 \\ 0 \\ 1 + \sigma_d^2 \end{bmatrix}.$$

Proof. The result just follows from the fact that the initial condition (3.4) is in $\mathcal{J}$, and therefore the trajectories of the system can be parametrized in the form

$$P(t) = \begin{bmatrix} p_{11}(t) & p_{12}(t) \\ p_{12}(t) & p_{22}(t) \end{bmatrix} \otimes \Omega$$

because of the invariance stated in Proposition 3.1.4. The same proposition gives also the update equations for the three parameters of the covariance matrix. ∎

Algorithm convergence

Theorem 3.1.6. *Consider the described network of clocks, with a gossip communication protocol over a complete graph and a edge selection probability matrix W as in (3.5). Let the system be initialized according to Assumption 3.1.2. Then the variance P of the synchronization error*

converges exponentially to zero if and only if

$$\alpha < \bar{\alpha} = \frac{3}{2} - N + \frac{1}{2}\sqrt{4N^2 - 12N + 17}. \tag{3.6}$$

Proof. The covariance matrix of the synchronization error evolves according to the linear update law and the initial conditions stated in Proposition 3.1.5.

The stability of the update equation can then be studied by eigenvalue analysis. The characteristic polynomial of the update matrix in Proposition 3.1.5 is

$$\begin{aligned}\Psi(z) = &-\frac{-N^2+2N-1}{(N-1)^2}z^3 - \frac{-8N+\alpha-\alpha N+3N^2+5}{(N-1)^2}z^2 \\ &- \frac{-8-\alpha+10N-3N^2-\alpha^2+\alpha^2 N}{(N-1)^2}z - \frac{4-2\alpha-4N+N^2+\alpha N-2\alpha^2+\alpha^3+\alpha^2 N}{(N-1)^2}\end{aligned}$$

An efficient way to study the stability of $\Psi(z)$ is by applying Routh criterion to the continuous time version of $\Psi'(z)$, where the discrete time to continuous time conversion is performed by Tustin transformation, which preserves stability and maps the unit circle into the left hand semiplane.

Substituting $z = (1+s)/(1-s)$ into $\Psi(z)$, one gets

$$\begin{aligned}\Psi_c(s) = \frac{1}{(N-1)^2(1-s)^3}\Big[&\left(18+\alpha^3-\alpha^2-24N+8N^2\right)s^3 \\ &+\left(-3\alpha^3+(5-2N)\alpha^2+(6-4N)\alpha-12+8N\right)s^2 \\ &+\left(2+3\alpha^3+(4N-7)\alpha^2+(4N-8)\alpha\right)s-\alpha^3+(3-2N)\alpha^2+2\alpha\Big]\end{aligned}$$

According to Routh criterion, the stability of the numerator of $\Psi_c(s)$, and therefore of the original matrix A (a part from the three poles in 1), can be studied by looking at the sign of

four terms, namely

$$\begin{aligned}
r_3 &= 18+\alpha^3-\alpha^2-24N+8N^2\\
r_2 &= -3\,\alpha^3+(5-2N)\,\alpha^2+(6-4N)\,\alpha-12+8N\\
r_1 &= \Big[\alpha^6+(2N-4)\,\alpha^5+\left(N^2-N-1\right)\alpha^4+\left(2N^2+13-11N\right)\alpha^3\\
&\quad+\left(1-7N-2N^3+7N^2\right)\alpha^2+\left(-9-2N^2+9N\right)\alpha+3-2N\Big]\\
&\quad\cdot\frac{8}{3\,\alpha^3+(2N-5)\,\alpha^2+(4N-6)\,\alpha+12-8N}\\
r_0 &= -\alpha^3+(3-2N)\,\alpha^2+2\,\alpha
\end{aligned}$$

Routh's criterion states that to have stability of the polynomial, all the four terms r_i must have the same sign. Under the technical assumption $N>2$ (which is a necessary condition to have a properly randomized algorithm) the terms r_3, r_2, and r_1 are always positive (as shown hereafter), so stability only depends on the sign of r_0.

Sign of the term r_3

Suppose that $\alpha=0$. In this case term r_3 reduces to the second-order polynomial in N

$$r_3\big|_{\alpha=0}=18-24N+8N^2$$

which can easily proved to be greater than zero for all $N\in\mathbb{N}$, as it describes a convex parabola with its vertex in $N=3/2$, where its value is zero. When $\alpha>0$ is considered, it is still easy to see that, even if the parabola now crosses the N-axis, r_3 is still greater than zero for any $N>2$ (more precisely, for $N>3/2+\alpha/4\cdot\sqrt{2-2\alpha}$, which is less than 2. Therefore $r_3>0$ for all N's greater than 2.

Sign of the term r_2

All the positive powers of α in r_2 have negative coefficient, if $N>2$. Therefore, it results

$$\begin{aligned}
r_2 &= -3\,\alpha^3+(5-2N)\,\alpha^2+(6-4N)\,\alpha-12+8N\\
&\geq (8-6N)\alpha-12+8N.
\end{aligned}$$

Therefore $r_2>0$ if

$$\alpha<\frac{12-8N}{8-6N}.$$

This is always true for $\alpha\in[0,1]$ and $N>2$. Indeed

$$\left.\frac{12-8N}{8-6N}\right|_{N=3}=\frac{6}{5}$$

and it is increasing in N, as

$$\left.\frac{12-8N}{8-6N}\right|_{N+1} - \left.\frac{12-8N}{8-6N}\right|_{N} = \frac{2}{(3N-1)(3N-4)} > 0\,.$$

Therefore $r_2 > 0$ for all the considered α's and N's.

Sign of the term r_1

As $\alpha \leq 1$, the denominator of r_1 satisfies

$$\mathrm{DEN}r_1 \leq -2N + 4 < 0$$

once $N > 2$. As the denominator is then negative, negativity of the numerator implies $r_1 > 0$.

The coefficients of the four higher powers of α in the numerator are all positive, so, as $\alpha < 1$, α^6, α^5 and α^4 can be bound by α^2, and α^3 by α, so that

$$\mathrm{NUM}_{r_1} < (-2N^3 + 8N^2 - 6N - 3)\alpha^2 + (4-2N)\alpha + 3 - 2N\,.$$

This expression is negative, as all the coefficients of the polynomial are negative. This is trivially true for the coefficients of α and for the costant term. The coefficient of α^2 requires a little more analysis. It is easy to check its negativity for the smallest N, $N = 3$. Its discrete increment is

$$\left(-2n^3 + 8n^2 - 6n - 3\right)_{n=N+1} - \left(-2n^3 + 8n^2 - 6n - 3\right)_{n=N} = -6N^2 + 10N$$

therefore it is decreasing for $N \geq 2$. As both the numerator and the denominator of r_1 are negative, r_1 is positive.

Sign of the term r_0

Term r_0 is a third order polynomial but its sign can be easily studied as $\alpha = 0$ is one of its roots. As $\alpha > 0$, it must be

$$-\alpha^2 + (3-2N)\alpha + 2 > 0\,. \tag{3.7}$$

The roots of (3.7) are

$$\alpha_{1,2} = \frac{3}{2} - N \pm \frac{1}{2}\sqrt{4N^2 - 12N + 17}$$

and therefore the stability condition results to be

$$0 < \alpha < \frac{3}{2} - N + \frac{1}{2}\sqrt{4N^2 - 12N + 17}. \tag{3.8}$$

■

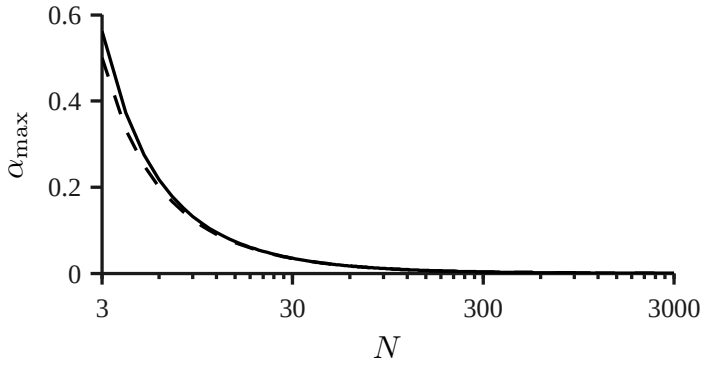


Figure 3.1: Comparison of the conservative bound (3.9) on α (dashed) and the exact bound (3.6) (continuous).

Corollary 3.1.7. *Under the same hypoteses of Theorem 3.1.6, a sufficient condition on α for the variance P of the synchronization error to go to zero is that*

$$0 < \alpha \leq \frac{1}{N-1} \tag{3.9}$$

Proof. The proof follows exactly what has been done for Theorem 3.1.6. When studying the positivity of the term r_0, though, one can use the fact that $\alpha^2 < \alpha$, obtaining

$$-\alpha^2 + (3-2N)\alpha + 2 > 2(1-N)\alpha + 2$$

which is greater than zero for

$$\alpha \leq \frac{1}{N-1}.$$

■

It is easy to see that condition on α expressed in Theorem 3.1.6 and in Corollary 3.1.7 are very close (still being the bound in the corollary far more readable and meaningful). Figure 3.1 put the two conditions in comparison.

The convergence result derived by analyzing the covariance matrix of the synchronization error is even more meaningful once it allows to prove the convergence of the system with probability 1, as the following corollary states.

Corollary 3.1.8. *Under the same hypoteses of Theorem 3.1.6, if $\alpha < \bar{\alpha}$, then there exist with probability 1 a positive real number η such that, eventually,*

$$\left\| \begin{matrix} \bar{x}(t) \\ \bar{v}(t) \end{matrix} \right\|_2^2 \leq e^{-\eta t}.$$

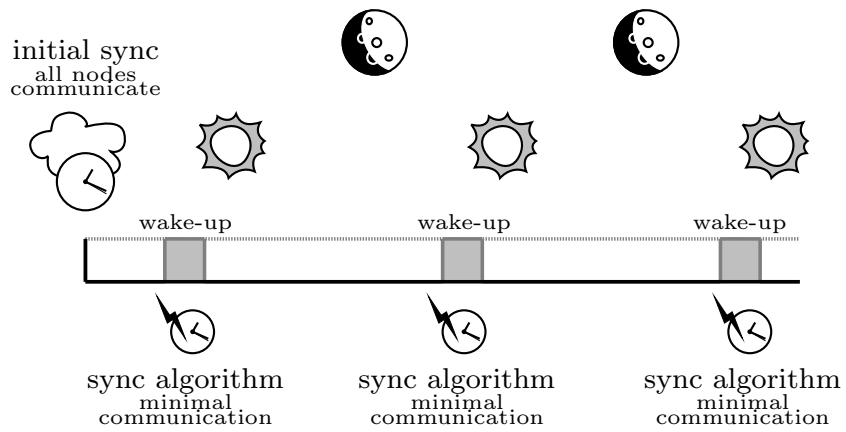


Figure 3.2: The testbed that inspires the choice of the numerical parameters for the presented simulations. Agents wake up daily for a small time window to perform their tasks (sensing, computing,...). In this window they are also allowed to perform few communications to keep their clocks synchronized.

Proof. From Theorem 3.1.6, any norm of $P(t)$ converges exponentially to zero if $\alpha < \bar{\alpha}$. Therefore, for the trace norm,

$$\mathrm{Tr}(P) = \mathrm{Tr}\left(\mathbb{E}\left[\begin{bmatrix}\bar{x}\\ \bar{v}\end{bmatrix}\begin{bmatrix}\bar{x}(t)^T & \bar{v}(t)^T\end{bmatrix}\right]\right) = \mathbb{E}\left[\left\|\begin{matrix}\bar{x}\\ \bar{v}\end{matrix}\right\|_2^2\right] \leq Ce^{-2\eta t}$$

for some positive C and η. Moreover,

$$\int \left\|\begin{matrix}\bar{x}\\ \bar{v}\end{matrix}\right\|_2^2 \mathrm{d}\mu \geq \epsilon(t) \int \chi_{\left\|\begin{matrix}\bar{x}\\ \bar{v}\end{matrix}\right\|_2^2 > \epsilon(t)} \mathrm{d}\mu = \epsilon(t)\mathbb{P}\left[\left\|\begin{matrix}\bar{x}\\ \bar{v}\end{matrix}\right\|_2^2 \geq \epsilon(t)\right]$$

where $\mathrm{d}\mu$ is the probability measure and $\epsilon(t)$ is a positive function of t (dependance on t has been omitted in the vector norms). By taking $\epsilon(t) = e^{-\eta t}$, one obtains

$$\mathbb{P}\left[\left\|\begin{matrix}\bar{x}(t)\\ \bar{v}(t)\end{matrix}\right\|_2^2 \geq e^{-\eta t}\right] \leq Ce^{-\eta t}.$$

Therefore, by Borell-Cantelli's lemma, w. p. 1 it will eventually hold $\left\|\begin{matrix}\bar{x}\\ \bar{v}\end{matrix}\right\|_2^2 \leq e^{-\eta t}$. ∎

Simulation results

As a testbed for the proposed algorithm, the following scenario has been considered, inspired by the wireless sensors power control that has been presented earlier in this section.

In a network of N agents, any agent can communicate with any other agent in a bi-directional fashion. Suppose that at regular time intervals the agents wake up and the communication between a randomly choosen node and one of its neighbors takes place. Based on the received data, the two nodes run the proposed algorithm (see Figure 3.2).

This way, for example, a network of battery-powered sensors can keep their clocks synchronized and reduce significantly the fraction of the time they have to be active for communication. The more precise the synchronization is, the thinner the active time window can be and the longer their batteries will last. Moreover, it is critical that the communication is kept to a minimum, to reduce energy consumption and to save communication resouces.

The numerical values for the parameters of the model come from a hypotetical case study of commercially available wireless sensors waking up once a day, initialized at deployment time with a synchronization error in the order of 1 second, and with relative clock speed drifts in the order of few seconds a month (characteristic of quartz oscillators).

Table 3.1: Simulation clock parameters

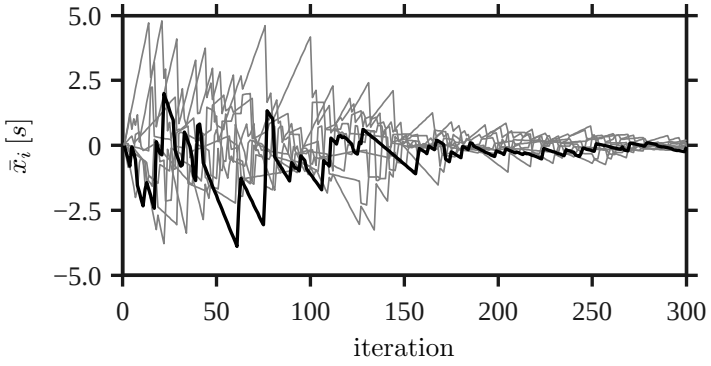
time interval between updates	T	1 day = 86400 s
clock offset standard deviation	σ_x	$10^{-5}T$
clock skew standard deviation	σ_v	10^{-6}

Figure 3.3 shows how the algorithm works for a small network of $N = 10$ clocks of this type.

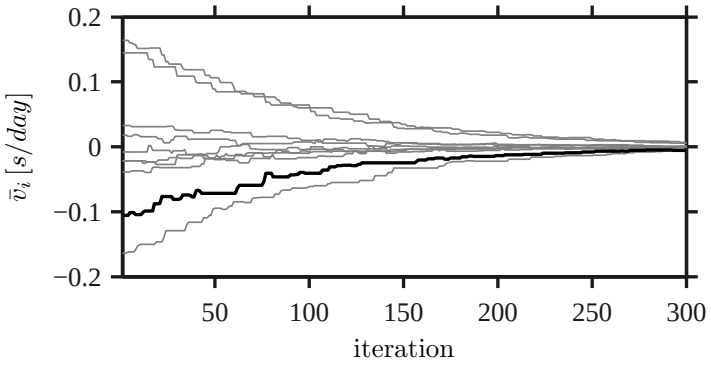
The following numerical simulation confirms that the bound on α is meaningful. In Figure 3.4a the mean square error of the clocks has been plotted for different values of α. One can see as the $\alpha_{\max}$ obtained in Theorem 3.1.6 is indeed critical in determining the convergence of the error. Just by choosing a value of α smaller or larger than $\alpha_{\max}$ by 10%, the behavior of the system is qualitatively different and corresponds to what the analysis says.

In this plot and in all the other plots of the mean square error, the curve represents an average curve over 100 realizations of the algorithm.

In Figure 3.4b the mean square error is plotted for different values of α, in a network of size $N = 50$. Note that there is a first transient in which the drift errors make the synchronization error to increase. After this transient the error start decreasing at a rate that depends on α (and an optimal α could then be searched). Due to the absence of quantization,



(a) Clock offsets



(b) Skew errors

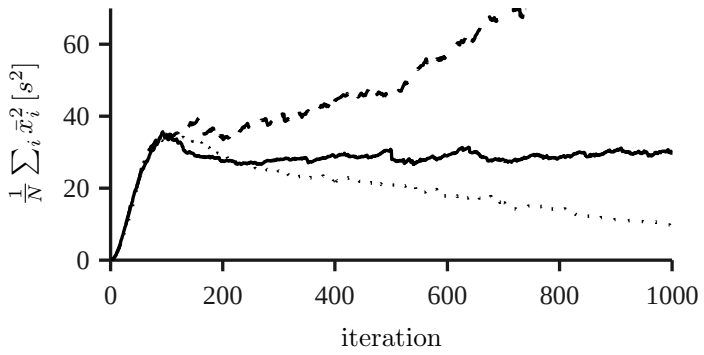
Figure 3.3: Single run of the algorithm with $N = 10$ and $\alpha = 0.01$. The clock offset and the skew error of one specific node have been highlighted.

noise, and time-variance of the drifts, the error then decreases to zero.

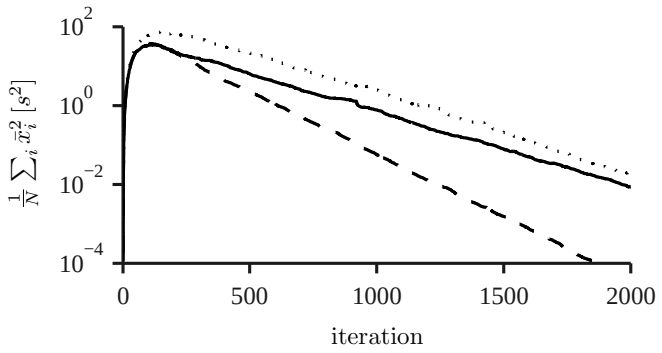
In Figure 3.5 the behavior of networks of different sizes is illustrated. $\alpha = 0.5\alpha_{\max}$ has been choosen for all the cases. It's easy to see that the bound on α that guarantees stability, makes the algorithm not scalable. That is, as the number of agents increases, the rate of convergence becomes slower (more than linearly in the network size). Moreover, the transient becomes unacceptable if the algorithm relies on synchronization to be implementable. Indeed, if the clock error becomes larger than the wake-up time window, communication becomes impossible for that node. This scalability issue is addressed later in this section.

Random geometric communication graphs

In many real networks the communication graph cannot be considered complete. For example when agents are spatially deployed and signal strength decreases with distance, it is reasonable to assume that the communication graph is a *geometric graph*, that is a



(a) $\alpha = 0.9\alpha_{\max}$ (dotted),
$\alpha = 1.1\alpha_{\max}$ (dashed),
$\alpha = \alpha_{\max}$ (continuous),



(b) $\alpha = 0.1\alpha_{\max}$ (dotted),
$\alpha = 0.5\alpha_{\max}$ (dashed),
$\alpha = 0.7\alpha_{\max}$ (continuous)

Figure 3.4: Average of the squared clock errors for different values of α ($N = 50$).

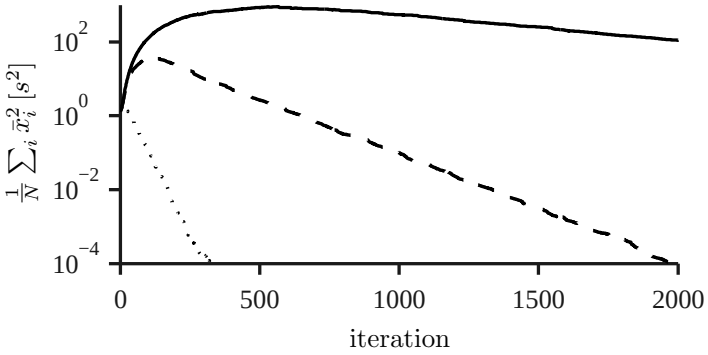
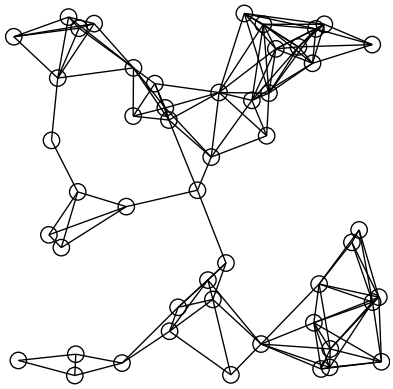
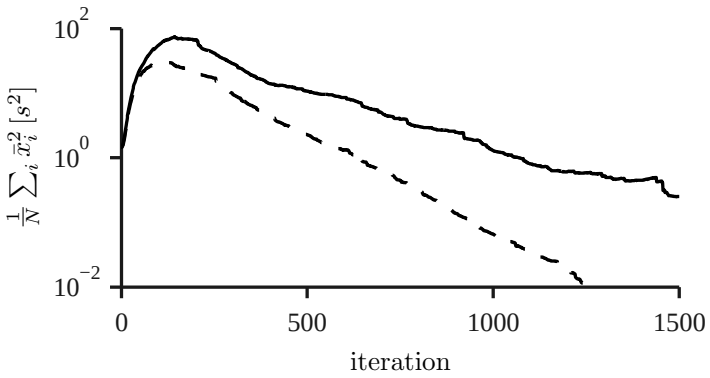


Figure 3.5: Average of the squared clock errors for networks of size $N = 10$ (dotted), $N = 50$ (dashed), and $N = 250$ (continuous). In all cases $\alpha = 0.5\alpha_{\max}$.



(a) Random geometric graph ($N = 50$). The nodes' average degree is 5.7.



(b) Average of the squared clock errors for the graph in (a) (continuous) and for the complete graph (dashed). $\alpha = 0.01$.

Figure 3.6: Simulation of the algorithm behavior for a random geometric graph.

bidirectional edge exists between two nodes when their distance is smaller than a certain threshold. One example of a geometric graph is represented in Figure 3.6a, while in Figure 3.6b it is plotted the mean square error obtained by running the synchronization algorithm on that graph.

Scalability issues

The result stated in Theorem 3.1.6 seems to say that, to preserve stability of the whole systems, the parameter α has to decrease as the inverse of the number of nodes. Simulations show that the rate of convergence of the clocks to a consensus value depends also on the value of the parameter α, and therefore having smaller α's correspond to slower convergence of the system and unacceptable transients that can make the algorithm unimplementable. In

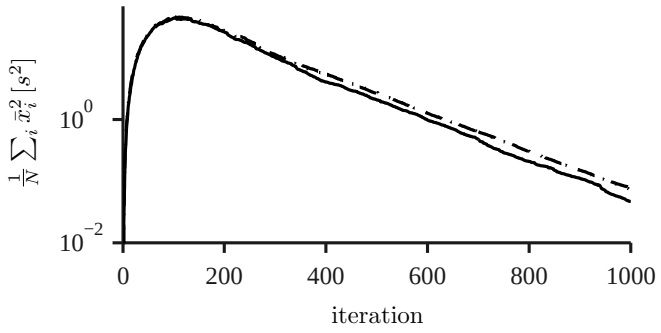


Figure 3.7: Average of the squared clock errors for networks of size $N = 50$ (continuous), $N = 500$ (dashed), and $N = 5000$ (dotted). $\alpha = 0.01$ and 2% of the total edges are activated at the same time.

this sense the system is therefore not scalable.

Multiple symmetric gossip

This scalability problem can be tackled by implementing *multiple gossip communications*, i.e. allowing for more than one edges to be activated at each time step. Indeed, the fact that the clock synchronization algorithm is not scalable mainly depends on the fact that at each iteration only two nodes update their state, while the others keep integrating their local skew error. There is no way to overcome this problem by choosing an appropriate α, on the contrary Theorem 3.1.6 states that α has to go to zero when N grows, to preserve stability.

Consider then the following modification of the algorithm. Suppose that at every iteration each node decides whether to communicate or not with probability p. If it opts for communication, then one of its neighbors is randomly choosen and a symmetric gossip communication takes place. This way, at every iteration, the expected number of communications that take place is pN. Note moreover that this solution is easier to implement on a large network than the original solution of activating one only edge in the whole system. The resulting process is a Bernoulli process, and the time between successive communication of a certain node is now described by a geometric random variable independent from N.

Figure 3.7 shows the behavior of the synchronization error for networks of different sizes, with $p = 0.02$ (that is, having in expectation 2% of the nodes starting a communication at each iteration). $\alpha = 0.5\bar{\alpha}_{\max}$ has been choosen, where $\bar{\alpha}_{\max}$ is the bound for a stabilizing α for a network of size $1/p$, that is a network that would have one only edge communicating (in expectation) with the given p. The plot clearly shows how the behavior of the synchronization error becomes independent from the number of agents.

Comments

The gossip-like PI synchronization algorithm presented in this section proves to be effective in achieving time synchronization and clock skew correction in a network of communicating agents. At the same time, the algorithm is leaderless, extremely lightweight in communication requirements, randomized, and because of its semplicity (and linearity) it could also be studied in terms of robustness and noise rejection. The issue of scalability has been rigorously analyzed, showing how convergence and performances are affected. This issue is tackled by allowing for more than one node to estabilish a connection with a neighbor at the same time, making the algorithm performance independent from the size of the network, and also allowing an easier implementation in a completely distributed scenario.

Among the issues that characterize NCS, the following have been addressed for this specific application:

- **Multi agent architecture** – the problem of clock synchronization can be casted into the problem of controlling a large number of unstable systems, with unknown initial condition, and with partial observations of their state (as clock offsets can be estimated via a simple exchange of timestamps, while skew differences cannot);
- **Scalability** – for the proposed algorithm, an analytical study of the scalability properties has been carried out, showing that the performances scale badly with the number of nodes, and suggesting one possible solution to this problem;
- **Communication constraints** – the proposed algorithm requires gossip communication among nodes, which can be implemented quite easily in WSN and requires little coordination; however, the analytic study of the algorithm stability required the relaxation of the communication constrained induced by a given communication graph (assuming instead that every pair of node is allowed to communicate), while a simulative approach is needed to include these constraints;
- **Robustness to systems changes** – while the possibility of system changes has not been explicitely considered in this application, it is worth noticing that the algorithm allows the insertion (or removal) of agents without any modification.

3.2 Wireless channel parameter estimation

Wireless sensor networks (WSNs), i.e. networks of smart devices that can sense, compute and exchange information wirelessly with their neighbors, are becoming very popular because of their promise to revolutionize many engineering areas involving monitoring and control. These sensor networks have been object of extensive technological and scientifc research in the last ten years (see for example the Special issue on sensor networks and applications 2003). The strength of WSNs resides in their flexibility and scalability, since the same hardware and software can be rapidly reconfigured and adapted to manage rather different applications, from ambient monitoring to people tracking, from industrial control to energy management in buildings. However, many challenges ranging from hardware design, to real-time middleware prototyping, from data routing protocols to distributed signal processing still remain to be solved before WSNs can become really ubiquitous and successful.

Among the most promising applications for WSNs is localization and target tracking (Hu and Evans, 2004; Lorincz and Welsh, 2005). In fact, the wireless radio in each node of the WSN can be used not only to communicate but also to measure the radio signal strength associated with the received packet. Since the signal strength is a function of the locations of the transmitter and the receiver, this information can be used to estimate their relative position. There are two main approaches to target tracking: map-based and range-based. In the map-based approach the position of the moving target is obtained by finding the most likely location which matches the recorded signal strength based on previously learned maps, as in Lorincz and Welsh (2005); Spagnolini and Bosisio (2005). This strategy can be a good solution but it requires extensive work to learn the maps. Differently, the range-based algorithms first try to estimate relative distances based on simple models of the wireless channel and then they estimate the position by triangulation, similarly to the GPS system where the static nodes of the WSN play the role of the satellites in the GPS. This approach, presented in Hu and Evans (2004), requires a higher node density than the map-based one, but it does not require an extensive learning phase.

Most of the existing literature on range-based tracking focuses on triangulation algorithms where the wireless channel model parameters are assumed to be known or are identified off-line by collecting all data in some centralized location, as in Patwari, Hero, Perkins, Correal, and O'Dea (2003). Unfortunately, these parameters are strongly dependent on the environment (Goldsmith, 2005), in particular indoor, therefore it is desirable to identify them in-situ, possibly using distributed algorithms suitable for the limited computational resources of the WSN nodes. Moreover, the radio signal strength measure provided by the radio chips of the sensor nodes are not very precise, mainly due to uncalibrated offsets in the receiving nodes. As a consequence, the estimated distance can be constantly biased in some nodes,

thus degrading tracking performance. Therefore, it is necessary to devise some strategies to compensate these offsets (see for example Whitehouse and Culler 2002).

In Bolognani, Del Favero, Schenato, and Varagnolo (2008, 2010) the use of consensus algorithms has been proposed for automatically calibrating the sensors without the use of a reference node, and for least–square–estimating the wireless channel parameters without collecting data at any central location. The two proposed algorithms are however very general. In fact, any problem concerned with the compensation of measurement offsets affecting a network of sensors, where the offsets cannot be directly measured but only pairwise differences of offsets are available, can be solved with the proposed approach. Moreover, any least-square parameter identification problem based on distributed measurements can be solved in a distributed fashion via the proposed algorithm, without collecting all data in single location to compute the optimal (centralized) solution.

WSN model

Connectivity and communication

Let a WSN be modeled as a set of N nodes $\mathcal{V} = \{1, \dots, N\}$. Since nodes communicate using a wireless channel, the transmission is not reliable, i.e. there is a non–zero packet loss probability. This communication unreliability can be modeled by the *connectivity matrix* $C \in \mathbb{R}^{N \times N}$, where $C_{ij} \in [0,1]$ is the probability that node j can successfully transmit a message to node i. Since the wireless channel is approximately symmetric, it is reasonable to assume that $C = C^T$ and $C_{ii} = 1, \forall i$. The connectivity matrix induces the *c-connectivity graph* $\mathcal{G}_c = (\mathcal{V}, \mathcal{E}_c)$, defined as the graph s.t. (i,j) belongs to the set of the edges $\mathcal{E}_c$ if and only if $c_{ij} \geq c$. This graph is undirected since the matrix C is symmetric.

The matrix C can be experimentally estimated by letting each node broadcast M packets at random instants (with retransmission intervals sufficiently large in order to avoid collisions), making each node i record the number m_{ij} of messages received by each node j and setting $\hat{c}_{ij} = \frac{m_{ij}}{M}$. Subsequently, each node communicates its $\hat{c}_{ij}$ to its neighbors and sets $c_{ij} = \frac{\hat{c}_{ij} + \hat{c}_{ji}}{2}$ since $\hat{c}_{ij}$ and $\hat{c}_{ji}$ are different being empirical means.

The formulation presented in Section 2.1 can then be adopted to describe the possible communication strategies that can be implemented in a WSN. In particular, because of the specific character of wireless communication, *broadcast* communication can be implemented with no extra effort with respect to *asymmetric gossip*, while exchanging information in a *symmetric gossip* manner requires some extra coordination to deal with the possibility of packet collision (and consequent drop).

Wireless Channel Model

For the implementation of range-based localization algorithms, a model for the behavior of the wireless channel between two nodes in terms of received power P_{rx} (in dBm) is needed. The Radio Signal Strength Indicator (RSSI) measured by a generic node i after having successfully received a packet sent by the generic node j can be modeled in the most general form as:

$$P_{\text{rx}}^{ij} = f\left(P_{\text{tx}}^{j}, x_i, x_j, i, j, t\right) \tag{3.10}$$

where P_{tx}^{j} (in dBm) is the nominal transmitted power, i and j are the indices of the receiver and the transmitter nodes respectively, $x_i, x_j \in \mathbb{R}^3$ are their spatial positions and t is the time when the communication occurs. Equation (3.10) can be decomposed into simpler elements which takes into account different effects. Combining the models of each element, described in Gudmundson (1991) and Goldsmith (2005), and adding parts due to offsets in the RSSI measurements and in the power transmission, yields the following model:

$$P_{\text{rx}}^{ij} = P_{\text{tx}}^{j} + r_j + f_{\text{pl}}(\|x_i - x_j\|) + f_{\text{sf}}(x_i, x_j) + f_{\text{a}}(x_i, x_j) + v_{\text{ff}}(t) + o_i \tag{3.11}$$

where:

- P_{tx}^{j} is the nominal transmitted power and r_j is the *transmission offset* between the nominal and the effectively transmitted power; this factor is due to fabrication mismatches and it is assumed to be constant in time;
- $f_{\text{pl}}(\cdot)$ represents the *path loss* effect, modeled as in Goldsmith (2005):

  $$f_{\text{pl}}(d_{ij}) = \beta - 10\gamma \log_{10}\left(d_{ij}\right) \tag{3.12}$$

  where $d_{ij} = \|x_i - x_j\|$ is the euclidean distance between the nodes i and j, β represents the radio receiver gain at a nominal distance of $d = 1\text{m}$, and γ is the loss factor (in an ideal outdoor setting $\gamma \approx 2$); the parameters β and γ are in general unknown since they depend on the specific environment where the WSN is placed;
- $f_{\text{sf}}(\cdot)$ takes into account the *shadow fading* and other slow fading components; it is assumed (see Gudmundson 1991) to be symmetric – i.e. $f_{\text{sf}}(x_i, x_j) = f_{\text{sf}}(x_j, x_i)$ – and Gaussian with a spatial correlation dependent on the difference between the distances of the various points; more precisely, $\mathbb{E}_x\left[f_{\text{sf}}(x_i, x_j)\right] = 0$ and $\mathbb{E}_x\left[(f_{\text{sf}}(x_i, x_j))^2\right] = \sigma_{\text{sf}}^2$ are the spatial mean and variance where the expectation is performed w.r.t. to the random node positions; moreover, let x_i, x_j^a and x_i, x_j^b two different configurations s.t.

$\delta = \|x_j^a - x_j^b\|$, then the spatial correlation is:

$$\mathbb{E}_x \left[f_{sf}\left(x_i, x_j^a\right) f_{sf}\left(x_i, x_j^b\right) \right] = \sigma_{\text{sf}}^2 \rho_D^{\delta/D}$$

where ρ_D is a parameter and D is the typical correlation distance; note that the expected value of f_{sf} is assumed to be zero;

- $f_{\text{a}}(\cdot)$ represents the *channel asymmetry* factor; it is due to non symmetric reflections, and can be modeled as a Gaussian r.v. with zero-mean and covariance $\mathbb{E}_x \left[f_{\text{a}}^2(x_i, x_j) \right] = \sigma_{\text{a}}^2$;
- $v_{\text{ff}}(\cdot)$ represents the *fast fading* component that can be modeled (see Goldsmith 2005) as a white temporal noise with zero-mean and covariance $\mathbb{E}_t \left[v_{\text{ff}}^2(t) \right] = \sigma_{\text{ff}}^2$;
- $o_i(\cdot)$ represents the *receiver offset* that affects the measured received strength due to fabrication mismatches in the radio chip.

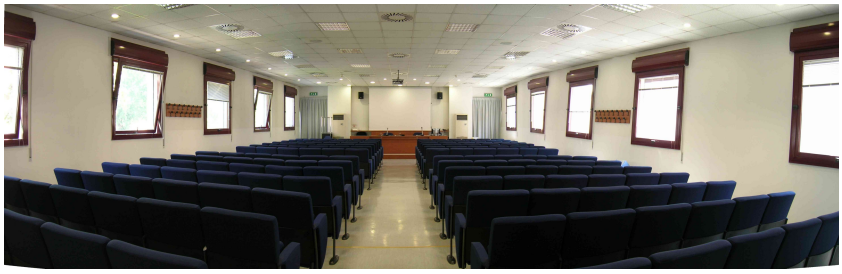
Equation (3.11) is a general model for the wireless channel, in which parameters depend on the physical environment where the WSN is placed and on the sensors under consideration. It is important to remark that these parameters are not known in advance but they need to be estimated on-site.

Experimental testbed and model validation

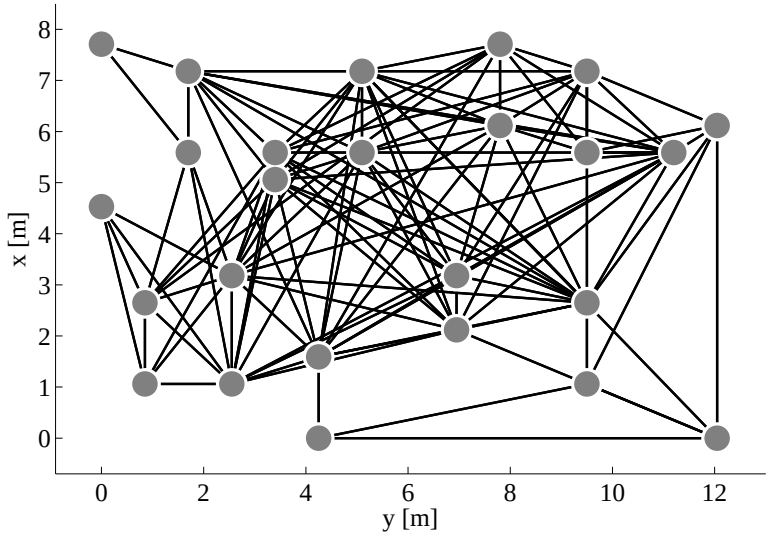
In Bolognani et al. (2010) an experimental testbed has been set up to validate the model. Numerical values for some of the parameters have been obtained, showing that some of the terms in (3.11) are negligible, and pointing out which other parameters have instead to be compensated or estimated.

The experimental data used in Bolognani et al. (2010) consist in a series of measurements relative to packet transmissions and receptions performed by a net of 25 Tmote-Sky nodes equipped with the Chipcon CC2420 RF Transceiver (CC2420 Datasheet) and powered by alkaline batteries. These nodes were randomly placed inside a single conference room of 15m $\times$ 10m at about 50cm from the ground (Figure 3.8a). The relative position of the nodes is shown in Figure 3.8b.

Each node implemented the randomized broadcast communication using the same transmission power P_{tx} and intercommunication interval $\tau = 15$s so that the event of a packet collision is negligible. Each node sent a fixed number of packets $M = 500$, each one including the sender node ID, and also stored a table with the total number of messages received from their neighbors and the corresponding RSSI measures P_{rx}^{ij}.



(a) Picture of the testbed room: Aula Magna "A. Lepschy", Department of Information Engineering, University of Padova, Italy.



(b) Network topology and node displacement. Only edges with empirical packet loss smaller than 25% are displayed.

Figure 3.8: The experimental testbed used in Bolognani et al. (2010) for collecting the data used in model validation and in the testing of the proposed algorithms.

These tables have been then collected for off-line data processing. In particular, from these data the connectivity matrix C has been constructed. Given the short distance among nodes, each node received at least one packet from any other node, however the empirical packet reception probability was different. In fact, the c-connectivity graph $\mathcal{G}_c$ obtained for $c = 0.75$ (i.e. removing poor links with showed an empirical packet loss probability greater than 25%) is not the complete graph, even if it is still connected, as shown in Figure 3.8b.

The different parameters of the wireless channel model (3.11) have been estimated using the various $P_{\mathrm{rx}}^{ij}(t)$ collected from the nodes.

The transmission power offsets r_j can be directly measured substituting the antenna of the nodes with a probe connected to a power meter. Measurements made by Zanca and Zorzi (2008) on the set of the nodes used for the experimental data showed that these offsets are

negligible, i.e. $r_i \approx 0$, $\forall i$.

Then for every link $(i,j) \in \mathcal{E}$ in the connectivity graph, the empirical mean of the received power $\bar{P}_{\text{rx}}^{ij} = \frac{1}{M_{ij}} \sum_t P_{\text{rx}}^{ij}(t)$, and the empirical variance $(\hat{\sigma}_{ff}^{ij})^2 = \frac{1}{M_{ij}} \sum_t (P_{rx}^{ij}(t) - \bar{P}_{rx}^{ij})^2$ have been computed, where M_{ij} is the total number of messages received. The empirical variance of the measurement associated to each link is due to fast fading, and therefore

$$\sigma_{\text{ff}}^2 = \frac{1}{|\mathcal{E}|} \sum_{(i,j)\in\mathcal{E}} \left(\sigma_{\text{ff}}^{ij}\right)^2.$$

The measurements $\bar{P}_{\text{rx}}^{ij}$ include the effects of path loss, shadow-fading, channel asymmetry and reception offsets. The contribution of the channel asymmetry and reception offset together can be estimated by noting that the path loss and the shadow fading are symmetric, i.e. $\Delta\bar{P}_{\text{rx}}^{ij} = \bar{P}_{\text{rx}}^{ij} - \bar{P}_{\text{rx}}^{ji} = f_a^{ij} - f_a^{ji} + o_i - o_j$, where for ease of notation $f_a^{ij} = f_a(x_i, x_j)$. Moreover, it is possible to remove the effects of the offsets by noting that $\Delta\bar{P}_{\text{rx}}^{ijk} = \Delta\bar{P}_{\text{rx}}^{ij} + \Delta\bar{P}_{\text{rx}}^{jk} + \Delta\bar{P}_{\text{rx}}^{ki} = f_a^{ij} - f_a^{ji} + f_a^{jk} - f_a^{kj} + f_a^{ki} - f_a^{ik}$. Experiments showed that $\Delta\bar{P}_{\text{rx}}^{ijk}$ has approximately zero-mean over the set $\mathcal{C}$ of all the independent feasible cycles (i,j,k). Since the nodes are sufficiently far from each other and the shadow fading correlation distance $D \approx 10$cm, all f_a^{ij} can be considered uncorrelated, and therefore the covariance of the channel asymmetry results to be:

$$\sigma_{\text{a}}^2 = \frac{1}{6|\mathcal{C}|} \sum_{(i,j,k)\in\mathcal{C}} \left(\Delta\bar{P}_{\text{rx}}^{ijk}\right)^2.$$

By assuming independence between channel asymmetry components f_{a}^{ij} and offsets o_i, the offset variance σ_{o}^2 can be estimated from the following formula:

$$2\sigma_{\text{o}}^2 + 2\sigma_{\text{a}}^2 = \frac{1}{|\mathcal{E}|} \sum_{(i,j)\in\mathcal{E}} \left(\Delta\bar{P}_{\text{rx}}^{ij}\right)^2.$$

Finally, the parameters $\theta = [\beta \ \ \gamma]^T$ of the path loss model remain to be estimated. Suppose that sensors have been calibrated by adding a compensating offset $\hat{o}_i$ such that $o_i + \hat{o}_i = \alpha$ for all nodes. Averaging all sensor readings received from the same node removes the effect of fast-fading, therefore the calibrated average received power $\hat{P}_{\text{rx}}^{ij} = \bar{P}_{\text{rx}}^{ij} + \hat{o}_i$ is given by:

$$\hat{P}_{\text{rx}}^{ij} = P_{\text{tx}} + \beta - 10\gamma \log\left(d_{ij}\right) + f_{\text{sf}}^{ij} + f_{\text{a}}^{ij} + \alpha$$

where $f_{\text{sf}}^{ij} = f_{\text{sf}}(x_i, x_j)$. Since α is constant, its contribution can be included in the estimated β, therefore assuming $\alpha = 0$. Shadow fading f_{sf}^{ij} and channel asymmetry f_{a}^{ij} are unknown but they can be assumed to be independent zero-mean disturbances, therefore it is possible to find the best mean square estimate of the unknown parameter as $\hat{\theta}_{LS} = (A^T A)^{-1} A^T b$, where

β [dBm]	γ [dBm]	σ_{sf} [dBm]	σ_{a} [dBm]	σ_{ff} [dBm]	σ_{o} [dBm]	r_i [dBm]
-45.7	1.76	3.78	0.16	1.31	1.01	≈ 0

Table 3.2: Estimated parameters of the channel model (3.11) via centralized processing of the experimental data collected in the testbed.

$A = [a_1 \ \dots \ a_M]^T$, $b = [b_1 \ \dots \ b_M]$, and $M = |\mathcal{E}|$. The generic elements of matrix A and vector b are $a_m = [1 \ \ -10\log(d_{ij})]^T$ and $b_m = (\hat{P}_{\text{rx}}^{ij} - P_{\text{tx}})$, where $d_{ij} = \|x_i - x_j\|$ and $\hat{P}_{\text{rx}}^{ij}$ are known. Figure 3.9 shows the identified path-loss model and all collected pairs $(\hat{P}_{\text{rx}}^{ij}, d_{ij})$. The residues obtained from the path-loss model correspond to the variance due to the shadow fading and channel asymmetry, i.e.:

$$\sigma_{\text{a}}^2 + \sigma_{\text{sf}}^2 = \frac{1}{|\mathcal{E}|} \left\| A\hat{\theta}_{LS} - b \right\|^2 .$$

Table 3.2 summarizes the estimated parameters of the model (3.11) based on the experimental data collected. Note that the terms due to the asymmetry in the channel, f_{a}^{ij}, can be safely neglected when compared to the slow fading terms, f_{sf}^{ij}, i.e. the wireless channel is indeed symmetric.

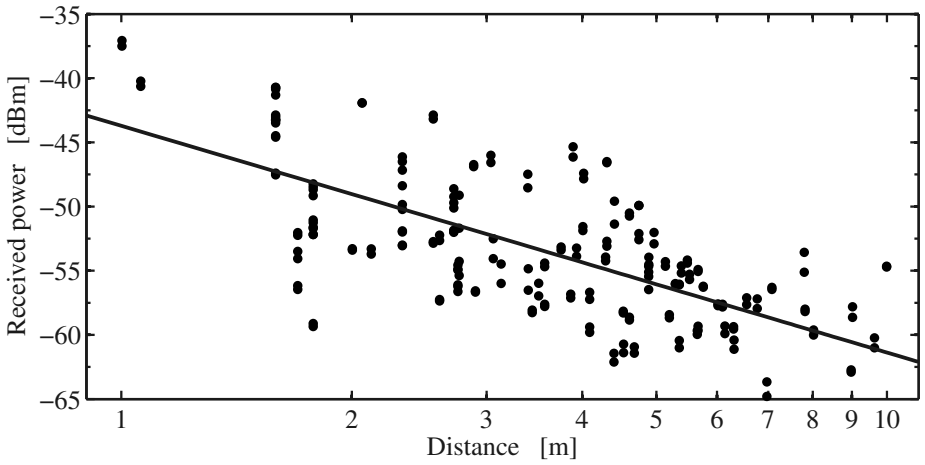


Figure 3.9: Estimated path-loss model for the wireless channel of the testbed using standard least square estimation. The continuous line represents the path-loss function, while the dots are the collected data.

Consensus-based sensor calibration

Experimental evidence indicates that sensor offsets o_i in the nodes are not negligible and can be substantially large for some node (up to 6 dBm according to CC2420 Datasheet).

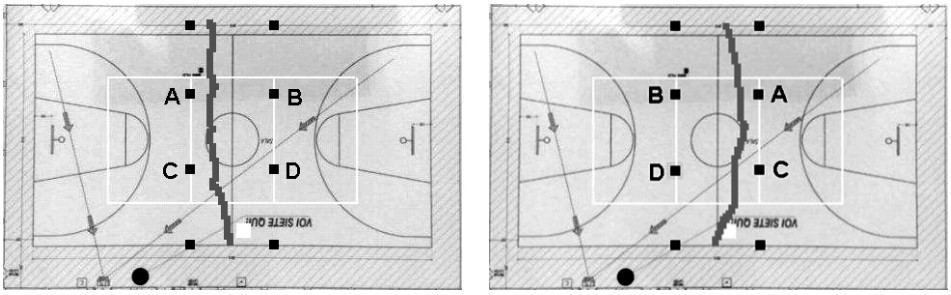


Figure 3.10: Experiment inside a basketball court showing the effects of reception offsets in WSN tracking when nodes are swapped. True trajectory in both panels is the court cenerline.

The effect of this offset is to bias the estimate of the distance between two nodes, which is particularly harmful in tracking application. In fact, if one node has a high offset o_i, then its estimated distance from a moving node is smaller than the true distance. Since the unknown location of a moving target is obtained, similarly to the GPS, by triangulating its position from three or more static nodes whose position is known, the estimated position will be closer to the node with high offset o_i than it should be. This is particularly clear in Figure 3.10, which reports a tracking experiment where the moving node to be tracked is following a straight line (the basketball court centerline) between two rows of nodes of a WSN. However, its estimated trajectory is not straight but it is bent to the left (left panel). When the two central nodes on one side are swapped with the other side, the estimated trajectory is now bent to the right, thus clearly showing a problem due to uncalibrated offsets.

An offset calibration algorithm would ideally add a compensation offset $\hat{o}_i$ to the reading of the received power such that $o_i + \hat{o}_i = 0$, and then use the compensated received power $\widehat{P}_{\text{rx}}^{ij} = P_{\text{rx}}^{ij} + \hat{o}_i$ to estimate the relative distances.

In Bolognani et al. (2010) a fully distributed and simple strategy has been presented, showing how it is possible to estimate the offsets o_i of every node up to a constant. Indeed, as it is not possible to directly measure o_i of each node, the proposed algorithm is only capable to find offset estimates such that

$$o_i + \hat{o}_i = \alpha, \quad \alpha \approx 0$$

for all nodes. If $\alpha \neq 0$ such strategy does not compensate the offset, but at least all nodes either underestimate or overestimate the relative distance similarly, therefore after a GPS-like triangulation stage these errors should partially cancel out. Moreover, if a channel identification procedure follows, then the common offset α is going to be absorbed in the identified parameters, becoming ininfluent.

The proposed algorithm, which is now going to be illustrated, is based of the crucial step

of casting the offset calibration problem into a consensus problem. Consider a static WSN where the nodes are at fixed positions and transmit at the same power P_{tx}. The average received power $\bar{P}_{\text{rx}}^{ij}$ satisfies the following:

$$\bar{P}_{\text{rx}}^{ij} = \frac{1}{T}\sum_{t=1}^{T} P_{\text{rx}}^{ij}\left(P_{\text{tx}}, \mathbf{x}_i, \mathbf{x}_j, i, j, t\right) = f_{ij} + o_i + f_{ij}^{\text{a}} + r_j + \frac{1}{T}\sum_{t=1}^{T} v_{\text{ff}}(t) \approx f_{ij} + o_i \tag{3.13}$$

where P_{rx}^{ij} is modeled as in (3.11), $f_{ij} = P_{\text{tx}} + f_{\text{pl}}(\|\mathbf{x}_j - \mathbf{x}_i\|) + f_{\text{sf}}(\mathbf{x}_j, \mathbf{x}_i)$, and $f_{ij}^{\text{a}} = f_{\text{a}}(\mathbf{x}_i, \mathbf{x}_j)$. The approximation is based on parameters in Table 3.2 which imply that

$$\left| f_{ij}^{\text{a}} + r_j + \frac{1}{T}\sum_{t=1}^{T} v_{\text{ff}}(t) \right| \ll |o_i|$$

for T sufficiently large, $v_{\text{ff}}(t)$ being white noise. Note that f_{ij} is symmetric, i.e. $f_{ij} = f_{ji}$.

The next theorem shows how the problem of compensating the offset o_i can be casted as a consensus problem.

Theorem 3.2.1. *Consider the c-connectivity graph $\mathcal{G}_c = (\mathcal{V}, \mathcal{E}_c)$ of a WSN, and let $Q(t) \sim \mathcal{G}_c$ be a sequence of stochastic matrices that solves the (probabilistic) consensus problem. Assume that $y_{ij} = f_{ij} + o_i$ is the average received signal strength by node i from node j, and consider the following algorithm:*

$$\hat{o}_i(0) = 0, \quad i \in \mathcal{V} = \{1, \dots, N\} \tag{3.14}$$

$$\hat{o}_i(t+1) = \hat{o}_i(t) + \sum_{j \in \mathcal{N}(i)} q_{ij}(t)\big(y_{ji} - y_{ij} + \hat{o}_j(t) - \hat{o}_i(t)\big) \tag{3.15}$$

where $q_{ij}(t) = [Q]_{ij}(t)$. Then $\lim_{t\to\infty} o_i + \hat{o}_i(t) = \alpha$ where $\alpha \in [\min_i(o_i), \max_i(o_i)]$. If in addition $Q(t)$ is doubly stochastic $\forall t$, then $\alpha = \frac{1}{N}\sum_{i\in\mathcal{V}} o_i$.

Proof. Define the new variables $x_i(t) = o_i + \hat{o}_i(t)$. From this definition it follows that $x_i(0) = o_i + \hat{o}_i(0) = o_i$. Moreover, as $Q(t)$ are stochastic matrices, (3.15) can be rewritten as

follows:

$$\begin{aligned}
\hat{o}_i(t+1)+o_i &= \hat{o}_i(t)+o_i+\sum_{j\in\mathcal{N}(i)} q_{ij}(t)\big(f_{ji}+o_j-f_{ij}-o_i+\hat{o}_j(t)-\hat{o}_i(t)\big)\\
x_i(t+1) &= x_i(t)+\sum_{j\in\mathcal{N}(i)} q_{ij}(t)\big(x_j(t)-x_i(t)\big)\\
&= \Big(1-\sum_{j\in\mathcal{N}(i)} q_{ij}(t)\Big)x_i(t)+\sum_{j\in\mathcal{N}(i)} q_{ij}(t)x_j(t)\\
&= q_{ii}(t)x_i(t)+\sum_{j\in\mathcal{N}(i)} q_{ij}(t)x_j(t)
\end{aligned}$$

The last equation can be written in compact form as $x(t+1)=Q(t)x(t)$. Since $Q(t)$ solves the (probabilistic) consensus problem, then $\lim_{t\to\infty} x_i(t)=\alpha$. The claim that $\alpha\in[\min_i(o_i),\max_i(o_i)]$ follows from the property that if Q is a stochastic matrix, and therefore $\max_i[Qx]_i\le\max_i x_i$ and $\min_i[Qx]_i\ge\min_i x_i$ (Doob, 1953). ■

The previous theorem indicates how the offsets o_i can be compensated without knowing their values. Also, it is not necessary to know the exact value of f_{ij} since it is symmetric. In practice the assumption $\bar{P}_{\text{rx}}^{ij}=y_{ij}=f_{ij}+o_i$ is not exact, leading to an oscillating steady state behavior in the consensus algorithm that is evident in the numerical simulations.

It is interesting to consider the problem of choosing appropriate $Q(t)$ to have $\alpha\approx 0$, which is the ideal solution. This may be desiderable, for example, when the offset calibration algorithm is not followed by any identification stage and is therefore required to achieve precise offset removal. It is reasonable to assume that the offsets o_i of the radio chips are on average null, have some dispersion due to imperfect fabrication and are independent, i.e. $\mathbb{E}\left[o_i\right]=\mu_o=0$, $\mathbb{E}\left[o_i^2\right]=\sigma_o^2$, and $\mathbb{E}\left[o_io_j\right]=\mathbb{E}\left[o_i\right]\mathbb{E}\left[o_j\right]=0$. It is well known that the best estimate of the mean μ_o given a set of offsets is $\mathbb{E}\left[\mu_o\,|\,o_1,\ldots,o_N\right]=\frac{1}{N}\sum_{i\in\mathcal{V}}o_i=\alpha^*$ which has the property that $\mathbb{E}\left[\alpha^*\right]=\mu_o=0$ and $\mathbb{E}\left[(\alpha^*)^2\right]=\frac{\sigma_o^2}{N}$, i.e. the average consensus is the strategy for which α is closer to zero in mean square sense and its error becomes smaller and smaller as the number of nodes N increases. Note however that, although the best choice for the compensation of offsets o_i is to choose doubly stochastic $Q(t)$'s, this can be difficult to be enforced in a WSN, since it requires synchronization among the nodes and compensation for packet loss.

Simulations based on experimental data

The proposed algorithm for distributed offset calibration has been tested off–line on the same set of data collected during the experimental setup described before. A c-connectivity

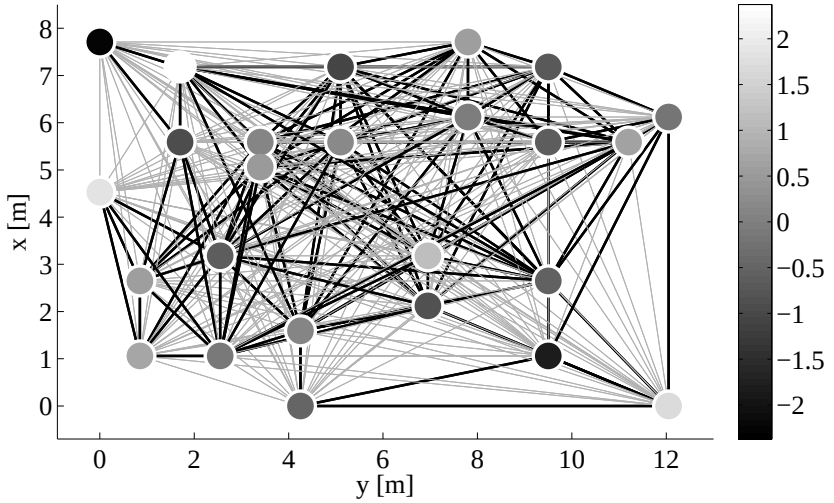
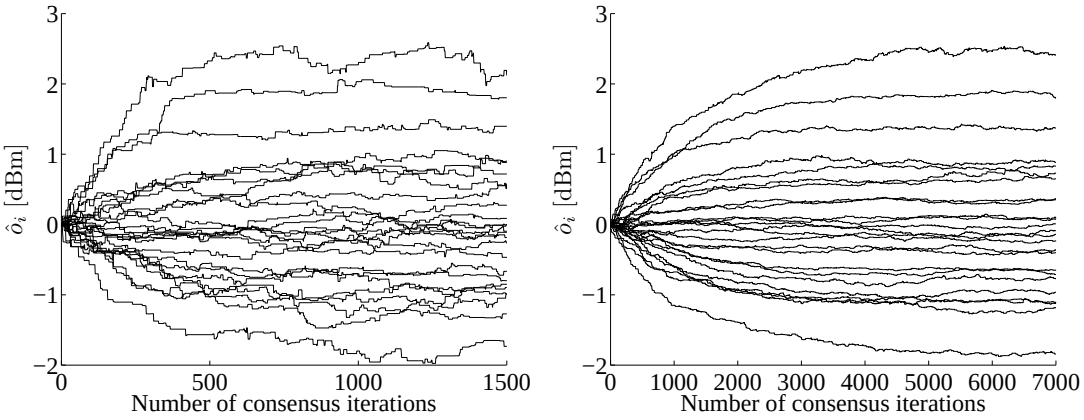


Figure 3.11: Network topology and node displacement for c-connectivity graph with $c = 0.1$. Nodes' grey intensity represents the estimated offset $\hat{o}_i$ after calibration. Black and grey edges represent the edges used for training and validation data sets, respectively.

graph $\mathcal{G}_c$ with $c = 0.1$, i.e. all links which received at least 10% of the packets, have been considered. Differently from the graph with $c = 0.75$ shown in Figure 3.8b, the resulting graph with $c = 0.1$ reported in Figure 3.11 is complete, i.e. all edges exist. The set of all the edges has been divided into two subsets: the first subset of edges (60% of the total edges, in black in Figure 3.11) has been used for the estimation of the node offsets. Therefore the proposed distributed sensor calibration algorithm has been executed on the data collected on these edges. In particular, the calibration algorithm was set with $y_{ij} = \bar{P}_{\text{rx}}^{ij}$ corresponding to these edges. The second subset (40% of the total edges, in grey in Figure 3.11) has been used in a second stage for validation purposes: the asymmetric difference $(\bar{P}^{ij} + \hat{o}_i) - (\bar{P}^{ji} + \hat{o}_j)$ has been evaluated on the data collected on this subset. This approach allows to both evaluate the effect of the offset removal in a rigorous way, and to validate at the same time the proposed model.

Randomized broadcast consensus has been simulated on the graph $\mathcal{G}_c$ using the experimental data and including i.i.d. packet loss failure set by the connectivity matrix C. Figure 3.12 shows the behavior of the consensus algorithm for a specific realization with two different values of the weight parameter w in matrices $Q(t)$ as defined in Section 2.2. The steady state compensation offsets $\hat{o}_i(\infty)$ are displayed in Figure 3.11 where the gray intensity of the nodes is proportional to $\hat{o}_i(\infty)$. Since the true node offsets o_i are unknown it is not possible to plot the behavior of $x_i(t) = o_i + \hat{o}_i(t)$, which are the variables that should converge to a common value, however the fact that all $\hat{o}_i$ converge to a steady state is an indication of correct functioning. It is also interesting to note the effect of unmodeled



(a) $w = 0.05$ **(b)** $w = 0.01$

Figure 3.12: Offset estimation $\hat{o}_i$ for each node of the considered WSN using randomized broadcast consensus for different values of the consensus weight w.

measurement noise arising from having neglected channel asymmetry and fast fading. In fact for larger w, i.e. for larger weight on the off-diagonal terms in the consensus matrix, the oscillation at steady state is not negligible, i.e. a large w tends to amplify noise. On the other hand, a small w leads to slower rate of convergence, thus indicating a tradeoff between convergence rate and noise sensitivity. Note also that the magnitude of steady state values of $\hat{o}_i$ is consistent with the a-priori dispersion indicated by the standard deviation σ_o reported in Table 3.2.

In order to evaluate the effectiveness of the offset calibration, the channel asymmetry after calibration has been tested on a validation set different from the set used for computing the offsets $\hat{o}_i$. The results of this second stage has been plotted in Figure 3.13. The white bars represent the distribution of $|\bar{P}^{ij} - \bar{P}^{ji}|$ on the validation edges, before the distributed sensor calibration algorithm is executed. The black bars, instead, show the distribution of $|(\bar{P}^{ij} + \hat{o}_i) - (\bar{P}^{ji} + \hat{o}_j)|$ after the algorithm has run. The offset reduction clearly appears. After the calibration, 56% of the validation edges have an asymmetric difference smaller than $0.5dBm$ (it was 24% before calibration), while 88% of them have an absolute error smaller than $1dBm$ (it was 50% before calibration). After the offset removal algorithm, almost all the measurements (99.4% of them) are affected by an asymmetric error smaller than $2dBm$.

The importance of offset removal in the received power measurements is evident when these measurements are used for wireless–based localization. In fact, relative distance is estimated by inverting the path-loss function based on the calibrated measured power $\hat{P}_{\text{rx}}^{ij}$:

$$\hat{d}_{ij} = 10^{\frac{P_{\text{tx}}^{j} - \hat{P}_{\text{rx}}^{ij} + \beta}{10\gamma}} = 10^{\frac{P_{\text{tx}}^{j} - P_{\text{rx}}^{ij} + \hat{o}_i + \beta}{10\gamma}}.$$

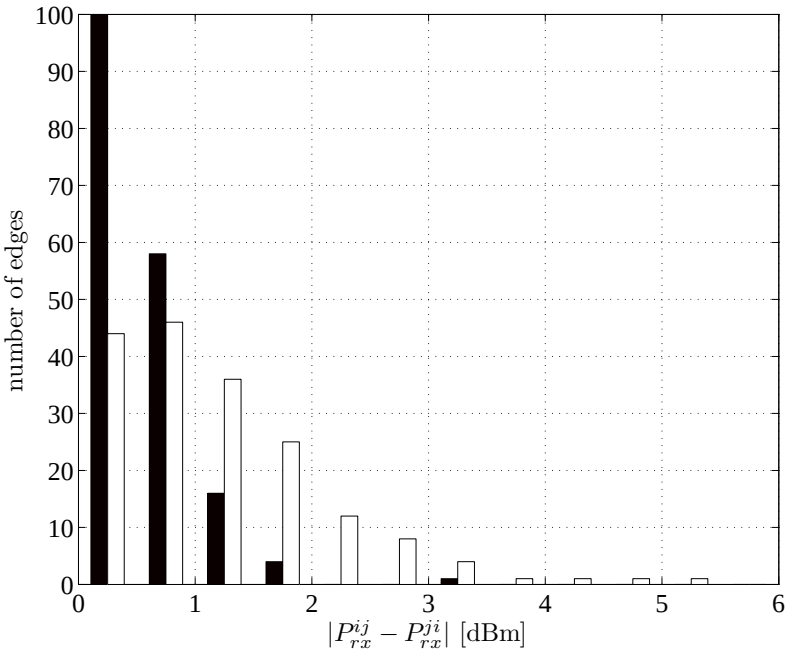


Figure 3.13: Asymmetric error distribution before ($|\tilde{P}_{rx}^{ij} - \tilde{P}_{rx}^{ji}|$, white) and after ($|\tilde{P}_{rx}^{ij} + \hat{o}_i - \tilde{P}_{rx}^{ji} - \hat{o}_j|$, black) distributed sensor calibration.

If the calibration offset $\hat{o}_i$ is not included in the previous formula, there can be measurements errors up to 6 dBm due to uncalibrated offsets, as Figure 3.13 suggests. In fact, a systematic calibration error of 6 dBm corresponds to an uncertainty range from 0.9 m to 4.4 m when estimating the relative position of a node at 2 m, and to a practically useless estimation when the node is farther. An error of 1 dBm, on the other hand, corresponds to a error in the distance of only 28 cm for a 2 m long link, and to a 1.4 m error when the node is at 10 m distance.

Consensus-based least square parameteric identification

A least square parameter identification (LSPI) problem arises when a set of data $\mathcal{D} = \{(a_m, b_m), m = 1, \ldots, M\}$ is available, where $a_m \in \mathbb{R}^\ell$ and $b_m \in \mathbb{R}$. The data set is generated according to the model $a_m^T \theta = b_m + v_m$, where $\theta \in \mathbb{R}^\ell$ is a parameter vector to be estimated and $v_m \in \mathbb{R}$ is an unknown error. Define the matrix $A \in \mathbb{R}^{M \times \ell}, A = [a_1 \ldots a_M]^T$ and the vectors $b, v \in \mathbb{R}^M, b = [b_1 \ldots b_M]^T, v = [v_1 \ldots v_M]^T$. The least square identification of the parameter θ is defined as follows:

$$\hat{\theta}_{LS} = \arg\min_{\theta} \|v\| = \arg\min_{\theta} \|A\theta - b\| = \left(A^T A\right)^{-1} A^T b \tag{3.16}$$

where non-singularity of the matrix A^TA has been assumed.

The problem that arises when implementing LSPI in wireless sensor networks, for example to identify the channel parameters, is that the data $\mathcal{D}$ is dispersed among agents of the network, and a straightforward implementation of (3.16) would require that a central node collects all the data, build the arrays A and b, compute the matrix multiplications and the matrix inversion, and then share the resulting estimate in the network.

This approach scales badly with the number of nodes, requires a leader node that takes care of the data processing, and is not robust against changes in the number of nodes and appearance of new data. Distributing the LSPI algorithm among the agents would be beneficial in this sense, and preferrable.

The next theorem shows how the least square parameter estimate (3.16) can be computed distributively over a network of agents via a consensus based strategy.

Theorem 3.2.2. *Let $\mathcal{G}_c = (\mathcal{V}, \mathcal{E}_c)$ be the c-connectivity graph associated to a communication network with N nodes, i.e. $N = |\mathcal{V}|$, and let $\mathcal{D}(i) = \{(a_m, b_m)\}$ the partition of the whole data set $\mathcal{D}$ available to i-th node, satisfying $\mathcal{D}(i) \cap \mathcal{D}(j) = \emptyset, i \neq j$, $\cup_{i\in\mathcal{V}}\mathcal{D}(i) = \mathcal{D}$, $|\mathcal{D}(i)| = M_i$ and $|\mathcal{D}| = M = \sum_{i\in\mathcal{V}} M_i$. Suppose that $Q(t)$ are stochastic matrices consistent with the graph, i.e. $Q(t) \sim \mathcal{G}_c, \forall t$, which solve the (probabilistic) average consensus problem. Let $x_i^A \in \mathbb{R}^{\ell\times\ell}$ and $x_i^b \in \mathbb{R}^\ell$ for $i = 1, \dots, N$ be auxiliary variables individually stored in the nodes, and consider the following algorithm:*

$$x_i^A(0) = \sum_{m\in\mathcal{D}(i)} a_m a_m^T, \quad \forall i \in \mathcal{V} \tag{3.17}$$

$$x_i^b(0) = \sum_{m\in\mathcal{D}(i)} a_m b_m, \quad \forall i \in \mathcal{V} \tag{3.18}$$

$$x_i^k(t+1) = q_{ii}(t)x_i^k(t) + \sum_{j\in\mathcal{N}(i)} q_{ij}(t)x_j^k(t), \quad k = A, b \tag{3.19}$$

$$\eta_i(t) = \left(x_i^A(t)\right)^{-1} x_i^b(t) \tag{3.20}$$

where $q_{ij}(t) = [Q]_{ij}(t)$. Then

$$\lim_{t\to\infty} \eta_i(t) = \hat{\theta}_{LS}, \quad \forall i \in \mathcal{V}.$$

Proof. Consider the matrix $S = A^TA = \sum_{i=1}^M a_i a_i^T$ and the vector $d = A^T b = \sum_{i=1}^M a_i b_i$; therefore $\hat{\theta}_{LS} = S^{-1}d$. As the matrices $Q(t)$ solve the (probabilistic) average consensus problem, it holds $\lim_{t\to\infty} x_i^A(t) = \frac{1}{N}\sum_{i=1}^N x_i^A(0) = \frac{1}{N}\sum_{i=1}^M a_i a_i^T = \frac{1}{N}S$ and similarly

$\lim_{t\to\infty} x_i^b(t) = \frac{1}{N}\sum_{i=1}^{N} x_i^b(0) = \frac{1}{N}\sum_{i=1}^{M} a_i b_i = \frac{1}{N}c$. By continuity

$$\lim_{t\to\infty} \eta_i(t) = \left(\frac{1}{N}S\right)^{-1}\frac{1}{N}d = S^{-1}d = \hat{\theta}_{LS}.$$

Note that the sums are defined from 1 to M due to the structure of (3.17) and (3.18). ■

This theorem shows that LSPI can be computed as the solution of a distributed algorithm which does not require the knowledge of the total number of nodes N or the total number of data M available. Moreover, the data can be arbitrarily partitioned among nodes. Since the matrix $S = A^T A$ is symmetric it is not necessary to compute all its ℓ^2 entries, therefore the x_i^A can be reduced to a vector of size $(\ell^2+\ell)/2$. Nonetheless the complexity in terms of communication, i.e. the dimension of the vector of parameters to be averaged, is $O(\ell^2)$ which can be impractical if the dimension ℓ of the unknown parameter θ is large. Strategies that trade-off accuracy in the identification of $\hat{\theta}_{LS}$ for a decrease in communication complexity to $O(\ell)$ will be presented in the next paragraph.

The problem addressed in the previous theorem belong to a more general class of problems that can be solved with consensus algorithms. In fact any optimization problem can be written as:

$$\xi = f\left(\frac{1}{N}\sum_{i\in\mathcal{V}} g_i\left(z_i\right)\right)$$

for some appropriate choice of functions f and g_i, where z_i represents the data available to node i. Some examples of the previous class of problem includes generalized means (Bauso, Giarré, and Pesenti, 2006), χ–consensus (Cortés, 2008) and distributed Kalman filter (Spanos et al., 2005).

Simulations based on experimental data

The algorithm proposed in Theorem 3.2.2 for the distributed computation of least square parameter estimates will now be applied to the problem of identifying the path-loss wireless channel parameters (β,γ) given in (3.12). As mentioned above, these two parameters are used in localization and target tracking algorithms in order to estimate relative distances between the moving node and the nodes of the static WSN. Therefore, it is critical to be able to identify the path-loss parameters in a distributed way, in a manner that is robust to node failure, with minimal exchange of data and low computational power, and without a central unit. It has to be noted that an accurate *a-priori* model for power loss in different indoor environments is almost unavailable (for example γ can vary from 1 to 6 according to the room sizes, the amount of furniture and people and the number of walls that the signal

has to cross in average). Furthermore, the same environment can present a hourly or daily variation of these parameters due to the periodic presence of people populating the indoor spaces (Dominguez-Duran, Claros, Urdiales, and Coslado, 2008). Distributed algorithms with these features can be used to periodically or adaptively identify the channel parameters in a changing environment.

Based on these considerations, the focus of this set of simulations is to compare the performances of three different communication strategies which have different characteristics in terms of rate of convergence, communication complexity and parameter identification accuracy. The first and the second strategies are based on the implementation of the distributed least square identification described in Theorem 3.2.2 using the randomized broadcast and the randomized symmetric gossip, respectively. The third strategy performs the randomized symmetric gossip consensus on local estimates $\hat{\theta}_i$ of the channel parameters vector θ, rather than on the local least-square sufficient statistics $\left(x^A, x^b\right)$ relative to $\left(A^T A,\, A^T b\right)$ of Theorem 3.2.2. Each strategy has its own advantages. In fact, the randomized symmetric gossip guarantees average consensus, therefore it is guaranteed to provide the best identification accuracy since it satisfies the hypotheses of Theorem 3.2.2. Randomized broadcast does not guarantee average consensus, and consequently the best performance, however it is very easy to implement since it needs no coordination between nodes. Moreover it is faster than the symmetric gossip since, on average, there are $d(i)$ updates per iteration compared with only 2 updates for the symmetric gossip. Finally, the strategy based on the average consensus of the local least-square estimates does not guarantee optimal performance nor best speed of convergence, however the number of parameters to be exchanged among nodes is equal to the size ℓ of the parameter vector θ, while for the first two strategies it is proportional to ℓ^2.

For the simulations presented hereafter, the c-connectivity graph $\mathcal{G}_c = (\mathcal{V}, \mathcal{E}_c)$ for $c = 0.75$ has been considered (see Figure 3.8b). The data set $\mathcal{D}(i)$ available to each node i is given by $\mathcal{D}(i) = \{(\bar{P}_{\text{rx}}^{ij}, d_{ij}) \,|\, j \in \mathcal{V}(i)\}$, i.e. all the averaged received power measurements from each neighbor coupled with the corresponding relative distance (note that the distances are assumed to be known by the nodes). The data set of all measurements is indicated with $\mathcal{D} = \cup_{i\in\mathcal{V}}\mathcal{D}(i)$. It has been assumed that an offset calibration procedure has been performed, in order to obtain the compensating offsets $\hat{o}_i$, and that the effect of fast-fading can be neglected since the measurements have been averaged over a large number of packets. Therefore the channel parameters $\theta = [\beta\ \gamma]^T$ can be identified using a least square minimization by setting $a_m = [1\ \ -10\log(d_{ij})]$, $b_m = \bar{P}_{\text{rx}}^{ij} - P_{\text{tx}} + \hat{o}_i$, where $m = 1, \ldots, M$ indicates a generic data element, and $M = |\mathcal{D}| = |\mathcal{E}_c|$. Using the same terminology of Theorem 3.2.2, $\hat{\theta}_{LS}$ denotes the centralized least-square estimate using the complete data set $\mathcal{D}$. On the other hand, $\hat{\theta}_{LS}^i$ denotes the least-square estimate performed by the i-th node using

only its data set $\mathcal{D}(i)$, which is the best estimate a node can have without communicating with the others. The performance (in terms of identification accuracy) is based on the residues of the estimate $\hat{\theta}$ given by

$$J(\hat{\theta}) = \left\| A\hat{\theta} - b \right\|.$$

Note that A and b are constructed using the whole data set, and therefore $J(\hat{\theta})$ represents the global residual. Since $\hat{\theta}_{LS} = \arg\min_\theta J(\hat{\theta})$, it is obvious that $J(\hat{\theta}_{LS}) \leq J(\hat{\theta}^i_{LS}), \forall i$ from which it follows $J(\hat{\theta}_{LS}) \leq \frac{1}{N}\sum_{i\in\mathcal{V}} J(\hat{\theta}^i_{LS})$. Being $\eta_i(0) = \hat{\theta}^i_{LS}$, if all $Q(t)$'s are doubly stochastic then from Theorem 3.2.2 it follows that $\lim_{t\to\infty} J(\eta_i(t)) = J(\hat{\theta}_{LS}), \forall i$, and so $\lim_{t\to\infty} \frac{1}{N}\sum_{i\in\mathcal{V}} J(\eta_i(t)) = J(\hat{\theta}_{LS})$.

In the first simulation, the randomized broadcast least-square strategy has been implemented, also including the link failure probabilities of each edge. Figure 3.14 shows the identified channel parameters of all nodes $\eta_i(t) = [\hat{\beta}_i(t)\ \hat{\gamma}_i(t)]^T$ as a function of the number of iterations for a typical realization of the system (thought as the stochastic process of information exchange). It can be seen that the identified parameters of all nodes converge to a common value, however, since broadcast does not guarantee average consensus, identified parameters do not necessarily coincide with the optimal estimate $\hat{\theta}_{LS}$. It is also interesting to note that most of the nodes have already good estimates of the parameters without communicating with the others, since most of them have lots of links and there are only two parameters to estimate. However, there are some nodes which have poor initial estimates, especially the ones on the perimeter of the graph which have few links. Nonetheless, thanks to the consensus algorithm, they rapidly converge to a good value.

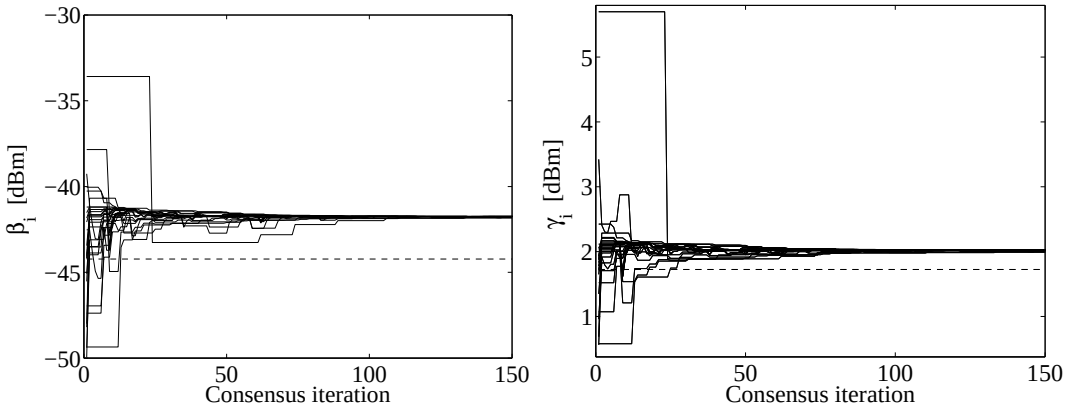


Figure 3.14: Convergence of parameter estimates β_i, γ_i using randomized broadcast least-square consensus and consensus weight $w = 0.5$. The dashed lines are the centralized least squares estimates $\hat{\beta}_{LS}, \hat{\gamma}_{LS}$.

Consensus algorithm	$\mathbb{E}[\bar{J}(\infty)]$	$\max \bar{J}(\infty)$	$\min \bar{J}(\infty)$
Broadcast $w = 0.5$	3.9816	4.1477	3.9320
Broadcast $w = 0.25$	3.9615	4.0919	3.9318
Symmetric Gossip	3.9307	3.9307	3.9307
Average of local estimates	3.9635	3.9635	3.9635

	$J_{Cent.L.S.}$
Centralized LS	3.9307

Table 3.3: Comparison of the mean estimation residual.

In the second set of simulations, shown in Figure 3.15, the rate of convergence and the steady state identification error for the three different strategies described above have been compared. More precisely, the average estimation residual $\bar{J}(k) = \frac{1}{N}\sum_{i\in\mathcal{V}} J(\hat{\theta}_i(k))$ of all nodes has been plotted as a function of iteration error. To reduce the randomness due to the choice of a particular realization of $\{P(t)\}_{t\in\mathbb{N}}$, the expectation $\mathbb{E}\left[\bar{J}(k)\right]$ has been approximately computed as the average of 50 independent extractions of the sequence $\{P(t)\}_{t\in\mathbb{N}}$ and depicted.

In Table 3.3 it is reported also the steady state dispersion of $\bar{J}(k)$ around its mean value, obtained by recording the maximum and the minimum value of $\bar{J}(k)$ over the 50 extractions. In the bottom line the residual of the centralized optimal estimate is also reported for comparison.

Initially, the randomized broadcast least square algorithm has been tested for two different weights w. As already mentioned, larger w leads to faster convergence rates, however it also leads, in mean, to a larger steady state identification error (see Fagnani and Zampieri 2008). The steady state value results also to be strongly realization-dependent, as it can be noticed from to the large dispersion interval. This is due to the fact the first communications tend to bias the final value toward the initial condition of that node. Differently, if w is reduced, then this bias is smoothed out and $\mathbb{E}\left[\bar{J}(k)\right]$ ends up closer to exact average consensus. Also the dispersion of the single realizations with respect to $\mathbb{E}\left[\bar{J}(k)\right]$ reduces. Moreover it has been proved in Fagnani and Zampieri (2008) that the distance of $\mathbb{E}\left[\bar{J}(k)\right]$ form the average consensus decreases by increasing the number of nodes in the network, thus suggesting fast convergence rate with negligible performance degradation as compared, for example, to random symmetric gossip.

The same Figure 3.15 also shows the performance of the randomized symmetric gossip

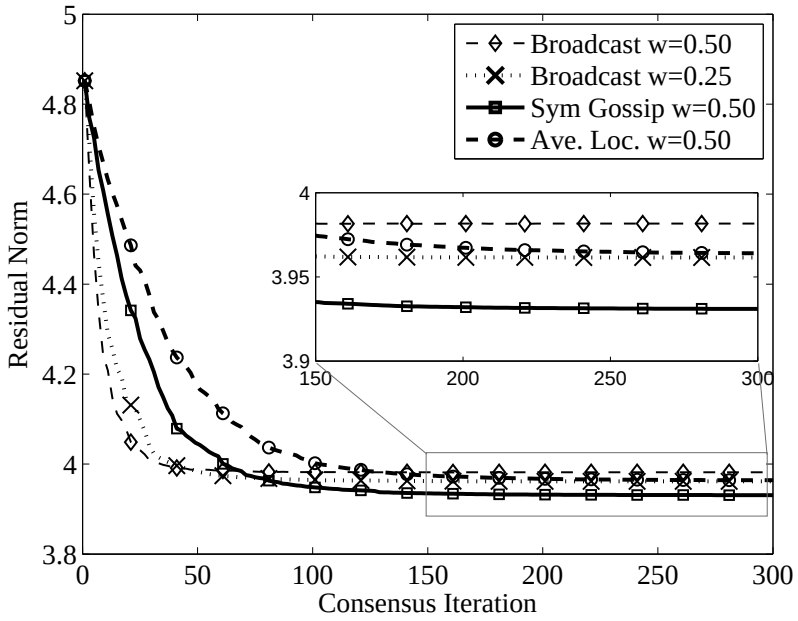


Figure 3.15: Comparison of the mean estimation residual $\mathbb{E}\left[\bar{J}(k)\right]$ for different randomized consensus algorithms. $\mathbb{E}\left[\bar{J}(k)\right]$ is approximately computed as the average of 50 independent extractions of the sequence $\{P(t)\}_{t\in\mathbb{N}}$.

least square algorithm. As expected, the rate of convergence is slower, but the final value converges to the minimum identification error given by the centralized least-square estimate $J(\hat{\theta}_{LS})$. It is important to notice that all the single realizations converge to the exact optimal value, as shown by the fact that there is no dispersion around the mean value (Table 3.3), not only that $\mathbb{E}\left[\bar{J}(k)\right]$ tends to optimal value.

Finally, a randomized gossip algorithm that directly averages the local least-square estimates has been tested. As shown in Figure 3.15, this strategy has the same rate of convergence of the randomized symmetric gossip (which computes the exact centralized least-square solution), but a slightly worse performance. However, in terms of communication complexity this algorithm only requires the exchange of 2 parameters while the exact distributed least square one requires in this example the exchange of 4 parameters. It has to be noticed, though, that if the initial estimates were less reliable (for instance because the graph topology were much less connected) then the distributed least square would behave far better that the simple solution of an average of the local least squares estimations.

Comments

Consensus based algorithms have proven to be effective to remove unknown offsets from the sensor measurements and to identify the parameters of the wireless channel for localization and tracking purposes. However, the derived algorithms are rather general and can be applied in other fields and research areas. Indeed, a wide class of problems can be tackled in very similar ways, such as problems in which the agents have to actually agree on a common estimate of few parameters (like in the least-square fitting), and problems in which every agent has to estimate its own parameter (like in the offset-removal algorithm).

Among the issues that characterize NCS, the following have been addressed for this specific application:

- **Multi agent architecture** – the fact of having a leader-less architecture is the motivating fact for the development of fully distributed algorithms for this application, avoiding collecting all the measured data in any central location with particular data processing capabilities;

- **Distributed information** – making the nodes share (but not collect) the measures that they have collected proves to be beneficial in the distributed least square estimation problem, and indeed the proposed algorithm can guarantee that the node agree eventually on the optimal estimate; in the offset removal algorithm the issue of having dispersed information in the network is even more critical, as the individual nodes cannot estimate their offsets without communicating with their neighbors;

- **Interaction with an underlying physical system** – not only a model of the underlying physical system is needed to define the parameters that are going to be identified by the proposed methods; there is also a strong coupling between the data that are collected for the estimation and the communication and information sharing constraints that have to be satisfied, as both depend on the network topology and on the nodes' position in space;

- **Performances** – the numerical simulation confirm known results on the performances of different consensus algorithms and show that, also in identification problems, a tradeoff often exists between speed of convergence and variance of steady state error.

3.3 Reactive power compensation in smart grids

The electric power system is undergoing a deep change driven by a number of needs, as depicted in Santacana, Rackliffe, Tang, and Feng (2010) and Ipakchi and Albuyeh (2009). Among them is the concern for global environmental issues, which have been also translated into economical incentives and operational constraints. This yielded the introduction of new renewable (but unreliable and intermittent) energy sources into the picture, both in large installations (farms) and in dispersed micro-generators. At the same time the power demand has increased both in its amount and in the quality of the required service, challenging an aging infrastructure that is working close to its full utilization limits. Last, but not least, new technologies like plug-in electric vehicles are likely to become reality in the next future, requiring deep changes in the operational strategies of the grid. All these phenomena are happening extremely fast, compared to the past development of the power grid. Because of the extremely high costs and long times for upgrades to the physical infrastructure (which sometimes are impossible), there is a strong agreement on the fact that the next generation grid will host a large amount of information and communication technologies (ICT) instead of being subject to structural changes. The power grid is by many figures the biggest machine ever built by man, and therefore it should not be surprising that this transition to a *smart grid* is attracting an enormous amount of effort and investments, providing motivating applications for almost all the fields of ICT, including control theory and automation. Hereafter it will be shown how the methodologies of NCS can fit in this scenario, before focusing on a specific application.

Networked control systems in smart power distribution grids

Among the different parts of today's power network (see Figure 3.16), the *distribution grid* has probably the largest room for improvement through automation, information management, and state monitoring.

The following features are desired in a smart power distribution grid, are currently largely missing, and can be efficiently tackled with the approach of networked control systems.

Architecture and microgrids

Today's power distribution grid could well be described as an infrastructure with the only duty of delivering electric power from the transmission grid to the loads, at an appropriate voltage level and with basic quality standards. In a smart grid, a series of new "actors" connects to it: micro-generators that inject power instead of being supplied with, electronic loads with their specific dynamic behavior, "smart" customers that can postpone or shape

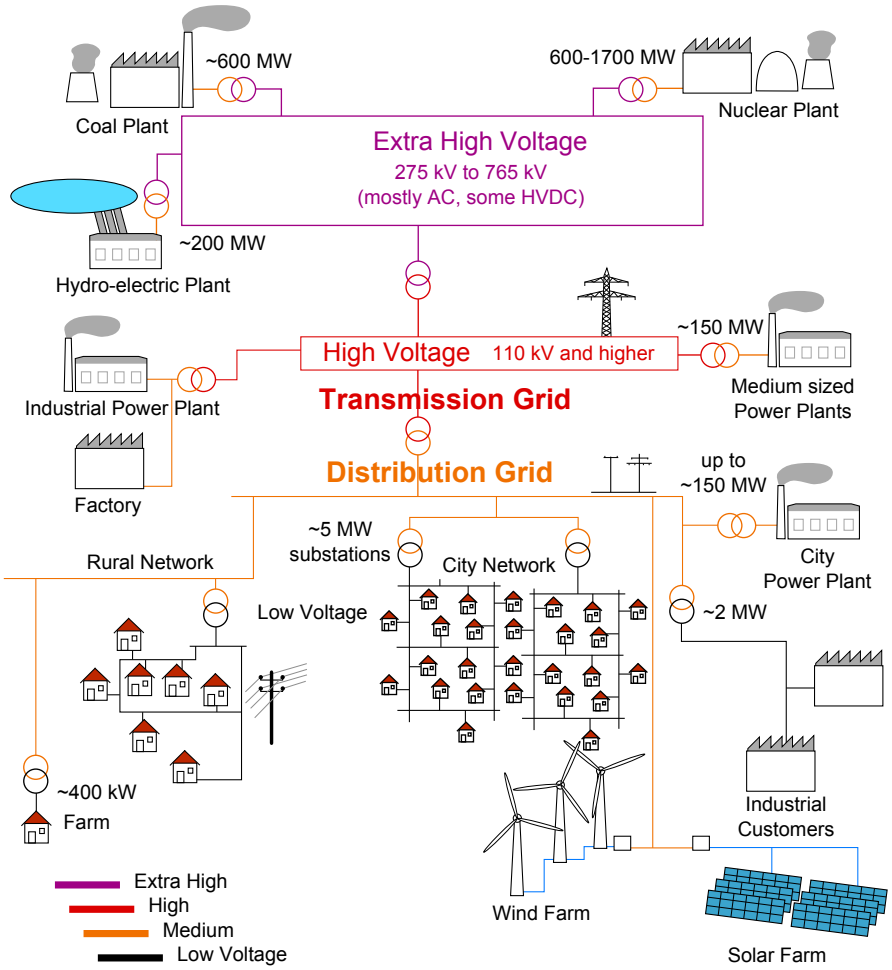


Figure 3.16: Schematic representation of the electrical power grid.

their demand if financially rewarded, plug-in electric vehicles that can act as a moving storage device, and many others. The concept of multiagent architecture is likely to be a necessary approach to this scenario (Massoud Amin and Wollenberg, 2005), while it is still an open question if the management of this architecture should be centralized, hierarchical, or fully distributed. Because these agents are in large number, they are likely to connect and disconnect according to their needs, they are not fully reliable, and they have (at the moment) quite basic communication capabilities, a fully centralized architecture is unpractical. *Smart microgrids* are an interesting approach. Let them be defined as portions of the electrical power distribution network that connect to the transmission grid in one point and that are managed autonomously from the rest of the network. A smart microgrid is generally equipped by some central controllers, but distributed algorithms are also likely to be implemented by

its agents. Cooperation among microgrids and between microgrids and the transmission network operator may also be required. The resemblance of the resulting architecture to the Internet suggests that a *layered structure* should be employed: when many different algorithms are to be executed by the same agents for solving different problems (from electric stability to optimal operation of the electronics, from supply/demand matching to load scheduling), it is important that detrimental interaction is avoided, by making the operation of these algorithms transparent one to the other. The similarity of this approach to data networks (see Chiang et al. 2007 and consider, for example, how error correction, traffic congestion and optimal routing algorithms coexist in the OSI layered model) is therefore extremely motivating and still almost unexplored.

Real time energy pricing

The appearance of new intermittent and unreliable energy sources (wind, solar, combined heat and power generation) requires the design of new mechanisms for supply/demand matching. Real time pricing seems to be the main available tool to shape the consumers' demand for electrical energy, and it is currently used in the wholesale energy market (where however fluctuation in demand and supply are very limited nowadays, and only utilities participate).

It has been shown in Roozbehani, Dahleh, and Mitter (2010a,b) that relaying the wholesale market prices to the customers can easily cause instability in the prices and in the demands. Customers are also likely to exhibit very complex dynamics due to load-shifting and scheduling behavior. Therefore the use of the price signal to shape demand requires the design of proper pricing mechanisms via the tools of control theory (see for example Wang, Kowli, Negrete-Pincetic, Shafieepoorfard, and Meyn 2011) and should take into account the multiagent, heterogeneous, character of the customers.

Distributed generation and storage

Distribution networks are going to host an important number of micro-generators even at a residential level, ranging from solar panels to micro wind turbines, from urban or rural plants powered by waste to combined heat and power generation in big buildings and facilities. Major upgrades to the distribution system control and monitoring protocols may be required to address bidirectional flow patterns and increased loading.

All these small-size generators are connected to the grid via electronic inverters (while larger plants are connected via electro-mechanical generators). This is a great opportunity because the current injected by converters can be precisely controlled with very fast response. At the same time, it can lead to major stability issues, detrimental interaction, and suboptimal

operation of the grid, if they are not properly controlled. Electromechanical converters exhibit a inherent stabilizing behavior due to their mechanical inertia and to the reactance of the network (see Dorfler and Bullo 2010 for a description of the phenomenon via the tools of large scale dynamical systems); inverters, on the other hand, do not exhibit this behavior, and must be properly controlled to achieve frequency stability and correct operation.

This issue is marginal when few micro-generators connect to the grid, but becomes severe when their number and their size are larger. In some cases a microgrid (or in general, a portion of the power distribution network) may also want to disconnect from the transmission grid: because of economical interests, or because the main utility is experiencing a shortage in power generation and asks for load shedding, or because of a blackout on the transmission network. In this cases the aforementioned issues become critical and the microgrid controllers have to face the challenging (and still unresolved) issue of coordinating all the electronic interfaces and stabilizing their dynamic behavior (Lopes, Moreira, and Madureira, 2006).

Network monitoring and state estimation

To approach any of the problems that are being presented so far, it is necessary to obtain an estimate of the current working point and state of the network. This problem is almost nonexistent when power flows only from the point of connection to the transmission grid to the loads, and can be solved via simple monitoring solutions based on few measurements collected on site. If instead bidirectional flow patterns occur (because of distributed generation and storage), and if fault detection and self-healing capabilities are required, then a more informative knowledge of the network state is required.

To achieve so, agents can be instructed to sense the network at their point of connection (voltages and currents) and to build an estimate of the network topology in their neighborhood, for example via time-of-arrival measurements on power line communications. However, the great amount of data available from the agents cannot be collected centrally for different reasons: scalability issue, communication constraint, but also privacy of data and protection about cyber attacks. Distributed estimation and identification algorithms have to be designed and implemented, as this kind of methodologies is currently unavailable in the literature of state estimation in power distribution networks.

Ancillary services

Quality of service in power distribution networks can be considered among the most critical issues that have to be kept in mind in the management of these systems. Different aspect concur to the concept of quality of service in power distribution networks:

- minimization of power distribution losses
- voltage support and regulation
- harmonic suppression and phase balancing
- reliability and robustness to faults
- frequency stability and continuity of the service (blackout avoidance).

All these issues traditionally affected power distribution networks, and network operators have always been requested to provide extremely high reliability (compared, for example, to data networks), as they can be considered without doubts *critical infrastructures*. Distributed micro-generation, unreliability of renewable power supplies, and increased usage of the infrastructures, are worsening this scenario.

At the same time, the increased number of electronic interfaces, sensors, and processing units that are being deployed in the grid, offers a challenging opportunity to deal with these aspects. Indeed, these devices, if properly commanded and coordinated, can provide some on the *ancillary services* that are needed to guarantee a desired quality of the distribution. Because they are dispersed into the distribution network, they are much more effective in providing these services, compared to the traditional scenario in which the distribution network is largely unmonitored and is actuated almost only at its point of connection to the transmission grid.

In the microgrid scenario, this opportunity is even more intriguing, as there are specific operational and economical interests in relaying the burden of these ancillary services to microgrids:

- loads can benefit from a remarkably greater quality of the service if the devices connected to the same microgrid are properly commanded;
- microgrids' users could decide about the quality of the local energy distribution, tailoring it around their needs;
- the microgrid can be managed so that it appears as a very "well-behaved" aggregate user for the rest of the grid, satisfying given specifications, and it can therefore ask for more favorable contracts with the transmission network operator;
- the next generation energy market is likely to include economical compensation for those agents that participate to the provision of ancillary services (see Rodrigues Gomes and Saraiva 2008; Rabiee, Shayanfar, and Amjady 2009).

In the following, the problem of optimal reactive power compensation (one of the most important ancillary services) is described in detail.

Both residential and industrial users belonging to the microgrid may require a sinusoidal current which is not in phase with voltage. A convenient description for that consists in saying that they demand reactive power together with active power, associated with out-of-phase and in-phase components of the current, respectively.

Reactive power is not a "real" physical power, meaning that to produce it there is no energy conversion involved nor fuel costs. Like active power flows, reactive power flows contribute to power losses on the transmission and distribution lines, cause voltage drop, and may lead to grid instability (see Kundur 1994; Dobson, Glavitsch, Liu, Tamura, and Vu 1992). It is therefore preferable to minimize reactive power flows by producing reactive power as close as possible to the users that need it.

The number of compensators in a microgrid can be very large, as the electronic interface of any distributed generator (wind turbines, combined heat and power generators, micro hydroelectric, solar panels) can also produce reactive power at no additional cost. They have however a limit in the maximum amount of reactive power that every compensator can produce, which depends on the thermal limits of the interface (its rated power) and on the amount of active power that the micro-generator is producing at a given time (a larger amount of active power supplied by the micro-generator implies a lower limit of reactive power that can be injected into the network). These compensators are generally unaware of the topology of the whole distribution network, can sense the grid only in their closest neighborhoods, and may be subject to unannounced appearance and disappearance.

For these reasons, the problem of reactive power compensation in a smart microgrid, is quite different and new compared to the classical problems of optimal reactive power flow (OPRF) as depicted for example in Mamandur and Chenoweth (1981), and it requires the methodologies and tools of networked control systems.

Reactive power compensation in smart microgrids

Definition of reactive power

Let identify with a node i any device (inverter) connected to the power distribution network. Let $u_i(t)$ be the voltage at its point of connection, and $i_i(t)$ the injected current. If the network is operating in steady state, voltages and currents are sinusoidal signals at frequency $\omega_0/2\pi$:

$$u_i(t) = U_i \sin(\omega_0 t + \theta_i^u), \qquad i_i(t) = I_i \sin(\omega_0 t + \theta_i^i). \tag{3.21}$$

The active power P and reactive power Q injected (or delivered, if negative) at node i are

$$P_i = U_i I_i \cos\phi, \qquad Q_i = U_i I_i \sin\phi,$$

where ϕ is the phase difference $\theta_i^u - \theta_i^i$.

These power terms can be defined also in the case in which signals are not sinusoidal and in which they can be considered perturbed versions of periodic signals with period $T_0 = 2\pi/\omega_0$. Following Tenti and Mattavelli (2003), let us define the *homo-integral* of a generic function $x(t)$ as

$$\hat{x}(t) = \omega_0 \left(X(t) - \bar{X}(t)\right)$$

where $X(t) = \int_0^t x(\tau)\mathrm{d}\tau$ and $\bar{X}(t) = \frac{1}{T_0}\int_t^{t+T_0} X(t)\mathrm{d}t$.

By introducing the scalar product (function of time)

$$\langle x, y\rangle_t = \frac{1}{T_0}\int_t^{t+T_0} x(\tau)y(\tau)\mathrm{d}\tau \tag{3.22}$$

active and reactive powers can be defined in this more general framework as the instantaneous quantities

$$p_i(t) = \langle u_i, i_i\rangle_t, \qquad q_i(t) = \langle \hat{u}_i, i_i\rangle_t.$$

If $u_i(t)$ and $i_i(t)$ are sinusoidal signals as in (3.21), then $p_i(t) = P_i$ and $q_i(t) = Q_i$.

In the following, the phasorial notation will be adopted: the two signals $u_i(t)$ and $i_i(t)$ are then identified by the complex numbers

$$u_i = U_i e^{j\theta_i^u}, \qquad i_i = I_i e^{j\theta_i^i},$$

while active and reactive power can be expressed as

$$p_i = \mathrm{Re}(u_i i_i^\dagger), \qquad q_i = \mathrm{Im}(u_i i_i^\dagger),$$

where $\dagger$ means complex conjugation. Let $p_i + jq_i$ be defined as the *complex power* injected at the node i.

The optimal reactive power flow problem in a microgrid

Consider a portion of the power distribution network that is managed as a microgrid. Let its devices be described by $\overline{N}$ nodes, and let the electrical connections in the microgrid be described by a tree $\mathcal{T}$ connecting them. Each node corresponds therefore to an agent injecting

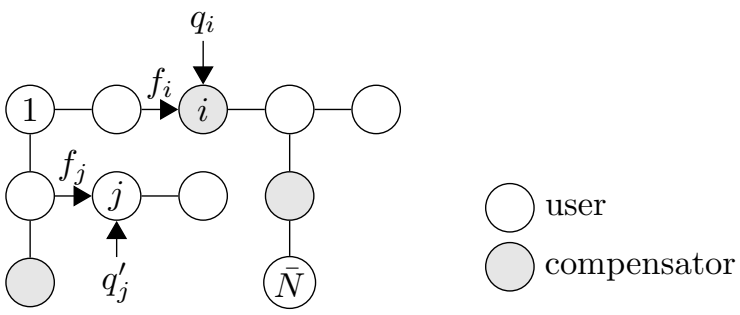


Figure 3.17: Direction of the flows and naming conventions in the tree of users and compensators.

a quantity p_i of active power and a quantity q_i of reactive power into the network. N of them (the compensators, whose indices belong to $\mathcal{C}$) can be commanded to inject a given amount of reactive power, while they inject a fixed amount of active power (the amount generated by the corresponding micro-generator). The other nodes (users, whose indices belong to $\mathcal{U}$) inject (or are supplied with, if negative) a fixed and unknown amount of both active and reactive power. Let $\mathcal{V} = \mathcal{C} \cup \mathcal{U}$ be the set of all the nodes of $\mathcal{T}$, and $\mathcal{E}$ be the set of edges, describing the electrical lines that connect these agents. Edges are oriented outbound from the tree root and indexed as the child node they point tool (see Figure 3.17). Let ζ_i be the current flowing on edge i, and let define by

$$h_i + jf_i = u_i \zeta_i^\dagger$$

the (complex) power flowing on the same edge (which, for quasi-periodic signal, would have been expressed as $\langle u_i(t), \zeta_i(t) \rangle_t + j\langle \hat{u}_i(t), \zeta_i(t) \rangle_t$).

One possible approach to the problem of distributed reactive power compensation in a smart microgrid has been proposed in Tedeschi, Tenti, and Mattavelli (2008), and is sketched in Figure 3.18.

It consists of a centralized controller that measures the reactive power flow at the input port of the microgrid, i.e. where the microgrid connects with the main grid. According to this measurement, the controller produces a reference for the amount of reactive power that has to be produced inside the microgrid. This reference has to be split by a power sharing unit (PSU) among compensators, in a way that minimizes reactive power flows inside the microgrid.

Power distribution losses are a quadratic function of the currents flowing on the lines:

$$L = \sum_{e \in \mathcal{E}} \mathrm{Re}(Z_e) |\zeta_e|^2 = \sum_{e \in \mathcal{E}} \mathrm{Re}(Z_e) \frac{g_e^2 + f_e^2}{|u_{t(e)}|^2},$$

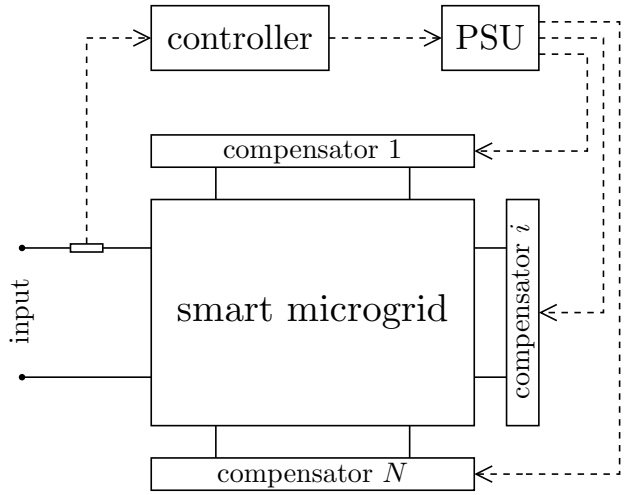


Figure 3.18: The controller structure of Tedeschi et al. (2008).

where Z_e is the impedance of the edge e (which goes linearly with the length of the line).

To construct an optimization problem corresponding to the task of power losses minimization, L must be expressed as a function of the decision variables q_i, $i \in \mathcal{C}$. To do so, the following assumption is needed.

Assumption 3.3.1. The complex power losses $L' = \left[u_{s(e)} - u_{t(e)}\right] \zeta_e^\dagger$ on any edge $e \in \mathcal{E}$, are much smaller than the delivered complex power $h_e + jf_e$.

Assumption 3.3.1 can be translated into two conditions. First,

$$\left|\left[u_{s(e)} - u_{t(e)}\right] \zeta_e^\dagger\right| = |Z_e||\zeta_e|^2 = \frac{|Z_e||h_e + jf_e|^2}{|u_{t(e)}|^2} \ll |h_e + jf_e|$$

implies

$$\frac{|Z_e||h_e + jf_e|}{|u_{t(e)}|^2} \ll 1. \tag{3.23}$$

Second,

$$\left|u_{s(e)} - u_{t(e)}\right| = |Z_e \zeta_e| = \left|Z_e \left(\frac{h_e + jf_e}{u_{t(e)}}\right)^\dagger\right| \ll \left|u_{t(e)}\right|, \tag{3.24}$$

that is, the voltage difference among nodes are small compared to the *nominal voltage* $U_0 \in \mathbb{R}$.

Power losses can then be rewritten by substituting $|u_e|$ with U_0, obtaining:

$$L \approx \sum_{e \in \mathcal{E}} \frac{\mathrm{Re}(Z_e)}{U_0^2} \left(h_e^2 + f_e^2\right).$$

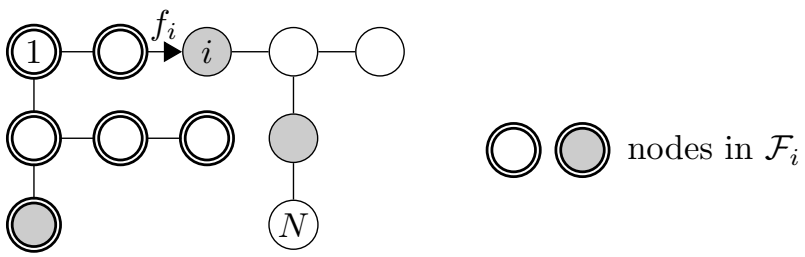


Figure 3.19: The subset $\mathcal{F}_i$, defined as the set of nodes containing the root for which the edge i is the only bridge.

Moreover, if Assumption 3.3.1 holds, then flow conservation law holds separately for active and reactive power. Indeed, Kirchoff current law guarantees that

$$i_i + \zeta_i = \sum_{e\,|\,s(e)=i} \zeta_e$$

for all $i \in \mathcal{V}$. This implies

$$p_i + jq_i + h_i + jf_i = \sum_{e\,|\,s(e)=i} u_s(e)\zeta_e^\dagger \approx \sum_{e\,|\,s(e)=i} u_t(e)\zeta_e^\dagger = \sum_{e\,|\,s(e)=i} h_e + jf_e.$$

Any reactive power flow f_i can then be expressed as the sum of the reactive power injected into the network by a subset $\mathcal{F}_i$ of the agents, as illustrated in Figure 3.19:

$$f_i = f_i(q) = \sum_{j\in\mathcal{F}_i} q_j, \quad \text{for} \quad i = 2,\dots,N, \tag{3.25}$$

while active power flows g_i depend only on the injected active power terms p_i (in a completely similar way). Moreover

$$\sum_{i\in\mathcal{V}} q_i = 0. \tag{3.26}$$

Let q be the vector of all the amounts of reactive power injected by the compensators, and q' those injected by the users that cannot be commanded. Then (3.25) can be expressed via two matrices $K \in \{0,1\}^{\overline{N}-1\times N}$ and $K' \in \{0,1\}^{\overline{N}-1\times\overline{N}-N}$ such that

$$f = Kq + K'q', \tag{3.27}$$

where $f \in \mathbb{R}^{\overline{N}-1}$ is the vector of all the flows f_i.

The optimization problem of having minimal power losses faced by the PSU corresponds

therefore to the cost function

$$F(f_2,\dots,f_N)=\sum_{i=2}^{N}\frac{\mathrm{Re}(Z_i)}{U_0^2}f_i^2. \tag{3.28}$$

subject to the constraints (3.27) and (3.26). By eliminating the first constraint, (3.28) can be rewritten as the following optimization problem:

$$\begin{aligned}\min_{q}\quad & F(q)=q^T\frac{M}{2}q+m^Tq\\ \text{subject to}\quad & \mathbf{1}^Tq=c,\end{aligned} \tag{3.29}$$

where

$$\begin{aligned}M&=2K^TDK,\\ m^T&=2q'^TK'^TDK,\\ D&=\mathrm{diag}\left(\frac{\mathrm{Re}(Z_2)}{U_0^2},\dots,\frac{\mathrm{Re}(Z_N)}{U_0^2}\right)\\ c&=-\mathbf{1}^Tq'.\end{aligned} \tag{3.30}$$

The problem of minimizing reactive power flows inside a microgrid has therefore been expressed as a single-commodity network flow problem, and then casted into the class of quadratic, linearly constrained, optimization problems. Its analytic solution is:

$$q^*=-M^{-1}m+\frac{\mathbf{1}^TM^{-1}m+c}{\mathbf{1}^TM^{-1}\mathbf{1}}M^{-1}\mathbf{1}. \tag{3.31}$$

As said before, the size of this problem (i.e. the number of compensators) can be very large, as the electronic interface of any distributed generator can also produce reactive power at no additional cost. Each of these units is capable of:

- sensing the electric network at its point of connection to the grid;
- performing some amount of computation and data processing;
- communicating with other agents, according to some communication graph $\mathcal{G}_C=(\mathcal{C},\mathcal{E}_C)$ that may or may not coincide with the electric network;
- actuating the system, by injecting a certain amount of reactive power.

The agents may have a partial knowledge of the problem parameters M and m (which depend on the electrical network topology and on the reactive power demand), while no agent knows them entirely.

For these reasons, the implementation of (3.31) is impractical, and iterative distributed algorithms should be considered. In the design of these algorithms coordination among agents must necessarily be achieved, because of the coupling constraint and because the cost function is not separable.

Gradient-driven algorithm are of particular interest because of their effectiveness and robustness in solving this class of problems (see Section 2.3). They however require that an estimate of the gradient is available to the agents. This is actually true for this specific problem, as shown in the following.

Estimation of the gradient of the cost function

The gradient of $F(q)$ can be expressed as

$$\begin{aligned} g(q) &= Mq + m \\ &= 2\left(K^T DKq + KDK'q'\right) \\ &= 2K^T Df(q), \end{aligned}$$

where

$$K^T Df(q) = \begin{bmatrix} \vdots \\ \sum_{e\in\mathcal{E}-\mathcal{P}_i} \frac{\mathrm{Re}(Z_e)}{U_0^2} f_e \\ \vdots \end{bmatrix} \tag{3.32}$$

and $\mathcal{P}_i \subseteq \mathcal{E}$ is the path from the root to node i. It is easy to see that the difference $g_i(q) - g_j(q)$, where $g_i(q)$ and $g_j(q)$ are the elements of $g(q)$ corresponding to compensators i and j, can be expressed as

$$g_i(q) - g_j(q) = 2\sum_{e\in\mathcal{P}_{ij}} \delta_e(i,j)\frac{\mathrm{Re}(Z_e)}{U_0^2} f_e(q) \tag{3.33}$$

where $\mathcal{P}_{ij} \subseteq \mathcal{E}$ is the path from node i to node j, and $\delta_\ell(i,j) \in \{+1,-1\}$ depends on whether the edge i appears in forward or backward direction in the path from i to j (see Figure 3.20).

Therefore if agents are capable of estimating the flows of reactive power on the electrical paths connecting them, they can compute the gradient of the cost function for the optimization problem (3.29), and use it to drive properly designed iterative algorithms. The way in which nodes compute this estimate is part of the wide research topic of state estimation in power distribution networks, and strongly depends on the sensing capabilities that have been implemented in the microgrid.

It is however worth reporting a simple example of how this can be done under specific assumptions on the line impedances.

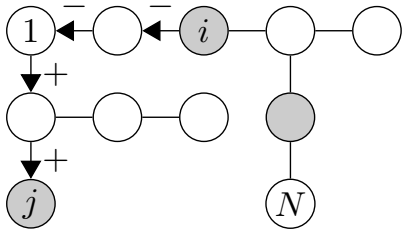


Figure 3.20: Dependence of the gradient g on the flows in the tree; e.g. $g_i - g_j$ equals to the flow on the path from i to j. According to the given convention, flows have to be counted with negative sign when going upwards in the tree, as shown.

Assume that the line inductance is small, compared to its resistance (which is true in power distribution networks, where cables are employed instead of aerial lines). Let therefore $r_e = \mathrm{Re}(Z_e) \approx Z_e$.

Then the voltage drop on an edge $e \in \mathcal{E}$ satisfies

$$u_{s(e)} - u_{t(e)} \approx \frac{r_e}{u_{t(e)}^{\dagger}} \left(h_e + j f_e\right)^{\dagger}. \tag{3.34}$$

Suppose that pair of adjacent nodes can measure the angle between their voltages

$$\Delta\theta_e^u = \theta_{s(e)}^u - \theta_{t(e)}^u \approx \angle\left(\frac{u_{s(e)}}{u_{t(e)}}\right),$$

where the approximation follows from the fact that, by Assumption 3.3.1, $|u_{s(e)}| \approx |u_{t(e)}|$.

By plugging in (3.34), one gets

$$\angle\left(\frac{u_{s(e)}}{u_{t(e)}}\right) = \angle\left(\frac{u_{t(e)} - \frac{r_e}{u_{t(e)}^{\dagger}}(h_e - j f_e)}{u_{t(e)}}\right) = \angle\left(1 - \frac{r_e}{|u_{t(e)}|^2}(h_e - j f_e)\right).$$

Then, by using (3.23),

$$\angle\left(\frac{u_{s(e)}}{u_{t(e)}}\right) \approx \mathrm{Im}\left(1 - \frac{r_e}{|u_{t(e)}|^2}(h_e - j f_e)\right) = \frac{r_e}{|u_{t(e)}|^2} f_e. \tag{3.35}$$

By approximating $|u_{t(e)}|$ with its nominal value U_0, one then obtains

$$\Delta\theta_e^u \approx \frac{r_e}{U_0^2} f_e. \tag{3.36}$$

Therefore, by (3.33) and (3.36), the difference $g_i - g_j$ can be approximated as

$$g_i - g_j \approx 2(\theta_i^u - \theta_j^u)$$

and therefore

$$g(q) = 2\theta^u - \xi \mathbf{1}$$

where θ_u is the vector of all voltage angles (with respect to an arbitrary time reference) and ξ is an unknown scalar. It will be clear later that this uncertainty in the gradient estimate is not harmful, as the term $\xi\mathbf{1}$ is orthogonal to the constraint $\mathbf{1}^T q = c$.

A quasi-Newton method for optimal reactive power compensation

One possible way to tackle this optimization problem has been proposed in Bolognani and Zampieri (2010) and consists in specializing some quite classical tools in convex optimization, quasi-Newton methods, to the case of a linearly constrained problem. The tool of average consensus is then exploited to apply these methods to a large scale complex system.

A general formulation for a gradient-driven optimization algorithm is the following:

$$\begin{aligned} q^+ &= q - \Gamma g(q) \\ \Gamma^+ &= \Phi(q, \Gamma, g(q)). \end{aligned} \tag{3.37}$$

where Γ is an $N \times N$ gain matrix[2].

In the specific problem (3.29), if feasibility of the decision variable is required at any iteration, the gain matrix Γ must satisfy

$$\mathbf{1}^T \Gamma u = 0 \quad \text{for all } u \in \mathbb{R}^N.$$

The next lemma show what it the best choice for Γ, if the Hessian M is fully known.

Lemma 3.3.2 (Constrained Newton algorithm)**.** *Let q be a feasible point for the optimization problem* (3.29)*. Assume*

$$\Gamma = M^{-1} - \frac{M^{-1}\mathbf{1}\mathbf{1}^T M^{-1}}{\mathbf{1}^T M^{-1}\mathbf{1}}. \tag{3.38}$$

Then q^+ defined by (3.37) *is the solution of the constrained optimization problem (1-step convergence).*

[2]In this and in the following sections, the shorter notation $q^+ = q(t_{k+1})$ and $q = q(t_k)$ (and similarly for other quantities) is introduced, when this does not lead to confusion.

Proof. To prove that q^+ is the solution of the constrained optimization problem, it has to be shown that

- q^+ is feasible
- the gradient $g(q^+)$ is orthogonal to the constraint.

The first claim is true for any $\mathbf{\Gamma}$ in the form $H - \frac{H\mathbf{1}\mathbf{1}^T H}{\mathbf{1}^T H \mathbf{1}}$, for any H, as

$$\mathbf{1}^T q^+ = \mathbf{1}^T q - \mathbf{1}^T H g(q) + \frac{\mathbf{1}^T H \mathbf{1}}{\mathbf{1}^T H \mathbf{1}} \mathbf{1}^T H g(q) = c.$$

Moreover

$$\begin{aligned} g(q^+) &= m + Mq^+ \\ &= m + Mq - MM^{-1} g(q) + \frac{MM^{-1}\mathbf{1}}{\mathbf{1}^T M^{-1}\mathbf{1}} \mathbf{1}^T M^{-1} g(q) \\ &= \mathbf{1} \frac{\mathbf{1}^T M^{-1} g(q)}{\mathbf{1}^T M^{-1} \mathbf{1}} \in \mathrm{Im}(\mathbf{1}), \end{aligned}$$

and therefore the second condition is also satisfied. ∎

If an approximation H of the inverse of the Hessian M is available, it can be plugged into (3.38), obtaining the *approximate Newton update step*

$$q^+ = q - \left(H - \frac{H\mathbf{1}\mathbf{1}^T H}{\mathbf{1}^T H \mathbf{1}} \right) g(q) = q - Hg(q) + \frac{\mathbf{1}^T H g(q)}{\mathbf{1}^T H \mathbf{1}} H\mathbf{1}. \tag{3.39}$$

In the update step (3.39) two parts can be recognized:

$$\Delta q = q^+ - q = \Delta q_{\mathrm{desc}} + \Delta q_{\mathrm{proj}}$$

where

- $\Delta q_{\mathrm{desc}} = -Hg(q)$ is a descent step toward the optimum of the unconstrained quadratic problem
- $\Delta q_{\mathrm{proj}} = \frac{\mathbf{1}^T H g(q)}{\mathbf{1}^T H \mathbf{1}} H\mathbf{1}$ projects $q - Hg(q)$ on the constraint (as proof of Lemma 3.3.2 shows, feasibility of q^+ is guaranteed regardless of the choice of H).

According to the available level of knowledge of the system, different approximations of the inverse of M can be used, resulting in different algorithms.

On one hand, Lemma 3.3.2 showed that if M^{-1} is completely known, it can be exploited to obtain the fastest (one-step) convergence. However, this choice requires that node i knows

the whole i-th row of M^{-1}. This may not be possible in large-scale systems, and jeopardizes the possibility of node insertion and removal.

On the other hand, when minimal knowledge is available, a diagonal $H = \alpha I$ can be used, obtaining the specialization of *steepest descent method* to the linearly constrained case.

Unfortunately, the steepest descent method may require a large number of iterations to converge, depending on the condition number of the Hessian M (Boyd and Vandenberghe, 2008). This results to be a major problem in this specific application, where estimating the gradient $g(q)$ for a given q consists in driving the systems into the state q and measuring its steady state response (therefore introducing an implicit trade-off between accuracy and time delay in the measurement).

Quasi-Newton methods (see for example Dennis and Schnabel 1983 and Nocedal and Wright 2006) have instead the useful feature of building an estimate of the inverse of the Hessian from the previous steps of the algorithm. In the framework of complex systems, these methods deserve special attention, as they require minimal knowledge of the problem, they can deal with time-varying structures via their adaptive behavior, and they exhibit faster convergence compared to steepest descent methods.

Consider the following specialization of *Broyden's algorithm* (belonging to the class of quasi-Newton methods) for the constrained optimization problem (3.29):

$$\begin{aligned} q^+ &= q - Gd \\ G^+ &= G + \frac{\left[\Delta q - G\Delta d\right]\Delta d^T}{\Delta d^T \Delta d} \end{aligned} \tag{3.40}$$

where $d = \Omega g$, $\Omega = I - \mathbf{1}\mathbf{1}^T/N$, is the projection of the gradient on the constraint, and

$$\Delta d = d^+ - d, \quad \Delta q = q^+ - q.$$

Suppose that G is initialized as αI, for some $\alpha > 0$, and that it is not updated if $\|\Delta d\| = 0$ (it will be clear in the proof of Theorem 3.3.6 that if $\|\Delta d\| = 0$, then the algorithm has converged.)

It is easy to see that (3.40) is a special case of (3.37), in which $\Gamma = G\Omega$ and in which the rank-1 update for G satisfies the *secant condition* $G^+\Delta d = \Delta q$.

The following lemmas will be helpful in proving the global convergence of this algorithm[3].

Lemma 3.3.3. *For any $G(t_k)$ returned by the algorithm (3.40), and for all $u \in \mathbb{R}^N$,*

$$\mathbf{1}^T u = 0 \quad \Rightarrow \quad \mathbf{1}^T G(t_k) u = 0.$$

[3]For these lemmas and for Theorem 3.3.6 it is convenient to express time dependence explicitly.

Proof. Consider the base case $G(0) = \alpha I$. Then

$$\mathbf{1}^T G(0)u = \alpha \mathbf{1}^T u = 0.$$

Suppose now that the condition is verified for $G(t_k)$, and consider $G(t_{k+1})$. Using the fact that $\Delta q(t_k) = G(t_k)d(t_k)$, and the fact that $\mathbf{1}^T \Delta d(t_k) = 0$ and $\mathbf{1}^T d(t_k) = 0$, one gets

$$\mathbf{1}^T G(t_{k+1})u = \mathbf{1}^T G(t_k)u + \frac{\mathbf{1}^T \Delta q(t_k)\Delta d(t_k)u}{\Delta d^T(t_k)\Delta d(t_k)} - \frac{\mathbf{1}^T G(t_k)\Delta d(t_k)\Delta d^T(t_k)u}{\Delta d^T(t_k)\Delta d(t_k)} = 0.$$

Therefore by induction the thesis is verified. ■

Lemma 3.3.3 guarantees that $\mathbf{1}^T \Delta q = 0$, or in other words it guarantees that the update step for q always returns a feasible point for the constrained optimization problem, if $q(0)$ is feasible.

Lemma 3.3.4. *The estimate $G(t_{k+1})$ has full rank as long as $G(0)$ is full rank and $d(t_j) \neq 0$ for $0 \leq t_j \leq t_k$.*

Proof. By Sherman-Morrison formula, $G(t_{k+1})$ has full rank whenever $G(t_k)$ is full rank and

$$\Delta d^T(t_k)G(t_k)^{-1}\Delta q(t_k) \neq 0.$$

But

$$\Delta d^T(t_k)G(t_k)^{-1}\Delta q(t_k) = -\Delta d^T(t_k)d(t_k) = d^T(t_k)G(t_k)Md(t_k)$$

which is zero if and only if $d(t_k) = 0$. Therefore by induction $G(t_{k+1})$ has full rank. ■

Lemma 3.3.5 (Lemma 2.1 in Gay 1979)**.** *Consider the k-th iteration of algorithm (3.40). If*

$$d(t_k) \text{ and } \Delta d(t_{k-1}) \text{ are linearly independent,}$$

then for $1 \leq j \leq \lfloor (k+1)/2 \rfloor$, the $j+1$ vectors

$$\left[\Omega M G(t_{k-2j+1})\right]^i d(t_{k-2j+1}), \quad 0 \leq i \leq j,$$

are linearly independent.

Proof. The proof is given in Gay (1979). ■

Via these lemmas, the following result on the global, finite-time convergence of (3.40) can be stated.

Theorem 3.3.6. *Consider the algorithm* (3.40) *initialized with* $G(0) = \alpha I$, $\alpha > 0$, *and any feasible state* $q(0)$. *The algorithm converges in at most* $2N$ *steps to the solution of* (3.29).

Proof. By Lemma 3.3.5, there exists k with $1 \leq k \leq 2N$ such that $d(t_k)$ and $\Delta d(t_{k-1})$ are linearly dependent. Trivially, if $d(t_k) = 0$, this solves the optimization problem (3.29), as the gradient is orthogonal to the constraint and Lemma 3.3.3 ensures that $q(t_k)$ is a feasible point. If instead $\Delta d(t_{k-1}) = 0$, then from the definition of d it follows $\Omega M \Delta q(t_{k-1}) = 0$, and therefore

$$Mq(t_k) = Mq(t_{k-1}) + \beta \mathbf{1} \text{ for some } \beta \in \mathbb{R}.$$

Being M invertible, this means that $\Delta q(t_{k-1}) = \beta M^{-1}\mathbf{1}$. By left multiplying both terms by $\mathbf{1}^T$ and using Lemma 3.3.3, one gets

$$\beta \mathbf{1}^T M^{-1}\mathbf{1} = \mathbf{1}^T \Delta q(t_{k-1}) = 0.$$

Therefore $\beta = 0$ and $\Delta q(t_{k-1}) = 0$. By Lemma 3.3.4 and by (3.40), this implies that $d(t_j) = 0$ for some $j \leq k-1$, and therefore the solution has been reached in at most $k \leq 2N$ steps. As a last case, suppose that $d(t_k)$ and $\Delta d(t_{k-1})$ are both non zero, but they are linearly dependent. Therefore there exists $\lambda \neq 0$ such that $d(t_k) = \lambda \Delta d(t_{k-1})$. From the algorithm equations and from the secant condition, it results

$$\Delta d(t_{k-1}) = \Omega M \Delta q(t_{k-1}) = \Omega M H(t_k) \Delta d(t_{k-1}).$$

The same is then true for $d(t_k)$, yielding $d(t_k) = \Omega M G(t_k) d(t_k)$. By rearranging the algorithm equations it is easy to see that $d(t_{k+1}) = d(t_k) + \Omega M \Delta q(t_k)$ and therefore

$$d(t_{k+1}) = d(t_k) - \Omega M G(t_k) d(t_k) = 0.$$

Even in this case, $d(t_{k+1}) = 0$ together with Lemma 3.3.3 guarantee that $q(t_{k+1})$ is the solution of (3.29). ∎

The proposed algorithm (3.40) does not take into account any constraint in the communication among agents. Indeed, consider the following decomposition of the generic algorithm (3.37), of which (3.40) is a specific case, into update laws for the generic agent i:

$$\begin{aligned} q_i^+ &= q_i - \Gamma_i^T g(q) \\ \Gamma_i^+ &= \Phi_i(q, \Gamma, g(q)), \end{aligned} \tag{3.41}$$

where Γ_i^T is the i-th row of Γ. It is not possible, in general, to implement (3.41) in a distributed manner, as both the update for q_i and for the vector Γ_i require quantities that are

not available at node i: q, $g(q)$, and Γ.

It is therefore necessary to *distribute* the algorithm, i.e. to design an implementation of it which is consistent with the given communication graph $\mathcal{G}_C = (\mathcal{C}, \mathcal{E}_C)$, which specifies from which agents node can gather information for the algorithm execution.

In the special case in which at every iteration

$$\Gamma_{ij} \neq 0 \quad \Rightarrow \quad (j,i) \in \mathcal{E}_C, \tag{3.42}$$

then the update law for q_i only requires information that can be gathered from the set of neighbors $\mathcal{N}(i)^C$. The issue of distributing the algorithm among the agents reduces then to a sparsity condition on Γ (and a similar condition can be stated for the update law for Γ_i).

However, note that the decision variables in the optimization problem in (3.29) are coupled in two ways: by a possibly non diagonal Hessian M, and by the *complicating constraint* in which all the variables appear. For this reason, it is convenient and effective to introduce some shared piece of information among all agents: let $x \in \mathbb{R}^p$ be an auxiliary *fusion* vector and suppose all the nodes agree on its value.

The local update laws (3.41) can then be replaced by

$$\begin{aligned} q_i^+ &= q_i - \gamma_i(q_j, g_j(q), j \in \mathcal{N}_i^C; \eta_i, x) \\ \eta_i^+ &= \phi_i(q_j, g_j(q), j \in \mathcal{N}_i^C; \eta_i, x). \end{aligned} \tag{3.43}$$

where η_i is a local parameter vector. The fusion vector x is a function of all the data available in the system:

$$x = f(q, g(q), \eta_i, i \in \mathcal{C}).$$

Algorithm (3.43) is now consistent with the communication graph, because the update laws for both q_i and η_i are function of local data (q_i, $g_i(q)$, η_i), of data that can be gathered from neighbors $j \in \mathcal{N}_i^C$, and of the fusion vector x.

Of course, the way in which x is computed and how agents agree on its value is a key point in the design of the algorithm. For example a finer time scale might exist: the fusion vector x can be obtained as the result of another distributed algorithm, which exploits the same communication graph. This faster algorithm is initialized locally on the basis of the data stored in the nodes and on the basis of measured steady state of the underlying system. As it runs at a much faster pace, by the end of the period of time T it is able to implement the function f of the data and to guarantee that all nodes agree on a common x.

Among the main algorithms that can be exploited to obtain the fusion vector x, the tool of *average consensus* (as described in Section 2.2) results to be quite useful when dealing with optimization problems with linear equality constraints. Consider indeed the expression

(3.39) for the update step in an approximate Newton method:

$$q^+ = q - Hg(q) + \frac{\mathbf{1}^T Hg(q)}{\mathbf{1}^T H\mathbf{1}} H\mathbf{1}.$$

Let $x = \frac{\mathbf{1}^T Hg(q)}{\mathbf{1}^T H\mathbf{1}}$ be the scalar quantity on which all nodes have to agree. It can be obtained via average consensus, once the communication graph $\mathcal{G}_C$ is strongly connected. Indeed, if every node is capable of initializing a vector

$$z^{(i)}(0) = \begin{bmatrix} H_i g(q) & H_i \mathbf{1} \end{bmatrix}^T, \tag{3.44}$$

where H_i is the i-th row of H, then by running an average consensus algorithm they eventually agree on

$$\bar{z} = \begin{bmatrix} \frac{1}{N}\mathbf{1}^T Hg(q) & \frac{1}{N}\mathbf{1}^T H\mathbf{1} \end{bmatrix}^T$$

from which every node can obtain x.

Therefore if the estimate H of M^{-1} satisfies the sparsity constraint

$$H_{ij} \neq 0 \quad \Rightarrow \quad (i,j) \in \mathcal{E}_C,$$

then the computation of the update step will require only communication between neighbors in $\mathcal{G}_C$.

This approach can be applied to the Broyden quasi-Newton method (3.40).

Define $g^{(i)}$ the part of the gradient corresponding to the set of the neighbors $\mathcal{N}_i^C = \{j_1, \dots j_{n_i}\}$ of i:

$$g^{(i)} \in \mathbb{R}^{n_i}, \quad g_\ell^{(i)} = g_{j_\ell} \text{ for } \ell = 1, \dots, n_i. \tag{3.45}$$

Following the sparse secant method proposed in (Dennis and Schnabel, 1983, Theorem 11.2.1) and including a projection step as proposed in (3.39), consider the algorithm:

$$\begin{aligned} q^+ &= q - Hg(q) + \frac{\mathbf{1}^T Hg(q)}{\mathbf{1}^T H\mathbf{1}} H\mathbf{1} \\ H^+ &= H + \mathcal{P}_{\mathcal{E}_C}\left[D^+ \left(\Delta q - H\Delta g(q)\right) \Delta g^T(q)\right] \end{aligned} \tag{3.46}$$

where $\mathcal{P}_{\mathcal{E}_C}$ is the projection operator on the sparsity constrain induced by $\mathcal{E}_C$:

$$\left[\mathcal{P}_{\mathcal{E}_C}(A)\right]_{ij} = \begin{cases} A_{ij}, & \text{if } (i,j) \in \mathcal{E}_C \\ 0, & \text{if } (i,j) \notin \mathcal{E}_C, \end{cases}$$

and where D^+ is the diagonal matrix defined as

$$\left[D^+\right]_{ii} = \begin{cases} 1/{g^{(i)}}^T(q(t_k))g^{(i)}(q(t_k)) & \text{if } g^{(i)}(q(t_k)) \neq 0 \\ 0, & \text{if } g^{(i)}(q(t_k)) = 0. \end{cases}$$

It is easy to see that (3.46) corresponds to (3.40) when the graph is complete, with the only difference that in the complete-graph case, thanks to Lemma 3.3.3, the projection on the constrain is performed on the measured gradient and not on the descent step.

Algorithm (3.46) can be distributed among the agents obtaining the following updates for q_i and $H_i \in \mathbb{R}^{n_i}$.

$$\begin{aligned} q_i^+ &= q_i - H_i^T g^{(i)}(q) + xH_i^T \mathbf{1}_{n_i} \\ H_i^+ &= H_i + \left[\Delta q_i - H_i^T \Delta g^{(i)}(q)\right] \frac{\Delta g^{(i)}(q)}{{\Delta g^{(i)}}^T(q)\Delta g^{(i)}(q)} \end{aligned}$$

where $\mathbf{1}_{n_i} \in \mathbb{R}^{n_i}$ is the vector of all ones, and $x = \bar{z}_1/\bar{z}_2$, where $\bar{z}_{1,2}$ are the elements of the 2-dimensional vector resulting from the average consensus algorithm initialized as

$$z^{(i)}(0) = \begin{bmatrix} H_i^T g^{(i)}(q) & H_i^T \mathbf{1}_{n_i} \end{bmatrix}^T.$$

Both the measure of the voltages v for the estimation of $g(q)$, and the initialization of z, take place as soon as the steady state response of the underlying system is available. Note that the update laws for q_i's and H_i's are consistent with the communication graph, and memory requirements for every node scale with the number of neighbors. The effectiveness of the quasi-Newton algorithm subject to sparsity constraints (including convergence properties) depends of course on the structure of the inverse of the Hessian, and therefore on the particular optimization problem that the algorithm has to solve. Simulations show that the problem of optimal reactive power compensation is a notable example in this sense.

Simulations

Consider a testbed of 33 nodes, 10 of which are compensators. Reactive power demands have been chosen according to a unitary variance normal distribution with negative mean $-\sigma$ (standard deviation). The initial injected reactive power of the compensators have been normally distributed too. The electrical connection tree has height 6, and every node that is not a leaf has an average of 2.4 children.

Assume that communications take place through the electrical lines, and that nodes that

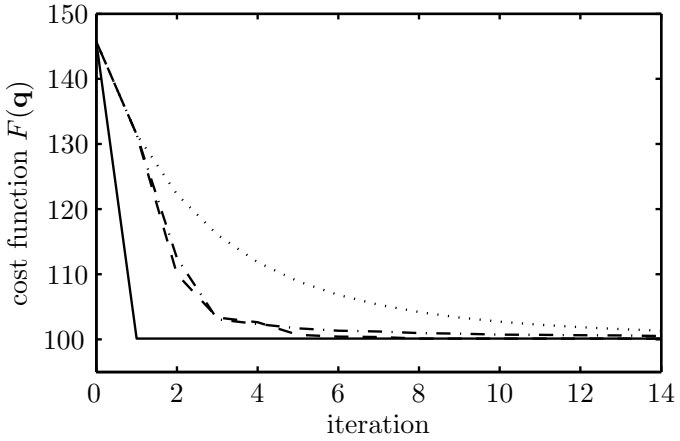


Figure 3.21: Comparison of different methods: Newton (solid), quasi-Newton with complete communication graph (dashed), quasi-Newton subject to the non-complete communication graph $\mathcal{G}_C$ (dash-dotted), fixed step steepest descent (dotted).

are connected by a path of length not greater than 4 are able to communicate. This is the simplest model for Power Line Communication, a promising communication solution for smart grids. The induced communication graph results to be connected, but not complete.

In Figure 3.21 the behavior of the different proposed algorithms has been plotted. For the sake of fairness in the comparison, the step length for the fixed-step-length, steepest descent algorithm, has been optimized for fastest convergence, and the same initial condition has been set for quasi-Newton methods. Note however that this optimal choice for the step length requires some global knowledge of the problem: if this knowledge is not available or if it is approximate, more conservative choices would be preferred (corresponding to a smaller step length) and therefore the advantage of quasi-Newton methods would be even larger.

In the top left quadrant of Figure 3.22, the inverse of the Hessian has been plotted. One can see how the sparsity constraint induced by the communication graph $\mathcal{G}_C$ (and plotted in the lower right quadrant) is meaningful in describing the largest elements (in absolute value) of the inverse of Hessian. The other two quadrant of Figure 3.22 show the estimates of the Hessian returned by the quasi-Newton methods when the algorithm has converged.

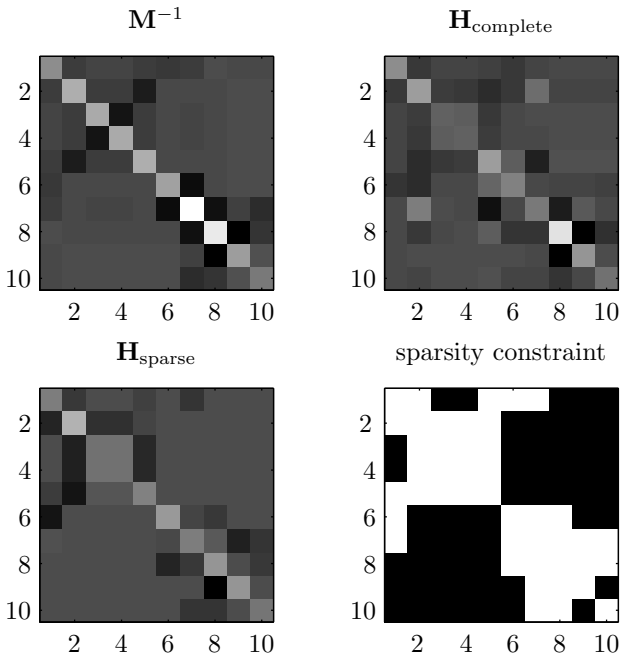


Figure 3.22: Estimates of the inverse of the Hessian returned by the proposed algorithms, together with the sparsity constraint induced by the communication graph.

A gossip-like algorithm for optimal reactive power compensation

Another way to tackle the optimal reactive power flow problem has been proposed in Bolognani and Zampieri (2011), inspired somehow by the randomized gossip algorithms widely applied in the consensus problems.

Let the set of compensators $\mathcal{C} = \{1, \dots, N\}$ be divided into ℓ possibly overlapping subsets $\mathcal{C}_1, \dots, \mathcal{C}_\ell$, with $\bigcup_{i=1}^{\ell} \mathcal{C}_i = \mathcal{C}$. This family of subsets can be interpreted as a hyper-graph defined over the node set $\mathcal{C}$.

Nodes belonging to the same set are able to communicate each other (they form a clique in the communication graph $\mathcal{G}_C$), and they are therefore capable of coordinating and sharing measurements and local knowledge of the problem parameters M and m. Assume that, by using this information, nodes belonging to the same set are capable of driving their state in a new feasible state that minimizes $F(q)$ as defined in (3.29), solving the optimization *subproblem* in which all nodes that are not in $\mathcal{C}_i$ keep their state constant:

$$\begin{aligned} \arg\min_{\Delta q} \quad & F(q + \Delta q) \\ \text{subject to} \quad & \Delta q \in \mathcal{S}_i, \end{aligned}$$

where

$$\mathcal{S}_i := \left\{ q \in \mathbb{R}^N \ : \ \sum_{j \in \mathcal{C}_i} q_j = 0 \ , q_j = 0 \ \forall j \notin \mathcal{C}_i \right\}.$$

One possible way in which nodes in $\mathcal{C}_i$ can solve this optimization subproblem is the following.

Let $\mathcal{A}, \mathcal{B} \subseteq \{1, \dots, N\}$ be two nonempty set of indices. Define $M_{\mathcal{A}\mathcal{B}}$ as the submatrix of M obtained by selecting the rows indexed by $\mathcal{A}$ and the columns indexed by $\mathcal{B}$. Let the same definition hold for a vector, i.e. $v_{\mathcal{A}}$ is the subvector of v obtained by selecting the elements indexed by $\mathcal{A}$. Let moreover $\bar{\mathcal{C}}_i$ be the set complement $\mathcal{C} \backslash \mathcal{C}_i$.

The optimization problem faced by the nodes in $\mathcal{C}_i$ can then be rewritten as

$$\begin{aligned} \min_{q_{\mathcal{C}_i}} \quad & q_{\mathcal{C}_i}^T \frac{M_{\mathcal{C}_i \mathcal{C}_i}}{2} q_{\mathcal{C}_i} + \left(q_{\bar{\mathcal{C}}_i} M_{\bar{\mathcal{C}}_i \mathcal{C}_i} + m_{\mathcal{C}_i}^T \right) q_{\mathcal{C}_i} \\ \text{subject to} \quad & \mathbf{1}^T q_{\mathcal{C}_i} = c - \mathbf{1}^T q_{\bar{\mathcal{C}}_i}. \end{aligned} \tag{3.47}$$

It is easy to see that agents in $\mathcal{C}_i$ can reach the optimal solution by adding to $q_{\mathcal{C}_i}$ the increment $\Delta q_{\mathcal{C}_i}$ which is equal to

$$\Delta q_{\mathcal{C}_i} = -M_{\mathcal{C}_i \mathcal{C}_i}^{-1} \nabla F_{\mathcal{C}_i} + \frac{\mathbf{1}^T M_{\mathcal{C}_i \mathcal{C}_i}^{-1} \nabla F_{\mathcal{C}_i}}{\mathbf{1}^T M_{\mathcal{C}_i \mathcal{C}_i}^{-1} \mathbf{1}} M_{\mathcal{C}_i \mathcal{C}_i}^{-1} \mathbf{1},$$

where $M_{\mathcal{C}_i \mathcal{C}_i}^{-1}$ is the inverse of submatrix $M_{\mathcal{C}_i \mathcal{C}_i}$ and

$$\nabla F_{\mathcal{C}_i} = M_{\mathcal{C}_i \mathcal{C}_i} q_{\mathcal{C}_i} + M_{\mathcal{C}_i \bar{\mathcal{C}}_i} q_{\bar{\mathcal{C}}_i} + m_{\mathcal{C}_i}$$

is the subvector of the gradient of $F(q)$ corresponding to the agents belonging to $\mathcal{C}_i$.

It has been shown earlier in this section that an estimate of the gradient $\nabla F(q) = Mq + m$ can be obtained by sensing the network when in the state q. More precisely, it has been shown that, under a certain assumption on the impedance of the lines, the steady state voltage angle measurement $\theta^u \in \mathbb{R}^N$ approximates $\nabla F(q)$ up to a scaling and a common additive term, namely

$$\theta^u \approx \frac{1}{2U_0^2} \nabla F(q) + \alpha \mathbf{1},$$

with α unknown.

Nodes in $\mathcal{C}_i$ can therefore solve their corresponding optimization subproblem by perform-

ing the update

$$\Delta q_{\mathcal{C}_i} = -M^{-1}_{\mathcal{C}_i\mathcal{C}_i} 2\theta^u_{\mathcal{C}_i} + \frac{\mathbf{1}^T M^{-1}_{\mathcal{C}_i\mathcal{C}_i} 2\theta^u_{\mathcal{C}_i}}{\mathbf{1}^T M^{-1}_{\mathcal{C}_i\mathcal{C}_i}\mathbf{1}} M^{-1}_{\mathcal{C}_i\mathcal{C}_i}\mathbf{1},$$

as the uncertain term $\alpha\mathbf{1}$ get canceled in the expression.

The data that any node k in $\mathcal{C}_i$ has to know are then its corresponding row in $M^{-1}_{\mathcal{C}_i\mathcal{C}_i}$, the angles $\theta^u_{\mathcal{C}_i}$ from the other nodes in $\mathcal{C}_i$, and its own state q_k.

Notice that the update law requires only local information or information that can be gathered inside the subset $\mathcal{C}_i$ (which is a clique in the communication graph). This is possible because the gradient, that otherwise would depend on the whole system state, can be estimated from some local measurements. Moreover, the elements of $M_{\mathcal{C}_i\mathcal{C}_i}$ depends only on the length of the electric paths that connect nodes in $\mathcal{C}_i$, and therefore it can be assumed that that this information is available and shared among the nodes of the same cluster.

The proposed optimization algorithm will therefore consists of the following, repeated steps:

i) a set $\mathcal{C}_i$ is chosen according to a sequence of symbols $\sigma(t) \in \{1, \dots, \ell\}$;

ii) agents in $\mathcal{C}_i$ sense the network and obtain, directly or via some filtering, an estimate of the gradient;

iii) they determine a feasible update step that minimizes the given cost function, possibly by coordinating their actions and communicating;

iv) they actuate the system by updating their state (the injected reactive power).

The iterated algorithm will then results in the following discrete time system for q

$$\begin{aligned} q(t+1) = T_{\sigma(t)}[q(t)] \quad := \quad & \arg\min_{q} \quad F(q) \\ & \text{subject to} \quad q - q(t) \in \mathcal{S}_{\sigma(t)}, \end{aligned} \tag{3.48}$$

with initial conditions $q(0)$ such that $\mathbf{1}^T q(0) = c$.

Observe that $\mathcal{S}_i = \operatorname{Im}\Omega_i$, $\Omega_i \in \mathbb{R}^{N\times N}$ being defined as

$$\Omega_i = I_{\mathcal{C}_i} - \frac{1}{|\mathcal{C}_i|}\mathbf{1}_{\mathcal{C}_i}\mathbf{1}^T_{\mathcal{C}_i},$$

where $|\mathcal{C}_i|$ is the cardinality of the set $\mathcal{C}_i$, $I_{\mathcal{C}_i}$ is the diagonal matrix having diagonal entries 1 in positions belonging to $\mathcal{C}_i$ and zero elsewhere and $\mathbf{1}_{\mathcal{C}_i}$ is the column vector having entries 1

in positions belonging to $\mathcal{C}_i$ and zero elsewhere. Notice that

$$\Omega_i = \frac{1}{2|\mathcal{C}_i|} \sum_{h,k \in \mathcal{C}_i} (e_h - e_k)(e_h - e_k)^T,$$

where e_i is the column vector having entry 1 in position i and zero elsewhere.

Convergence results

By introducing the auxiliary variable $x = q - q^*$, where q^* is given in (3.31), it can be shown that the optimization problem (3.29) is equivalent to

$$\begin{aligned} \min_x \quad & V(x) = x^T \frac{M}{2} x \\ \text{subject to} \quad & \mathbf{1}^T x = 0, \end{aligned} \tag{3.49}$$

and that the subproblems described in the previous section are equivalent to the subproblems

$$\begin{aligned} \min_{\Delta x} \quad & V(x + \Delta x) \\ \text{subject to} \quad & \mathbf{1}^T \Delta x = 0, \\ & \Delta x \in \operatorname{Im} \Omega_i. \end{aligned} \tag{3.50}$$

In this notation, it is possible to explicitly express the solution of the individual subproblems as a linear function of the starting point $x(t)$:

$$x(t+1) = F_i x(t), \quad F_i = I - (\Omega_i M \Omega_i)^\sharp M, \tag{3.51}$$

where $\sharp$ means pseudoinverse.

The discrete time system (3.48) in the x coordinates results then to be a linear time varying system of the form

$$x(t+1) = F_{\sigma(t)} x(t). \tag{3.52}$$

It is easy to verify from the properties of the pseudoinverse that $\ker(\Omega_i M \Omega_i)^\sharp = \ker \Omega_i$ and $\operatorname{Im}(\Omega_i M \Omega_i)^\sharp = \operatorname{Im} \Omega_i$. The matrices F_i are projection operators, i.e. $F_i^2 = F_i$, and they are orthogonal projections with respect to the inner product $\langle \cdot, \cdot \rangle_M$, defined as $\langle x, y \rangle_M := x^T M y$. In other words, $\langle F_i x, F_i x - x \rangle_M = x^T M (F_i x - x) = 0$. Moreover, they are self-adjoint matrices with respect to the inner product $\langle \cdot, \cdot \rangle_M$, i.e. $F_i^T M = M F_i$.

The following results characterize the uniqueness of the equilibrium for all maps $T_i[x] = F_i x$.

Lemma 3.3.7. *Consider the family of linear transformations* $\{F_i\}$ *as described in* (3.51). $\bar{x} = 0$ *is the only point in* $\ker \mathbf{1}^T$ *that is invariant for all* F_i*'s if and only if*

$$\mathrm{Im}[\Omega_1 \dots \Omega_\ell] = \ker \mathbf{1}^T.$$

Proof. Let us prove the reverse implication first. If $\mathrm{Im}[\Omega_1 \dots \Omega_\ell] = \ker \mathbf{1}^T$, then $\bar{x}$ can be expressed as

$$\bar{x} = \sum_i \Omega_i y_i.$$

Moreover, as $F_i \bar{x} = \bar{x}$ for all i, then $M\bar{x} \in \ker \Omega_i$. Therefore

$$\bar{x}^T M \bar{x} = \sum_i y_i^T \Omega_i M \bar{x} = 0,$$

and so, since M is positive definite, $\bar{x} = 0$.

Suppose conversely that

$$\ker \begin{bmatrix} \mathbf{1}^T \\ I - F_1^T \\ \vdots \\ I - F_\ell^T \end{bmatrix} = 0$$

and take any $\bar{x} \in \ker \mathbf{1}^T$. Then there exist a scalar α and vectors y_i such that

$$\begin{aligned} M\bar{x} &= \alpha \mathbf{1} + \sum_i (I - F_i)^T y_i \\ &= \alpha \mathbf{1} + \sum_i^T M(\Omega_i M \Omega_i)^\sharp y_i \end{aligned}$$

Then

$$\bar{x} = M^{-1}\alpha \mathbf{1} + \sum_i^T (\Omega_i M \Omega_i)^\sharp y_i$$

Now observe that $\mathbf{1}^T \bar{x} = \alpha \mathbf{1}^T M^{-1} \mathbf{1}$, which, since M^{-1} is positive definite, implies that $\alpha = 0$ and so $\bar{x} \in \mathrm{Im}[\Omega_1 \dots \Omega_\ell]$. The converse inclusion is trivial. ∎

The condition $\mathrm{Im}[\Omega_1 \dots \Omega_\ell] = \ker \mathbf{1}^T$ is then a necessary condition for the convergence of the algorithm. Notice that this condition is equivalent to the fact that $L + \mathbf{1}\mathbf{1}^T$ is positive

definite, where

$$\begin{aligned} L :=& [\Omega_1 \dots \Omega_\ell] \operatorname{diag}\{2|\mathcal{C}_1|I, \dots, 2|\mathcal{C}_\ell|I\} [\Omega_1 \dots \Omega_\ell]^T \\ =& \sum_i 2|\mathcal{C}_i|\Omega_i = \sum_{i=1}^{\ell} \sum_{h,k \in \mathcal{C}_i} (e_h - e_k)(e_h - e_k)^T \\ =& \sum_{h,k=1}^{N} (e_h - e_k)(e_h - e_k)^T \sum_{i=1}^{\ell} \delta_{\mathcal{C}_i}(h) \delta_{\mathcal{C}_i}(k), \end{aligned}$$

and where the symbol $\delta_{\mathcal{C}_i}(\cdot)$ means the characteristic function of the set $\mathcal{C}_i$, namely a function of the nodes that is 1 when the node belongs to $\mathcal{C}_i$ and is zero otherwise. The matrix L can be interpreted an the Laplacian matrix of a weighted graph $\mathcal{G}$ having nodes $\{1, \dots, N\}$ and weights on the edge h, k equal to the number of the sets $\mathcal{C}_i$ which contains both h and k. As before, the family of sets $\{\mathcal{C}_1, \dots, \mathcal{C}_\ell\}$ can be interpreted as an hyper-graph $\mathcal{H}$. It is quite easy to see that the hyper-graph $\mathcal{H}$ with edges $\mathcal{C}_i$ is connected if and only if $\mathcal{G}$ is a connected graph. From these arguments, the next result follows.

Proposition 3.3.8. *The condition* $\operatorname{Im}[\Omega_1 \dots \Omega_\ell] = \ker \mathbf{1}^T$ *holds if and only if* $\mathcal{H}$ *is a connected hyper-graph.*

To characterize now the convergence of the algorithm, the following assumption on the sequence $\sigma(t)$ is needed.

Assumption 3.3.9. The sequence $\sigma(t)$ is a sequence of independently, uniformly distributed symbols in $\{1, \dots, \ell\}$.

The main tools that are needed for the convergence analysis are the formalism of *set-valued* maps and the Invariance Principle for these maps.

A set-valued map $T : X \rightrightarrows X$ associates to an element of X a subset of X. T is non-empty if $T(x) \neq \emptyset$ for all $x \in X$. An evolution of the dynamical system determined by a non-empty set-valued map T is a sequence $\{x_t\}_{t \in \mathbb{Z}_{\geq 0}}$ with the property that $x_{t+1} \in T(x_t)$ for all $t \in \mathbb{Z}_{\geq 0}$. A set W is strongly positively invariant for T if $T(w) \subset W$ for all $w \in W$. The following theorem holds.

Theorem 3.3.10 (Th. 4.5 in Bullo, Carli, and Frasca Submitted)**.** *Let* (X, d) *be a metric space. Given a collection of maps* $T_1, \dots, T_\ell$*, define the set-valued map* $T : X \rightrightarrows X$ *by* $T(x) = \{T_1(x), \dots, T_\ell(x)\}$*. Given a stochastic process* $\sigma : \mathbb{Z}_{\geq 0} \to \{1, \dots, \ell\}$*, consider an evolution* $\{x_n\}_{n \in \mathbb{Z}_{\geq 0}}$ *of* T *satisfying*

$$x_{n+1} = T_{\sigma(n)}(x_n).$$

Assume that

i) there exists a compact set $W \subseteq X$ that is strongly positively invariant for T;

ii) there exists a function $U : W \to \mathbb{R}$ such that $U(w') < U(w)$, for all $w \in W$ and $w' \in T(w) \backslash \{w\}$;

iii) the maps T_i, for $i \in \{1, \dots, \ell\}$, and U are continuous on W; and

iv) there exists $p \in]0,1[$ and $h \in \mathbb{N}$ such that, for all $i \in \{1, \dots, \ell\}$ and $n \in \mathbb{Z}_{\geq 0}$

$$\mathbb{P}[\sigma(n+h) = i | \sigma(n), \dots, \sigma(1)] \geq p.$$

If $x_0 \in W$, then there exists $c \in \mathbb{R}$ such that almost surely the evolution $\{x_n\}_{n \in \mathbb{Z}_{\geq 0}}$ approaches the set

$$(J_1 \cap \dots \cap J_\ell) \cap U^{-1}(c),$$

where $J_i = \{w \in W | T_i(w) = w\}$ is the set of fixed points of T_i in W, $i \in \{1, \dots, \ell\}$.

This yields the following result.

Theorem 3.3.11. *Consider the discrete time system* (3.52)*, and let Assumption 3.3.9 hold. If* $\mathrm{Im}[\Omega_1 \dots \Omega_\ell] = \ker \mathbf{1}^T$*, then*

$$x(t) \to 0 \quad as \quad t \to \infty \quad almost\ surely$$

for all $x(0) \in \mathbb{R}^N$.

Proof. Consider the linear maps $F_i(x) = F_i x$ and the corresponding set-valued map $T(x) = \{F_1(x), \dots, F_\ell(x)\}$. Let W be the compact set $V^{-1}(x(0))$. V is strongly positively invariant for T as $V(F_i x) \leq V(x)$ for all x, i (as $F_i x$ solves the optimization subproblems initialized in x). As F_i's are orthogonal projection matrices for the norm $\langle \cdot, \cdot \rangle_M$, $V(F_i x) = V(x)$ implies $F_i x = x$ (as $V(x) = \|x\|_M^2/2$), and then $F_i x \neq x$ implies $V(F_i x) < V(x)$. Moreover, because of Assumption 3.3.9, for all n, i

$$\mathbb{P}[\sigma(n+1) = i | \sigma(n), \dots, \sigma(1)] = \mathbb{P}[\sigma(n+1) = i] = \frac{1}{\ell} > 0.$$

Theorem 3.3.10 then applies. Because of Lemma 3.3.7, the intersection of the fixed points of the maps F_i reduces to $x = 0$, and therefore $x(t) \to 0$ almost surely as $t \to \infty$. ∎

Rate of convergence

Consider the performance metric

$$R = \sup_{x(0)\in\ker \mathbf{1}^T} \limsup v(t)^{1/t}$$

where $v(t) = \mathbb{E}\left[V(x(t))\right]$. R describes the exponential rate of convergence to zero of $v(t)$ and so also the exponential rate of convergence of $q(t)$ to the optimal solution q^*. Then, by using (3.51),

$$\begin{aligned} v(t) &= \frac{1}{2}\mathbb{E}\left[x(t)^T M x(t)\right] \\ &= \frac{1}{2}\mathbb{E}\left[x(t)^T \Omega M \Omega x(t)\right] \\ &= \frac{1}{2}\mathbb{E}\left[x(t-1)^T F_{\sigma(t)}^T \Omega M \Omega F_{\sigma(t)} x(t-1)\right] \\ &= \frac{1}{2}x(0)^T \mathbb{E}\left[F_{\sigma(1)}^T \cdots F_{\sigma(t)}^T \Omega M \Omega F_{\sigma(t)} \cdots F_{\sigma(1)}\right] x(0). \end{aligned}$$

Let us then define

$$\Delta(t) = \mathbb{E}\left[F_{\sigma(1)}^T \cdots F_{\sigma(t)}^T \Omega M \Omega F_{\sigma(t)} \cdots F_{\sigma(1)}\right].$$

Via Assumption 3.3.9, the following linear system can be derived

$$\begin{aligned} \Delta(t+1) &= \mathbb{E}\left[F^T \Delta F\right] = \mathcal{L}(\Delta(t)), \quad \Delta(0) = \Omega M \Omega \\ \Xi(t) &= \Omega \Delta(t) \Omega, \end{aligned} \tag{3.53}$$

while the expected cost function can be expressed as

$$\mathbb{E}\left[V(x(t))\right] = v(t) = \frac{1}{2}x(0)^T \Xi(t) x(0).$$

Let denote by $\mathbf{F}$ the $N^2 \times N^2$ matrix associated with the linear transformation $\mathcal{L}$:

$$\mathrm{vec}\left(\Delta(t+1)\right) = \mathbf{F}\,\mathrm{vec}\left(\Delta(t)\right),$$

where vec$(\cdot)$ is the operation of vectorization. Then

$$\mathbf{F} = \mathbb{E}\left[F^T \otimes F^T\right],$$

which is self-adjoint with respect to the inner product $\langle \cdot, \cdot \rangle_{M^{-1}\otimes M^{-1}}$. Therefore $\mathbf{F}$ has real

eigenvalues. Consider the function

$$\lambda_{\mathcal{L}}(i) : \{1, \dots, N^2\} \to \mathbb{R}$$

that returns the i-th eigenvalue of $\mathbf{F}$. Assume that the function is monotonically non increasing, i.e. $\lambda_{\mathcal{L}}(i) \geq \lambda_{\mathcal{L}}(i+1)$ for all i. This map can be also represented as an N^2-dimensional ordered vector (in decreasing order, with possible repetitions) $\lambda_{\mathcal{L}} = [\lambda_{\mathcal{L}}(1) \cdots \lambda_{\mathcal{L}}(N^2)]^T$. Let moreover $\Delta_{\mathcal{L}}(i)$ be an eigenvector associated with the eigenvalue $\lambda_{\mathcal{L}}(i)$.

By decomposing $\Omega M \Omega$ into $\sum_i \alpha_i \Delta_{\mathcal{L}}(i)$, the convergence rate R can be rewritten as

$$R = \max \left\{ |\lambda_{\mathcal{L}}(i)| \mid \alpha_i \neq 0,\ \Omega \Delta_{\mathcal{L}}(i) \Omega \neq 0 \right\}. \tag{3.54}$$

The following proposition relates the convergence result of Theorem 3.3.11 with the approach of this section, showing how the same conditions for convergence also guarantee asymptotic stability of the dynamics of (3.53).

Proposition 3.3.12. *Let* $\operatorname{Im} \left[\Omega_1 \cdots \Omega_\ell \right] = \ker \mathbf{1}^T$. *Then* $R < 1$.

Proof. Let us define $\mathcal{L}_i$ as the linear transformation $\mathcal{L}_i(\Delta) = F_i^T \Delta F_i$. The N^2 eigenvalues of $\mathcal{L}_i$ are $\lambda(F_i^T \otimes F_i^T)$, and therefore belong to the set $\{0, 1\}$. As $F_i^T \otimes F_i^T$ is self-adjoint, it follows that $\|F_i^T \otimes F_i^T\|_2 \leq 1$. By using the fact that $\mathcal{L}(\Delta)$ is a convex combination of the elements of $\{\mathcal{L}_i(\Delta)\}$, it follows

$$\max\{|\lambda_{\mathcal{L}}(i)|\} \leq \|\mathbb{E}\left[F^T \otimes F^T \right]\| \leq 1.$$

Let us then consider $\lambda_{\mathcal{L}}(i)$ such that $|\lambda_{\mathcal{L}}(i)| = 1$, and let $x = \operatorname{vec}(\Delta_{\mathcal{L}}(i))$ be the corresponding eigenvector of $\mathbf{F}$. Then

$$\|x\| = \left\| \mathbb{E}\left[F^T \otimes F^T \right] x \right\| \leq \mathbb{E}\left[\left\| F^T \otimes F^T x \right\| \right] \leq \|x\|,$$

and therefore

$$\left\| F_i^T \otimes F_i^T x \right\| = \|x\| \quad \forall i.$$

$F_i^T \otimes F_i^T$ has only 0 and 1 eigenvalues, and eigenvectors $v_h^{(i)} \otimes v_k^{(i)}$, where $v_{h,k}^{(i)}$ are right eigenvectors of F_i. Therefore it must be

$$(F_i^T \otimes F_i^T) x = x \quad \forall i$$

and then

$$x = v_h \otimes v_k, \quad \Omega_i^T v_h = \Omega_i^T v_k = 0 \quad \forall i.$$

As

$$\text{Im}\left[\Omega_1 \cdots \Omega_\ell\right] = \ker \mathbf{1}^T \quad \Rightarrow \quad \bigcap_i \ker \Omega_i^T = \text{Im}\, \mathbf{1},$$

it must be $v_h = v_k = \mathbf{1}$, and therefore the only eigenvector of $\mathcal{L}$ corresponding to an eigenvalue of norm 1 is $\Delta_{\mathcal{L}}(1) = \mathbf{1}\mathbf{1}^T$. As it is not observable ($\Omega \mathbf{1}\mathbf{1}^T \Omega = 0$), then $R < 1$. ■

Computing R as defined in (3.54) is in general not simple. In the following, it is presented an upper bound for R that can be computed from $\bar{F} = \mathbb{E}[F]$. A few technical lemmas are needed.

Lemma 3.3.13. *Let $P, Q \in \mathbb{R}^{N\times N}$ and $P \geq Q$. Then $\mathcal{L}^k(P) \geq \mathcal{L}^k(Q)$ for all $k \in \mathbb{Z}_{\geq 0}$.*

Proof. From the definition of $\mathcal{L}$, it results

$$\begin{aligned} x^T [\mathcal{L}(P) - \mathcal{L}(Q)] x &= x^T \left[\mathbb{E}\left[F^T P F\right] - \mathbb{E}\left[F^T Q F\right]\right] x \\ &= \mathbb{E}\left[x^T F^T (P - Q) F x\right] \\ &\geq 0. \end{aligned}$$

By iterating these steps k times, $\mathcal{L}^k(M) \geq \mathcal{L}^k(N)$ follows. ■

Lemma 3.3.14. *The following holds for all Δ:*

$$\Omega \mathcal{L}^t(\Omega \Delta \Omega)\Omega = \Omega \mathcal{L}^t(\Delta)\Omega.$$

Proof. Proof is by induction. The statement is true for $t = 0$, as $\Omega^2 = \Omega$. Suppose it is true up to t. Then

$$\begin{aligned} \Omega \mathcal{L}^{t+1}(\Delta)\Omega &= \Omega \mathcal{L}(\mathcal{L}^t(\Delta))\Omega \\ &= \Omega \mathcal{L}(\Omega \mathcal{L}^t(\Delta)\Omega)\Omega \\ &= \Omega \mathcal{L}(\Omega \mathcal{L}^t(\Omega \Delta \Omega)\Omega)\Omega \\ &= \Omega \mathcal{L}^{t+1}(\Omega \Delta \Omega)\Omega. \end{aligned}$$

■

Lemma 3.3.15. *Let $\bar{F} = \mathbb{E}[F]$. If $\text{Im}\left[\Omega_1 \cdots \Omega_\ell\right] = \ker \mathbf{1}^T$, then all the eigenvalues of $\bar{F}$ have absolute value not larger than 1, and its only eigenvalue on the unitary circle is $\lambda = 1$, with associated left eigenvector $\mathbf{1}$ and right eigenvector $M^{-1}\mathbf{1}$.*

Proof. The fact that all eigenvalues lie inside or on the unit circle follows from the fact that $\bar{F}$ is the convex combination of matrices F_i that satisfies $\|F_i\|_M \leq 1$ for all i's. Consider then

an eigenvector x such that $\|x\| = \|\bar{F}x\|$. Then

$$\|\bar{F}x\| \leq \mathbb{E}\left[\|F_i x\|\right] \leq \|x\|,$$

and therefore $\|F_i x\| = \|x\|$ for all i's. As F_i are projection matrices, it means that $F_i x = x$ and then $Mx \in \ker \Omega_i^T, \forall i$. Similarly to what has been done in the proof of Proposition 3.3.12, using the fact that $\text{Im}\left[\Omega_1 \cdots \Omega_\ell\right] = \ker \mathbf{1}^T$, it must be $x = M^{-1}\mathbf{1}$. By inspection it can be verified that the left eigenvector corresponding to the same eigenvalue is $\mathbf{1}$. ■

The following result can now be presented.

Theorem 3.3.16. *Consider the linear system* (3.53) *and the rate of convergence R defined in* (3.54). *Define*

$$\beta = \max\{|\lambda| \mid \lambda \in \lambda(\bar{F}), \lambda \neq 1\}$$

where $\bar{F} = \mathbb{E}[F]$. Then $R \leq \beta$.

Proof. Let us first prove that $\Omega\mathcal{L}(\Omega M\Omega)\Omega \leq \beta\Omega M\Omega$. Indeed, using the fact that $\Omega F\Omega = F\Omega$ and that $F_i^T M F_i = M F_i$,

$$\begin{aligned} x^T\Omega\mathcal{L}(\Omega M\Omega)\Omega x &= \mathbb{E}\left[x^T\Omega F^T\Omega M\Omega F\Omega x\right] \\ &= \mathbb{E}\left[x^T\Omega F^T M F\Omega x\right] \\ &= x^T\Omega M^{1/2}\mathbb{E}\left[M^{1/2}FM^{-1/2}\right]M^{1/2}\Omega x. \end{aligned}$$

$\mathbb{E}\left[M^{1/2}FM^{-1/2}\right] = M^{1/2}\bar{F}M^{-1/2}$ is symmetric and, by Lemma 3.3.15, it has only one eigenvalue on the unit circle (precisely in 1), with eigenvector $M^{-1/2}\mathbf{1}$. As $M^{1/2}\Omega x \perp M^{-1/2}\mathbf{1}$ for all x, it must be

$$x^T\Omega\mathcal{L}(\Omega M\Omega)\Omega x \leq \beta\Omega M\Omega,$$

with $\beta = \max\{|\lambda| \mid \lambda \in \lambda(\bar{F}), \lambda \neq 1\}$.

From this result, using Lemmas 3.3.13 and 3.3.14, one can conclude

$$\begin{aligned} \Omega\mathcal{L}^t(\Omega M\Omega)\Omega &= \Omega\mathcal{L}^{t-1}(\mathcal{L}(\Omega M\Omega))\Omega \\ &= \Omega\mathcal{L}^{t-1}(\Omega\mathcal{L}(\Omega M\Omega)\Omega)\Omega \\ &\leq \Omega\mathcal{L}^{t-1}(\beta\Omega M\Omega)\Omega \\ &= \beta\Omega\mathcal{L}^{t-1}(\Omega M\Omega)\Omega \\ &\leq \cdots \leq \beta^t\Omega M\Omega, \end{aligned}$$

and therefore $R \leq \beta$. ■

The following result allows to compute R when the spectra of $\mathcal{L}$ and $\bar{F}$ are available (analytically or numerically).

Let define $\mathcal{O}$ as the non observable space for the system (3.53):

$$\mathcal{O} = \left\{ \Delta \in \mathbb{R}^{N\times N} \mid \Omega\Delta\Omega = 0 \right\},$$

and let $R_{\mathcal{O}}$ be the rate

$$R_{\mathcal{O}} = \max\left\{ |\lambda_{\mathcal{L}}(i) \mid \Delta_{\mathcal{L}}(i) \notin \mathcal{O} \right\}. \tag{3.55}$$

The following proposition holds.

Proposition 3.3.17. *Let R and $R_{\mathcal{O}}$ be defined by (3.54) and (3.55) respectively. Then*

$$R = R_{\mathcal{O}}.$$

Proof. For any eigenvector $\Delta_{\mathcal{L}}(i)$, there exists $\gamma > 0$ such that $\Delta_{\mathcal{L}}(i) \leq \gamma M$. Therefore $\Omega\mathcal{L}^t(\Delta_{\mathcal{L}}(i))\Omega \leq \gamma\Omega\mathcal{L}^t(M)\Omega$, and thus $\lambda_{\mathcal{L}}(i)^t\Omega\Delta_{\mathcal{L}}(i)\Omega \leq \gamma\Omega\mathcal{L}^t(M)\Omega$. If $\Delta_{\mathcal{L}}(i) \notin \mathcal{O}$, it must be $\lambda_{\mathcal{L}}(i) \leq R$, therefore $R_{\mathcal{O}} \leq R$. As of course $R_{\mathcal{O}} \geq R$, one concludes that $R = R_{\mathcal{O}}$. ■

The following result, illustrated also in Figure 3.23, can now be stated (remembering that $\lambda_{\mathcal{L}} \in \mathbb{R}^{N^2}$ and $\lambda_{\bar{F}} \in \mathbb{R}^N$ are the ordered vector of *possibly repeated* eigenvalues of $\mathcal{L}$ and $\bar{F}$).

Theorem 3.3.18. *The elements of the vector*

$$\lambda'_{\bar{F}} = [\lambda_{\bar{F}}(2), \ldots, \lambda_{\bar{F}}(N)]$$

appear twice in the vector

$$\lambda'_{\mathcal{L}} = [\lambda_{\mathcal{L}}(2), \ldots, \lambda_{\mathcal{L}}(N^2)],$$

and so R is the largest element in absolute value of the remaining ones in $\lambda'_{\mathcal{L}}$.

Proof. Via Lemma 3.3.14 it is possible to show that $\mathcal{O}$ is an invariant set:

$$\Omega\mathcal{L}(\Delta)\Omega = \Omega\mathcal{L}(\Omega\Delta\Omega)\Omega = 0 \quad \forall \Delta \in \mathcal{O}.$$

As the dimension of $\mathcal{O}$ is $2N-1$ (the dimension of the kernel of $\Omega \otimes \Omega$), there must exist $2N-1$ eigenvectors of $\mathcal{L}$ in $\mathcal{O}$. These eigenvectors can be constructed from the eigenvectors of $\bar{F}^T$. Indeed, consider N linearly independent vectors $v_1, \ldots, v_N$ such that $\bar{F}^T v_i = \mu_i v_i$ with

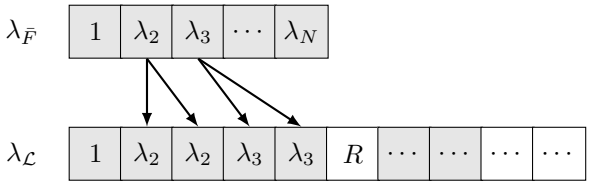


Figure 3.23: Representation of the eigenvalues of $\mathcal{L}$ and of $\bar{F}$, according to Theorem 3.3.18.

$1 = \mu_1 \geq \cdots \geq \mu_N$. Then for all i

$$\mathcal{L}(\mathbf{1}v_i^T) = \mathbb{E}\left[F^T\mathbf{1}v_i^T F\right] = \mathbf{1}v_i^T\mathbb{E}\left[F\right] = \mu_i\mathbf{1}v_i^T$$
$$\mathcal{L}(v_i\mathbf{1}^T) = \mathbb{E}\left[F^T v_i\mathbf{1}^T F\right] = \mathbb{E}\left[F\right]v_i\mathbf{1}^T = \mu_i v_i\mathbf{1}^T.$$

For all these eigenvectors it holds $\Omega v_i\mathbf{1}^T\Omega = \Omega\mathbf{1}v_i^T\Omega = 0$. They are therefore a basis of $2N-1$ linearly independent eigenvectors of $\mathcal{O}$. One of them, $\Delta_{\mathcal{L}}(1) = \mathbf{1}\mathbf{1}^T$, corresponds to the eigenvalue $\lambda_{\mathcal{L}}(1) = 1$. The remaining $2(N-1)$ correspond to the eigenvalues $\lambda_{\bar{F}}(2),\ldots,\lambda_{\bar{F}}(N)$, taken twice. According to Proposition 3.3.17, R is then the largest among the eigenvalues left when removing (twice) $[\lambda_{\bar{F}}(2),\ldots,\lambda_{\bar{F}}(N)]$ from $[\lambda_{\mathcal{L}}(2),\ldots,\lambda_{\mathcal{L}}(N^2)]$. ■

Clustering choices and performances

The rate of convergence R and the bound β has been analyzed for different networks and different decomposition choices. For some of them (namely for the 1-dimensional case) it is possible to compute them analytically, gaining also some insight on how they scale with the number of nodes. For a more general case, analytic results have been completed with numerical estimates and compared with simulations. Moreover, a bound on the best β has been found, showing how a specific clustering choice provides the best performance.

Consider the specific case of a 1-dimensional graph, i.e. an electrical network consisting in one single line with compensators equally distributed at unitary distances along the line[4]. Loads (passive agents) can be connected everywhere in this line, as their location and their demands are uninfluential on the matrix M and on the speed of convergence of the optimization algorithm.

Three different decompositions of the optimization problem have been considered, corresponding to different clustering of the nodes into subsets. In all of them that compensators have been allowed to update their state in pairs. This can be conveniently described by

[4]The hypothesis of equally spaced compensators on the line allows easier comparison between strategies and simplifies the analytical results, but it not critical. Indeed, some of the rates and bounds are independent from the distance between compensators. These cases will be pointed out later, when they appear in the text.

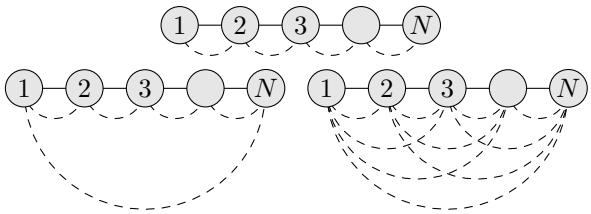


Figure 3.24: Three possible clustering choices (*1-step*, *circle*, *complete*) illustrated via their corresponding hyper-graphs $\mathcal{H}$ (in dashed line). Edges of $\mathcal{H}$ connect nodes that are allowed to update their state together (each edge corresponds to a different subproblem). The graph in continuous line describes the physical system.

an hyper-graph $\mathcal{H}$ (a graph in this case) where an edge connecting node i with node j corresponds to the optimization subproblem in which only the states q_i and q_j are updated.

The three following graphs $\mathcal{H}$ have been considered in the analysis (see Figure 3.24), corresponding to different clustering choices:

- edges of $\mathcal{H}$ connect compensators which are adjacent in the electric line (*1-step*);
- edges of $\mathcal{H}$ connect compensators which are adjacent in the electric line and moreover the first agent is connected with the last agent (*circle*);
- edges of $\mathcal{H}$ connect any pair of compensators (*complete*).

The Hessian M for the 1-dimensional electric network takes the form

$$M = M_0 - \begin{bmatrix} 0 & 1 & 2 & \cdots & N-1 \\ 1 & 1 & 2 & & N-1 \\ 2 & 2 & 2 & & N-1 \\ \vdots & & & & \vdots \\ N-1 & N-1 & N-1 & \cdots & N-1 \end{bmatrix},$$

where $M_0 = m_0 \mathbf{1}\mathbf{1}^T$ and therefore it can be safely ignored, as $x^T M_0 x = 0$ for all $x \in \ker \mathbf{1}^T$.

For the 1-step case, the i-th element of $\{F_i\}_{i=1}^{N-1}$ corresponds to the subproblem in which

node i and node $i+1$ are allowed to update their state:

$$F_i = \begin{bmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ -1 & \cdots & -1 & 0 & & & & \\ 1 & \cdots & \cdots & \cdots & 1 & & & \\ & & & & & \ddots & & \\ & & & & & & 1 & \end{bmatrix}.$$

It is possible to analytically compute $\bar{F}$, which results to be a lower triangular matrix with elements on the diagonal (and eigenvalues):

$$\lambda_{\bar{F}} = \left[1, \underbrace{1-\frac{1}{N-1}, \ldots, 1-\frac{1}{N-1}}_{N-1}\right]^T.$$

In this specific case it is also possible to compute the matrix $\mathbf{F}$ associated with the linear transformation $\mathcal{L}$, which results to be an upper triangular $N^2 \times N^2$ matrix whose elements on the diagonal are

$$\lambda_{\mathcal{L}} = \left[1, \underbrace{1-\frac{1}{N-1}, \ldots, 1-\frac{1}{N-1}}_{3(N-1)}, \underbrace{1-\frac{2}{N-1}, \ldots, 1-\frac{2}{N-1}}_{(N-1)(N-2)}\right]^T.$$

Therefore, from the analysis of the previous section, it follows

$$\beta_{1\text{step}} = R_{1\text{step}} = 1 - \frac{1}{N-1}.$$

Interestingly, both $\beta_{1\text{step}}$ and $R_{1\text{step}}$ do not depend on the length of the electric paths between adjacent compensators.

Consider now the case in which the graph $\mathcal{H}$ is a circle, i.e. every node can communicate with its closest neighbors on the line and in addition an edge connects the first and the last node.

In this case the set of matrices $\{F_i\}_{i=1}^N$ includes the F_i's of the previous case (1-step),

together with

$$F_N = \begin{bmatrix} 0 & \frac{1}{N-1}-1 & \frac{2}{N-1}-1 & \cdots & 0 \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ 1 & 1-\frac{1}{N-1} & \cdots & 1-\frac{N-2}{N-1} & 1 \end{bmatrix}.$$

Also in this case, by exploiting the block-triangular structure of the resulting $\bar{F}$, it is possible to list its eigenvalues:

$$\lambda_{\bar{F}} = \left[1, \underbrace{1-\frac{1}{N}, \ldots, 1-\frac{1}{N}}_{N-2}, 1-\frac{2}{N}\right]^T,$$

and therefore obtain

$$\beta_{\text{circle}} = 1 - \frac{1}{N}.$$

β_{circle} too does not depend on the length of the paths between compensators.

In this case, however, $\lambda_{\mathcal{L}}$ (and therefore R) cannot be easily expressed analytically as for the *1step* case. R_{circle} has then be computed numerically, together with the both the bound β_{complete} and the exact rate R_{complete} for the third clustering choice, in which every couple of nodes is allowed to communicate.

In Table 3.4 it is possible to compare the rate of convergence of these different clustering (or decomposition) choices for different values of N, and to realize how the bound is tight. The tightness of the bound justifies the choice of including in the table also the a larger network ($N = 500$), for which the problem of computing the exact convergence rate R results to be numerically intractable.

It is worth noticing that the well studied problem of randomized gossip algorithms for average consensus can be casted into the framework of this paper by choosing $M = I$. These results are therefore quite interesting in the fact that they contrast with the phenomena generally observed in gossip consensus algorithms (e.g. Fagnani and Zampieri 2008), in which long-distance communication, by decreasing the diameter of the graph, tends to be extremely beneficial for the rate of convergence.

Similar result can be obtained for a more general tree $\mathcal{T}$. Consider the matrix $L \in \mathbb{R}^{N \times N-1}$,

	$N=10$		$N=100$		$N=500$
	$1-\beta$	$1-R$	$1-\beta$	$1-R$	$1-\beta$
1-step	0.11111	0.11111	0.010101	0.010101	20.040×10^{-4}
circle	0.10000	0.10572	0.010000	0.010051	20.000×10^{-4}
complete	0.03129	0.05196	0.000427	0.000560	0.204×10^{-4}

Table 3.4: Exact convergence rate R and bound β for different network lengths N and different communication topologies.

whose elements are

$$L_{i,j}=\begin{cases}-1 & \text{if } i=j+1\\ 1 & \text{if } i=p(j+1)\\ 0 & \text{otherwise}\end{cases}$$

where $p(k)$ is the parent of node k in the tree.

It can be shown that

$$\begin{bmatrix}K\\ \mathbf{1}^T\end{bmatrix}\begin{bmatrix}L & e_1\end{bmatrix}=I,$$

where K has been defined in (3.27) and e_1 is the vector of zeros everywhere but in position 1. The Hessian M of the optimization problem can be rewritten as

$$M=\begin{bmatrix}K^T & \mathbf{1}\end{bmatrix}\begin{bmatrix}D & 0\\ 0 & \gamma\end{bmatrix}\begin{bmatrix}K\\ \mathbf{1}^T\end{bmatrix}$$

where $\gamma>0$ and D has been defined in (3.30). Therefore the Hessian inverse can be rewritten as

$$M^{-1}=\begin{bmatrix}L & e_1\end{bmatrix}\begin{bmatrix}D^{-1} & 0\\ 0 & \gamma^{-1}\end{bmatrix}\begin{bmatrix}L^T\\ e_1^T\end{bmatrix}.$$

By Theorem 3.3.16 and by (3.51),

$$\beta=\lambda_2(\bar{F})=1-\lambda_{N-1}\left(\bar{E}\right),$$

where $\bar{E}=\mathbb{E}\left[(\Omega_i M\Omega_i)^\sharp M\right]$. In the 1-step clustering strategy $\bar{E}=\sum_{i=2}^{N}p_i v_i(v_i^T M v_i)^{-1}v_i^T M$, where p_i is the probability of triggering the subset corresponding to the edge from $p(i)$ to i, and $v_i=e_i-e_{p(i)}$. The vector v_i is the $i-1$th column of L, and therefore, by introducing

$P = \mathrm{diag}(p_2, \dots, p_N)$,

$$\begin{aligned}
\bar{E} &= LP(L^T M L)^{-1} L^T M \\
&= LP \left(L^T \begin{bmatrix} K^T & \mathbf{1} \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & \gamma \end{bmatrix} \begin{bmatrix} K \\ \mathbf{1}^T \end{bmatrix} L \right)^{-1} L^T \begin{bmatrix} K^T & \mathbf{1} \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & \gamma \end{bmatrix} \begin{bmatrix} K \\ \mathbf{1}^T \end{bmatrix} \\
&= LPD^{-1}DK \\
&= \begin{bmatrix} L & e_1 \end{bmatrix} \begin{bmatrix} P & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} K \\ \mathbf{1}^T \end{bmatrix} \\
&= T^{-1} \begin{bmatrix} P & 0 \\ 0 & 0 \end{bmatrix} T.
\end{aligned}$$

Therefore $\lambda_{N-1}\left(\bar{E}\right) = \lambda_N(P)$ and then

$$\beta \geq 1 - \frac{1}{N-1}.$$

Moreover, as $\mathrm{Tr}(P) = 1$ and $p_i > 0$ for all i, the smallest β is obtained when all p_i are the same, i.e. the clusters are triggered with equal probability.

The following result gives also a bound on the smallest possible value for β that can be achieved with different clustering choices. Interestingly, the *1-step* strategy returns exactly this bound, therefore proving to be the optimal clustering strategy for this problem (at least according to the bound β, which has been proved to be very tight).

Theorem 3.3.19. *Consider a tree $\mathcal{T}$, with arbitrary electric lengths r_i associated to its edges. Consider the iterative algorithm (3.48), where the clusters $\mathcal{C}_i$ are the edges of an arbitrary connected graph $\mathcal{H}$ on the set of compensators $\mathcal{C}$, and they are triggered with arbitrary (non zero) probabilities. Then the bound β on the convergence rate of the algorithm, as defined in Theorem 3.3.16, satisfies*

$$\beta \geq 1 - \frac{1}{N-1}.$$

Proof. The sum of the eigenvalues of $\bar{E} = \mathbb{E}\left[(\Omega_i M \Omega_i)^\sharp M\right]$ is

$$\begin{aligned}\sum_{i=1}^{N} \lambda_i(\bar{E}) &= \operatorname{Tr}\left[\sum_{i=2}^{N} p_i v_i (v_i^T M v_i)^{-1} v_i^T M\right] \\ &= \sum_{i=2}^{N} p_i \operatorname{Tr}\left[v_i (v_i^T M v_i)^{-1} v_i^T M\right] \\ &= \sum_{i=2}^{N} p_i \operatorname{Tr}\left[v_i^T M v_i (v_i^T M v_i)^{-1}\right] \\ &= \sum_{i=2}^{N} p_i = 1.\end{aligned}$$

As $\lambda_N(\bar{E}) = 0$, then $\lambda_{N-1}(\bar{E}) \leq 1/(N-1)$, and therefore $\beta \geq 1 - \frac{1}{N-1}$. ∎

This result has been confirmed via numerical simulations on a test tree of height, with 33 nodes and an average of 2.4 children for every node that is not a leaf.

Two clustering choices have been implemented: in the first one, only nodes that are neighbors on the tree can communicate (*1-step*); in the second one, every pair of node is allowed to communicate (*complete*).

The following values for the convergence rates and the bound β have been numerically computed:

$$\begin{aligned}\beta_{1\text{step}} &= 0.9688 = 1 - \frac{1}{32}, & R_{1\text{step}} &= 0.9688, \\ \beta_{\text{complete}} &= 0.9967, & R_{\text{complete}} &= 0.9937.\end{aligned}$$

In the upper part of Figure 3.25 the signal $\bar{v}(t)$, corresponding to the cost function averaged over 100 realizations, has been plotted for the two strategies. In the lower part, instead, the function $\bar{v}(t)^{1/t}$ has been compared with the computed rates of convergence (and bounds).

Even in this case, it is clear how adding long distance links (i.e. enabling communication between agents with are connected to distant points of the distribution network) is detrimental for the convergence speed of the algorithm. On the contrary, the optimal strategy consists in choosing a clustering hyper-graph which resembles (or is the same in the case of clusters of two nodes) the graph describing the physical interconnection of the electric network.

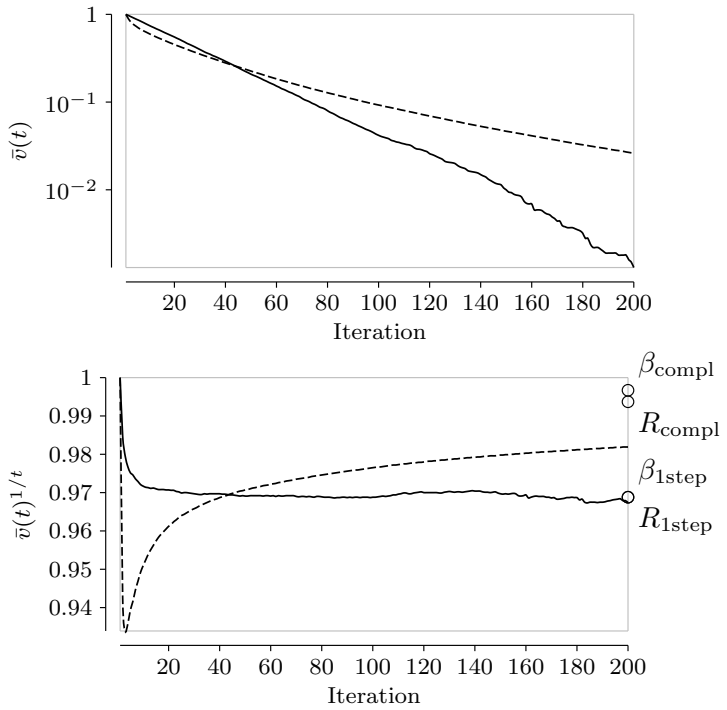


Figure 3.25: Algorithm behavior when applied to the generic case (tree). The continuous line refers to the *1-step* communication strategy, while the dashed line refers to the *complete* one.

Comments

Among the great number of possible applications of NCS to smart power distribution networks, the design of distributed optimization methods for reactive power compensation results to be effective and motivating.

Because the problem of optimal reactive power flows in a microgrid has been casted into the much wider class of convex quadratic optimization problems, the algorithms and the results presented in this section can be applied to completely different scenarios. In particular, it should be of interest for those distributed optimization problems in which the global cost function that have to be minimized is not separable into agents' individual additive term, and when one or more linear equality constraints couple the whole set of decision variables and do not allow independent updates.

There is however a characterizing aspect of the problem of optimal reactive power flows that is critical in both the presented algorithms: because of the physical laws of the underlying system, agents can estimate the gradient of the cost function (which is, in general, function of the whole state) from local measurements and from the information that they can gather in their neighborhood. To exploit this fact, the steps of the designed optimization algorithms must alternate the operations of sensing, processing and actuating the system. As showed in

Section 2.3, this aspect of the problem is among the most important facts that differentiate the application of distributed optimization methods in NCS from the methodologies available in the literature of distributed and parallel computation.

Among the issues that characterize NCS, the following have been addressed for this specific application:

- **Multi-agent architecture** – the development of an effective architecture is one of the critical and open issues in smart power distribution systems; by adopting the concept of microgrids and with the clustering idea proposed in the gossip-like algorithm, the concept of hierarchical structure has been kept in mind;

- **Distributed information** – it has been assumed that the agents in the system only know a small fraction of the system structure and state, corresponding to their neighborhood in the electrical graph; to deal with this issue, the proposed algorithms have been designed to build an estimate of the local system structure, to share data effectively (also via consensus), and to exploit the physics of the underlying system to avoid unnecessary exchange of information;

- **Communication constraints** – the communication graph among agents is not required to resemble the electrical grid, and different other options have been considered; however, it is remarkable to see how the proposed algorithms exhibit better performances when this correspondence is enforced, both in the quality of the quasi-Newton estimation process and in the performance of the gossip-like algorithm;

- **Interaction with an underlying physical system** – as remarked before, it is critical in this application how the agents exploit and have to cope with the physics of the underlying system; this also motivates a great number of directions of investigation about the dynamical aspect of this interaction;

- **Performances** – the precise analysis of the algorithm performances in the gossip-like method is of particular interest; while the algorithm is similar to the class of consensus problems, the dependence of the rate of convergence on the communication graph is very specific and does not follow some of the main intuitions developed in the general cases.

Part II

Feedback control design for quantum systems

4 Introduction

Since the pioneering intuitions in Feynman (1982), Quantum Information (QI) has been the focus of an impressive research effort. Its potential has been clearly demonstrated, not only as a new paradigm for fundamental physics, but also as the key ingredient for a new generation of information technologies. Today the goal is to design and produce quantum chips, quantum memories, and quantum secure communication protocols (Everitt, 2005; Nielsen and Chuang, 2002; Bouwmeester, Ekert, and Zeilinger, 2000).

The main difficulties in building effective QI processing devices are mainly related to scalability issues and to the disruptive action of the environment on the quantum correlations that embody the key advantage of QI. Many of these issues do not appear to be fundamental, and their solution is becoming mainly an engineering problem. Most of the proposed approaches to realize quantum information technology require the ability to perform sequences of a limited number of fundamental operations. Two typical key tasks are concerned with the preparation of states of maximal information (Di Vincenzo, 1995; Nielsen and Chuang, 2002; Viola, Knill, and Laflamme, 2001) and engineering of protected realization of quantum information (Viola, Knill, and Lloyd, 1999; Lidar, Chuang, and Whaley, 1997; Knill, Laflamme, and Viola, 2000; Knill, 2006), i.e. the realization of information encodings that preserve the fragile quantum states from the action of noise.

This paper will focus on these issues, providing a design strategy for engineering stable quantum subspaces.

While controllability issues and optimal and Lyapunov control design techniques are well established and provide effective tools for steering quantum systems driven by Hamiltonian dynamics (see e.g. D'Alessandro 2007; D'Alessandro and Dahleh 2001; Khaneja, Brockett, and Glaser 2001; Grivopoulos and Bamieh 2003; Ferrante, Pavon, and Raccanelli 2002; Altafini 2007; Wang and Schirmer 2010), their potential for state-preparation, as well as state purification and cooling problems, is limited by the isospectral character of unitary evolutions. Hence, feedback state stabilization, and in particular pure-state stabilization problems, have been tackled in the quantum domain under a variety of modeling and control assumptions, with a rapidly growing body of work dealing with the Lyapunov approach, see e.g. Wiseman (1994); van Handel, Stockton, and Mabuchi (2005); Mirrahimi, Rouchon, and Turinici (2005); Ticozzi and Viola (2008, 2009) and references therein. Lyapunov analysis of quantum discrete-time semigroups has been also considered, with emphasis on ergodicity properties, in Burgarth and Giovannetti (2007).

In this thesis unitary controllability has been assumed as a starting point, show how some of the limitations of this framework can be overcome with the aid of discrete-time measurements and a feedback control policy.

The same analysis and synthesis techniques are relevant in the preparation of quantum information in noiseless subspaces: in Shabani and Lidar (2005) and Ticozzi and Viola (2008) the seminal linear-algebraic approach of Lidar et al. (1997) to study noise-free subspaces has been extended to the general setting of noiseless subsystems (which usually entails an operator-algebraic approach, see Knill et al. 2000) and developed in two different directions, both concerned with the *robustness* of the encoded quantum information. In Shabani and Lidar (2005), it has been studied the cases in which the encoded information does not degenerate in the presence of initialization errors; Ticozzi and Viola (2008), instead, aims to ensure that the chosen encoding is an invariant, asymptotically stable set for the dynamics in presence of the noise. The latter tightly connects the encoding task to a set of familiar *stabilization* control problems.

The approach of Ticozzi and Viola (2008, 2009) has been later embraced in Bolognani and Ticozzi (2010b,a), where those techniques have been extended to Markovian discrete-time evolutions.

A good review of the role of discrete-time models for quantum dynamics and control problems can be found in Bouten, van Handel, and James (2009). In this work discrete-time quantum dynamics will be described by sequences of trace-preserving quantum operations in Kraus representation Kraus (1983); Nielsen and Chuang (2002). This choice implies the Markovian character of the evolution Kummerer (2006), which, along with a forward composition law, ensures a semigroup structure. The class of dynamics of interest and the

relevant notation will be introducted in Chapter 5, together with a quick review of the formalism of generalized measurement. A basic analysis of kinematic controllability for Kraus maps has been provided in Wu, Pechen, Brif, and Rabitz (2007).

After recalling the key concepts relative to *quantum subspaces* and dynamical stability, Section 5.3 is devoted to the analysis of the dynamics. The results provide necessary and sufficient conditions on the dynamical model that ensure global stability of a certain quantum subspace. A Lyapunov approach is employed, exploiting the linearity of the dynamics, as well as the convex character of the state manifold. With respect to the analysis in Ticozzi and Viola (2008, 2009) the main difference lies in the fact that the dynamical contributions given from the Hamiltonian and the noise terms cannot be separated, as one would be able to do when considering continuous-time model like the GKS-Lindblad master equations. In addition to this, the normalization condition on the Kraus operators needs to be taken into account.

The control scheme introduced in Chapter 7 modifies the underlying dynamics of the system by indirectly measuring it, and applying unitary control actions, conditioned on the outcome of the measurement. By *averaging over the possible outcomes*, a new semigroup evolution is obtained, where the choice of the control can be used to achieve the desired stabilization. This control scheme can be seen as an instance of discrete-time *Markovian reservoir engineering*: the use of "noisy" dynamics to obtain a desired dynamical behavior has long been investigated in a variety of contexts, see e.g. Poyatos, Cirac, and Zoller (1996); Carvalho, Milman, de Matos Filho, and Davidovich (2001); de Matos Filho and Vogel (1998); Verstraete, Wolf, and Ignacio Cirac (2009).

The synthesis results of Chapter 7 include a simple characterization of the controlled dynamics that can be enacted, and an algorithm that builds unitary control actions stabilizing the desired subspace. If such controls cannot be found, it is proven that no choice of controls can achieve the control task for the same measurement. The main tools employed for this result come from the stability theory of dynamical systems, namely Krasovskii-LaSalle's Invariance principle LaSalle (1980), and linear algebra, namely the QR matrix decomposition Horn and Johnson (1990). A "special form" of the QR decomposition need to be constructed: in particular, it will be proved that the upper triangular factor R can be rendered a *canonical form* with respect to the left action of the unitary matrix group. This result and the related discussion is presented in Chapter 6.

5

Discrete-time open quantum systems

Let $\mathcal{I}$ denote the physical quantum system of interest. Consider the associated separable Hilbert space $\mathcal{H}_I$ over the complex field $\mathbb{C}$. Finite-dimensional quantum systems are considered, i.e. $\dim(\mathcal{H}_I) < \infty$. In Dirac's notation, vectors are represented by a *ket* $|\psi\rangle \in \mathcal{H}_I$, and linear functionals by a *bra*, $\langle\psi| \in \mathcal{H}_I^\dagger$ (the adjoint of $\mathcal{H}_I$), respectively. The inner product of $|\psi\rangle, |\varphi\rangle$ is then represented as $\langle\psi|\varphi\rangle$.

Let $\mathfrak{B}(\mathcal{H}_I)$ represent the set of linear bounded operators on $\mathcal{H}_I$, $\mathfrak{H}(\mathcal{H}_I)$ denoting the real subspace of hermitian operators, with $\mathbb{I}$ and $\mathbb{O}$ being the identity and the zero operator, respectively. A (possibily uncertain) knowledge of the state of the quantum system is condensed in a density operator, or *state* ρ, with $\rho \geq 0$ and $\operatorname{Tr}\rho = 1$. Density operators form a convex set $\mathfrak{D}(\mathcal{H}_I) \subset \mathfrak{H}(\mathcal{H}_I)$, with one-dimensional projectors corresponding to extreme points (pure states, $\rho_{|\psi\rangle} = |\psi\rangle\langle\psi|$). Given an $X \in \mathfrak{H}(\mathcal{H}_I)$, $\ker(X)$ indicates its kernel (0-eigenspace) and $\operatorname{supp}(X) := \mathcal{H}_I \ominus \ker(X)$ indicates its range, or *support*. If a quantum system $\mathcal{Q}$ is obtained by composition of two *subsystems* $\mathcal{Q}_1$, $\mathcal{Q}_2$, the corresponding mathematical description is carried out in the tensor product space, $\mathcal{H}_{12} = \mathcal{H}_1 \otimes \mathcal{H}_2$ (Sakurai, 1994), observables and density operators being associated with Hermitian and positive-semidefinite, normalized operators on $\mathcal{H}_{12}$, respectively.

5.1 Quantum measurements

Projective Quantum Measurements

In quantum mechanics, observable quantities are associated to Hermitian operators, with their spectrum associated to the possible outcomes. Consider the measure of the *observable* $C = \sum_i c_i \Pi_i$. The basic postulates that describe the quantum (strong, projective, or von Neumann's) measurements are the following:

(i') The probability of obtaining c_i as the outcome of a measure on a system described by the density operator ρ is $p_i = \mathrm{Tr}(\rho \Pi_i)$.

(ii') (Lüders's Postulate) Immediately after a measurement that gives c_i as an outcome the system state becomes: $\rho|_i = \frac{1}{\mathrm{Tr}(\Pi_i \rho \Pi_i)} \Pi_i \rho \Pi_i$.

Notice that the spectrum of the observable does not play any role in the computation of the probabilities.

Generalized Quantum Measurements

If the information about a quantum system is obtained by measuring another system which is correlated to the former, the projective measurement formalism is not enough, but it can be used to derive a more general one. A typical procedure to obtain generalized measurements on a quantum system of interest is the following:

- the system of interest $\mathcal{A}$ is augmented by adding another subsystem $\mathcal{B}$, initially decoupled from $\mathcal{A}$. Let $\rho_\mathcal{A} \otimes \rho_\mathcal{B}$, with $\rho_\mathcal{B} = |\phi\rangle\langle\phi|$, be the joint state;
- the two systems are coupled through a joint unitary evolution $U_{\mathcal{AB}}$;
- a direct, von Neumann measurement of an observable $X_\mathcal{B} = \sum_j x_j \Pi_j$, $\Pi_j = |\xi_j\rangle\langle\xi_j|$, is performed on $\mathcal{B}$;
- the conditioned state of the joint system after the measurement is then of the form

$$\rho_{\mathcal{AB}}|_j = \frac{1}{p_j}(I_\mathcal{A} \otimes \Pi_j) U_{\mathcal{AB}} (\rho_\mathcal{A} \otimes \rho_\mathcal{B}) U_{\mathcal{AB}}^\dagger (I_\mathcal{A} \otimes \Pi_j) = \rho'_{\mathcal{A},j} \otimes \Pi_j,$$

  with p_j the probability of obtaining the j-th outcome;
- one can compute the effect of the measurement on $\mathcal{A}$ alone, which is nontrivial if $U_{\mathcal{AB}}$ entangled the two subsystems, i.e. $U_{\mathcal{AB}}(\rho_\mathcal{A} \otimes \rho_\mathcal{B}) U_{\mathcal{AB}}^\dagger$ cannot be written in factorized form; one then gets that $\rho'_{\mathcal{A},j} = \frac{1}{p_j} M_j \rho_\mathcal{A} M_j^\dagger$, with $M_j = \langle \xi_j | U_{\mathcal{AB}} | \phi \rangle$.

By taking the average over the possible outcomes, a state transformation in Kraus form is obtained. This construction is actually general, in the sense that if the dimension of $\mathcal{B}$ corresponds (at least) to the necessary number of outcomes, *any* Kraus map can be actually generated this way.

5.2 Discrete-time quantum dynamical semigroups

In the presence of coupling between subsystems, quantum measurements, or interaction with surrounding environment, the dynamics of a quantum system cannot in general be described by Schrödinger's dynamics: the evolution is no longer unitary and reversible, and the formalism of open quantum systems is required (Davies, 1976; Breuer and Petruccione, 2006; Alicki and Lendi, 1987; Nielsen and Chuang, 2002). An effective tool to describe these dynamical systems, of fundamental interest for QI, is given by quantum operations (Nielsen and Chuang, 2002; Kraus, 1983). The most general, linear and physically admissible evolutions which take into account interacting quantum systems and measurements, are described by Completely Positive (CP) maps, that via the Kraus-Stinespring theorem (Kraus, 1983) admit a representation of the form

$$\mathcal{T}[\rho] = \sum_k M_k \rho M_k^\dagger \tag{5.1}$$

(also known as operator-sum representation of $\mathcal{T}$), where ρ is a density operator and $\{M_k\}$ a family of operators such that the completeness relation

$$\sum_k M_k^\dagger M_k = I \tag{5.2}$$

is satisfied. Under this assumption the map is then Trace-Preserving and Completely-Positive (TPCP), and hence maps density operators to density operators. In Alicki and Lendi (1987); Nielsen and Chuang (2002); Breuer and Petruccione (2006); Davies (1976) a detailed discussions of the properties of quantum operations and the physical meaning of the complete-positivity property is available.

One can then consider the discrete-time dynamical semigroup, acting on $\mathfrak{D}(\mathcal{H}_I)$, induced by iteration of a given TPCP map. The resulting discrete-time quantum system is described by

$$\rho(t+1) = \mathcal{T}[\rho(t)] = \sum_k M_k \rho(t) M_k^\dagger. \tag{5.3}$$

Given the initial conditions $\rho(0)$ for the system, it results $\rho(t) = \mathcal{T}^t[\rho(0)],\ t = 1, 2, \ldots$

where $\mathcal{T}^t[\cdot]$ indicates t applications of the TPCP map $\mathcal{T}[\cdot]$. Hence, the evolution obeys a forward composition law and, in the spirit of Alicki and Lendi (1987), is called a Discrete-time Quantum Dynamical Semigroup (DQDS). Notice that while the dynamic map is linear, the "state space" $\mathfrak{D}(\mathcal{H}_I)$ is a convex, compact subset of the cone of the positive elements in $\mathfrak{H}(\mathcal{H}_I)$.

While a TPCP maps can indeed represent general dynamics, assuming dynamics of the form (5.3), with M_k's that do not depend on the past states, is equivalent to assume Markovian dynamics (see Kummerer 2006 for a discussion of Markovian properties for quantum evolutions). From a probabilistic viewpoint, if density operators play the role of probability distributions, TPCP maps are the analogue of transition operators for classical Markov chains.

5.3 Quantum subspaces, invariance and attractivity

Follow the subsystem approach of Ticozzi and Viola (2008, 2009), and focusing on the case of subspaces, some definitions of quantum subspaces invariance and attractivity are now recalled. This is motivated by the fact that the general subsystem case is derived in the continuous-time case as a specialization with some additional constraints, and that for many applications of interest for the present work, namely *pure-state preparation* and *engineering of protected quantum information*, the subspace case is enough, as it is suggested by the results in Ticozzi and Viola (2008).

Definition 5.3.1 (Quantum subspace)**.** A quantum subspace $\mathcal{S}$ of a system $\mathcal{I}$ with associated Hilbert space $\mathcal{H}_I$ is a quantum system whose Hilbert space is a subspace $\mathcal{H}_S$ of $\mathcal{H}_I$,

$$\mathcal{H}_I = \mathcal{H}_S \oplus \mathcal{H}_R, \tag{5.4}$$

for some remainder space $\mathcal{H}_R$. The set of linear operators on $\mathcal{S}$, $\mathcal{B}(\mathcal{H}_S)$, is isomorphic to the algebra on $\mathcal{H}_I$ with elements of the form $X_I = X_S \oplus \mathbb{O}_R$.

Let $n = \dim(\mathcal{H}_I)$, $m = \dim(\mathcal{H}_S)$, and $r = \dim(\mathcal{H}_R)$, and let $\{|\phi\rangle_j^S\}_{j=1}^m$, $\{|\phi\rangle_k^R\}_{k=1}^r$ denote orthonormal bases for $\mathcal{H}_S$ and $\mathcal{H}_R$, respectively. Decomposition (5.4) is then naturally associated with the following basis for $\mathcal{H}_I$:

$$\{|\varphi_l\rangle\} = \{|\phi\rangle_j^S\}_{j=1}^m \cup \{|\phi\rangle_k^R\}_{k=1}^r.$$

This basis induces a block structure for matrices representing operators acting on $\mathcal{H}_I$:

$$X = \left[\begin{array}{c|c} X_S & X_P \\ \hline X_Q & X_R \end{array}\right].$$

In the rest of the paper the subscripts S, P, Q and R will follow this convention. Let moreover Π_S and Π_R be the projection operators over the subspaces $\mathcal{H}_S$ and $\mathcal{H}_R$, respectively. The following definitions are independent of the choices of $\{|\phi\rangle_j^S\}_{j=1}^m$, $\{|\phi\rangle_k^R\}_{k=1}^r$.

Definition 5.3.2 (State initialization)**.** The system $\mathcal{I}$ with state $\rho \in \mathcal{D}(\mathcal{H}_I)$ is initialized in $\mathcal{S}$ with state $\rho_S \in \mathcal{D}(\mathcal{H}_S)$ if ρ is of the form

$$\rho = \left[\begin{array}{c|c} \rho_S & 0 \\ \hline 0 & 0 \end{array}\right]. \tag{5.5}$$

Let $\mathcal{J}_S(\mathcal{H}_I)$ denote the set of states of the form (5.5) for some $\rho_S \in \mathcal{D}(\mathcal{H}_S)$.

Definition 5.3.3 (Invariance)**.** Let $\mathcal{I}$ evolve under iterations of a TPCP map. The set $\mathcal{J}_S(\mathcal{H}_I)$ is invariant if the evolution of any initialized $\rho \in \mathcal{J}_S(\mathcal{H}_I)$ obeys

$$\rho(t) = \left[\begin{array}{c|c} \mathcal{T}_S^t[\rho_S] & 0 \\ \hline 0 & 0 \end{array}\right] \in \mathcal{J}_S(\mathcal{H}_I)$$

$\forall t \geq 0$, and with $\mathcal{T}_S$ being a TPCP map on $\mathcal{H}_S$.

Definition 5.3.4 (Attractivity)**.** Let $\mathcal{I}$ evolve under iterations of a TPCP map $\mathcal{T}$. The set $\mathcal{J}_S(\mathcal{H}_I)$ is attractive if

$$\lim_{t\to\infty} \left\| \mathcal{T}^t(\rho) - \Pi_S \mathcal{T}^t[\rho] \Pi_S \right\| = 0$$

for all $\rho \in \mathcal{D}(\mathcal{H}_I)$.

Definition 5.3.5 (Global asymptotic stability)**.** Let $\mathcal{I}$ evolve under iterations of a TPCP map $\mathcal{T}$. The set $\mathcal{J}_S(\mathcal{H}_I)$ is *Globally Asymptotically Stable* (GAS) if it is invariant and attractive.

Characterization of invariance and global asymptotic stability

Necessary and sufficient conditions on the form of the TPCP map $\mathcal{T}$ for *a given* quantum subspace $\mathcal{S}$ to be GAS will now be presented.

The following proposition gives a sufficient and necessary condition on $\mathcal{T}$ such that $\mathcal{J}_S(\mathcal{H}_I)$ is invariant.

Proposition 5.3.6. *Let the TPCP transformation $\mathcal{T}$ be described by the Kraus map (5.1). Let the matrices M_k be expressed in their block form*

$$M_k = \begin{bmatrix} M_{k,S} & M_{k,P} \\ M_{k,Q} & M_{k,R} \end{bmatrix}$$

according to the state space decomposition (5.4). Then the set $\mathcal{J}_S(\mathcal{H}_I)$ is invariant if and only if

$$M_{k,Q} = 0 \quad \forall k\,. \tag{5.6}$$

Proof. Verifying Definition 5.3.3 is equivalent to verifying that there exists a TPCP map $\mathcal{T}_S$ such that

$$\mathcal{T}\left[\begin{bmatrix} \rho_S & 0 \\ 0 & 0 \end{bmatrix}\right] = \begin{bmatrix} \mathcal{T}_S(\rho_S) & 0 \\ 0 & 0 \end{bmatrix} \tag{5.7}$$

for all ρ_S in $\mathcal{D}(\mathcal{H}_S)$. The block form of the M_k matrices in (5.1) given by the decomposition (5.4) yields

$$\begin{aligned} \mathcal{T}\left[\begin{bmatrix} \rho_S & 0 \\ 0 & 0 \end{bmatrix}\right] &= \sum_k M_k \begin{bmatrix} \rho_S & 0 \\ 0 & 0 \end{bmatrix} M_k^\dagger \\ &= \sum_k \begin{bmatrix} M_{k,S} & M_{k,P} \\ M_{k,Q} & M_{k,R} \end{bmatrix} \begin{bmatrix} \rho_S & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} M_{k,S}^\dagger & M_{k,Q}^\dagger \\ M_{k,P}^\dagger & M_{k,R}^\dagger \end{bmatrix} \\ &= \sum_k \begin{bmatrix} M_{k,S}\rho_S M_{k,S}^\dagger & M_{k,S}\rho_S M_{k,Q}^\dagger \\ M_{k,Q}\rho_S M_{k,S}^\dagger & M_{k,Q}\rho_S M_{k,Q}^\dagger \end{bmatrix} \end{aligned} \tag{5.8}$$

Sufficiency of (5.6) to have invariance of $\mathcal{J}_S(\mathcal{H}_I)$ is trivial. Necessity is given by the fact that the lower right blocks $M_{k,Q}\rho_S M_{k,Q}^\dagger$ are positive semi-definite for all k's, and therefore, for (5.7) to hold, it has to be $M_{k,Q}\rho_S M_{k,Q}^\dagger = 0\ \forall k$. For $M_{k,Q}\rho_S M_{k,Q}^\dagger$ to be zero for any state $\rho_S \in \mathcal{D}(\mathcal{H}_S)$, it has then to be $M_{k,Q} = 0$. Equation (5.7) then implies that the completely-positive transformation

$$\mathcal{T}_S[\rho_S] = \sum_k M_{k,S}\rho_S M_{k,S}^\dagger$$

is also trace preserving. ∎

To derive a characterization of TPCP maps that render a certain $\mathcal{H}_S$ GAS, LaSalle's invariance principle in its discrete-time form is needed (LaSalle, 1980).

Theorem 5.3.7 (La Salle's theorem for discrete-time systems)**.** *Consider a discrete-time system*

$x(t+1) = \mathcal{T}[x(t)]$. *Suppose V is a $\mathcal{C}^1$ function of $x \in \mathbb{R}^n$, bounded below and satisfying*

$$\Delta V(x) = V(\mathcal{T}[x]) - V(x) \leq 0, \quad \forall x \tag{5.9}$$

i.e. $V(x)$ is non-increasing along forward trajectories of the plant dynamics. Then any bounded trajectory converges to the largest invariant subset W contained in the locus $E = \{x | \Delta V(x) = 0\}$.

Being any TPCP map a map from the compact set of density operators to itself, any trajectory is bounded. Let us then consider the function

$$V(\rho) = \mathrm{Tr}(\Pi_R \rho) \geq 0. \tag{5.10}$$

The function $V(\rho)$ is $\mathcal{C}^1$ and bounded from below, and it is a natural candidate for a Lyapunov function for the system. In fact, it represents the probability of the event Π_R (see Section 5.1), that is, the probability that the system is found in the reminder subspace $\mathcal{H}_R$ after the measurement. The following lemma shows that $V(\rho)$ is non-increasing along the trajectories of the system when $\mathcal{J}_S(\mathcal{H}_I)$ is invariant.

Lemma 5.3.8. *Let $\mathcal{T}$ be the generator of a DQDS, and assume the set $\mathcal{J}_S(\mathcal{H}_I)$ to be invariant. Then $V(\rho) = \mathrm{Tr}(\Pi_R \rho)$ satisfies the hypothesis (5.9) of Theorem 5.3.7.*

Proof. The variation of $V(\rho)$ along forward trajectories of the system (5.3) is

$$\begin{aligned} \Delta V(\rho) &= \mathrm{Tr}\left(\Pi_R \mathcal{T}[\rho]\right) - \mathrm{Tr}(\Pi_R \rho) \\ &= \mathrm{Tr}\left[\Pi_R \left(\sum_k M_k \rho M_k^\dagger - \rho\right)\right] \end{aligned} \tag{5.11}$$

Notice that $\mathrm{Tr}(\sum_k M_k \rho M_k^\dagger - \rho) = 0$, and that $V(\rho) = 0$ for all ρ's that have support in $\mathcal{H}_S$. Let us express $\sum_k M_k \rho M_k^\dagger - \rho$ in its block form, using the fact that $M_Q = 0$ by assuming invariance of $\mathcal{J}_S(\mathcal{H}_I)$:

$$\sum_k \begin{bmatrix} M_{k,S} & M_{k,P} \\ 0 & M_{k,R} \end{bmatrix} \begin{bmatrix} \rho_S & \rho_P \\ \rho_P^\dagger & \rho_R \end{bmatrix} \begin{bmatrix} M_{k,S}^\dagger & 0 \\ M_{k,P}^\dagger & M_{k,R}^\dagger \end{bmatrix} - \rho = \begin{bmatrix} \Delta_S[\rho] & \Delta_P[\rho] \\ \Delta_Q[\rho] & \Delta_R[\rho] \end{bmatrix}, \tag{5.12}$$

where

$$\Delta_S[\rho] = \sum_k \left(M_{k,S}\rho_S M_{k,S}^\dagger + M_{k,P}\rho_P^\dagger M_{k,S}^\dagger + M_{k,S}\rho_P M_{k,P}^\dagger + M_{k,P}\rho_R M_{k,P}^\dagger \right) - \rho_S$$
$$\Delta_P[\rho] = \sum_k \left(M_{k,S}\rho_P M_{k,R}^\dagger + M_{k,P}\rho_R M_{k,R}^\dagger \right) - \rho_P$$
$$\Delta_Q[\rho] = \sum_k \left(M_{k,R}\rho_P M_{k,S}^\dagger + M_{k,R}\rho_R M_{k,P}^\dagger \right) - \rho_P^\dagger$$
$$\Delta_R[\rho] = \sum_k \left(M_{k,R}\rho_R M_{k,R}^\dagger \right) - \rho_R.$$

Therefore

$$\Delta V(\rho) = \mathrm{Tr}\left[\Pi_R \left(\sum_k M_k \rho M_k^\dagger - \rho \right) \right] = \mathrm{Tr}\left[\sum_k M_{k,R}\rho_R M_{k,R}^\dagger - \rho_R \right], \tag{5.13}$$

so that in order to get $\Delta V \leq 0$ the map $\mathcal{T}_R[\rho_R] := \sum_k M_{k,R}\rho_R M_{k,R}^\dagger$ has to be trace non-increasing.

Note that this condition is automatically verified, once $\mathcal{T}$ is a TPCP map. Indeed, consider the application of $\mathcal{T}$ on a state $\bar{\rho}$ which has support on $\mathcal{H}_R$. According to the block form in (5.12), the total trace of $\mathcal{T}[\bar{\rho}]$ is

$$\mathrm{Tr}\left(\mathcal{T}[\bar{\rho}]\right) = \mathrm{Tr}\left(\sum_k M_{k,P}\bar{\rho}_R M_{k,P}^\dagger \right) + \mathrm{Tr}\left(\sum_k M_{k,R}\bar{\rho}_R M_{k,R}^\dagger \right).$$

Therefore, as both the terms are positive, being $\bar{\rho}_R \geq 0$, and as $\mathcal{T}$ is TP, it results for any $\bar{\rho}_R \in \mathcal{D}(\mathcal{H}_R)$

$$\mathrm{Tr}\left(\sum_k M_{k,R}\bar{\rho}_R M_{k,R}^\dagger \right) \leq \mathrm{Tr}\left(\mathcal{T}[\bar{\rho}]\right) = \mathrm{Tr}\left(\bar{\rho}_R\right)$$

and thus $\mathcal{T}_R$ is trace non-increasing. ∎

To determine when $\mathcal{J}_S(\mathcal{H}_I)$ contains the largest invariant set in E, conditions will be derived that ensure that no other invariant set W exists in $E = \{\rho | \Delta V(\rho) = 0\}$ such that $\mathcal{J}_S(\mathcal{H}_I) \subset W$.

The next technical result shows that the existence of invariant sets is in fact equivalent to the existence of invariant states on the same support.

Lemma 5.3.9. *Let $\mathcal{T}$ be a TPCP transformation described by the Kraus map (5.1). Consider an orthogonal subspace decomposition $\mathcal{H}_S \oplus \mathcal{H}_R$. Then the set $\mathcal{J}_R(\mathcal{H}_I)$ contains an invariant subset if and only if it contains an invariant state.*

Proof. The "if" part is trivial. On the other hand, $\mathcal{J}_R(\mathcal{H}_I)$ is convex and compact, hence if it contains an invariant subset W it also contains the closure of its convex hull, call it $\bar{W}$. The map $\mathcal{T}$ is linear and continuous, so the convex hull of an invariant subset is invariant, and so is its closure. Hence, by Brouwer's fixed point theorem (Zeidler, 1999) it admits a fixed point $\bar{\rho} \in \bar{W} \subseteq \mathcal{J}_R(\mathcal{H}_I)$. ∎

The next lemma states that if a certain subset of states is invariant, its support is itself an invariant subspace.

Lemma 5.3.10. *Let W be an invariant subset of $\mathcal{D}(\mathcal{H}_I)$ for the TPCP transformation $\mathcal{T}$, and let*

$$\mathcal{H}_W = \mathrm{supp}(W) = \bigcup_{\rho \in W} \mathrm{supp}(\rho).$$

Then $\mathcal{J}_W(\mathcal{H}_I)$ is invariant.

Proof. Let $\hat{W}$ be the convex hull of W. By linearity of dynamics it is easy to show that $\hat{W}$ is invariant too. Furthermore, from the definition of $\hat{W}$, there exists a $\hat{\rho} \in \hat{W}$ such that $\mathrm{supp}(\hat{\rho}) = \mathrm{supp}(\hat{W}) = \mathcal{H}_W$. Consider the decomposition $\mathcal{H}_I = \mathcal{H}_W \oplus \mathcal{H}_W^{\perp}$, and the corresponding matrix partitioning

$$X = \begin{bmatrix} X_W & X_L \\ X_M & X_N \end{bmatrix}.$$

With respect to this partition, $\hat{\rho}_W$ is full rank while $\hat{\rho}_{L,M,N}$ are zero blocks. The state $\hat{\rho}$ is then mapped by $\mathcal{T}$ according to (5.8) (up to a relabeling of the blocks) and therefore, as $\hat{\rho}_W$ is full rank, it has to be $M_{k,M} = 0$ for all k's. By comparing it with the conditions given in Proposition 5.3.6, invariance of $\mathcal{J}_W(\mathcal{H}_I)$ can be inferred. ∎

The following proposition is key in the proof of a characterization of attractive subspaces, since it indicates where to look for invariant sets that are not contained in the target one.

Proposition 5.3.11. *Consider $\bar{\rho} \in \mathcal{J}_R(\mathcal{H}_I)$ and evolving under the TPCP transformation $\mathcal{T}$ described by the Kraus map (5.1). Let the matrices M_k be expressed in the block form*

$$M_k = \begin{bmatrix} M_{k,S} & M_{k,P} \\ 0 & M_{k,R} \end{bmatrix}$$

according to the state space decomposition $\mathcal{H}_S \oplus \mathcal{H}_R$, with $\mathcal{J}_S(\mathcal{H}_I)$ invariant. Then $\bar{\rho} \in E =$

$\{\rho \in \mathcal{D}(\mathcal{H}_I) | \Delta V(\rho) = 0\}$, *where* $V(\rho)$ *is defined by* (5.10)*, if and only if its* ρ_R *block satisfies*

$$supp(\rho_R) \subseteq \bigcap_k \ker\left(M_{k,P}\right).$$

Proof. Note that from the definition of $V(\rho)$ in (5.10), it follows that $\rho \in \mathcal{J}_R(\mathcal{H}_I)$ if and only if $V(\rho) = 1$. Therefore $\bar{\rho} \in E \cap \mathcal{J}_R(\mathcal{H}_I)$ if and only if also $\mathcal{T}[\bar{\rho}] \in \mathcal{J}_R(\mathcal{H}_I)$. Consider then the following expression for $\mathcal{T}[\bar{\rho}]$, obtained from the expansion of $\sum_k M_k \bar{\rho} M_k^\dagger$ in (5.12) specialized to the case $\bar{\rho} \in \mathcal{J}_R(\mathcal{H}_I)$:

$$\mathcal{T}[\bar{\rho}] = \sum_k M_k \bar{\rho} M_k^\dagger = \sum_k \begin{bmatrix} M_{k,P}\rho_R M_{k,P}^\dagger & M_{k,P}\rho_R M_{k,R}^\dagger \\ M_{k,R}\rho_R M_{k,P}^\dagger & M_{k,R}\rho_R M_{k,R}^\dagger \end{bmatrix}. \tag{5.14}$$

By the form of the upper-left block of (5.14), it can be concluded that $\bar{\rho} \in E \cap \mathcal{J}_R(\mathcal{H}_I)$ if and only if $\text{supp}(\rho_R) \subseteq \bigcap_k \ker\left(M_{k,P}\right)$. ∎

Proposition 5.3.11, along with lemmas above, yields the following characterization of global, asymptotical stability of $\mathcal{J}_S(\mathcal{H}_I)$.

Theorem 5.3.12. *Let the TPCP transformation* $\mathcal{T}$ *be described by the Kraus map* (5.1)*. Consider an orthogonal subset decomposition* $\mathcal{H}_S \oplus \mathcal{H}_R$*, with* $\mathcal{J}_S(\mathcal{H}_I)$ *invariant. Let the matrices* M_k *be expressed in their block form*

$$M_k = \begin{bmatrix} M_{k,S} & M_{k,P} \\ 0 & M_{k,R} \end{bmatrix}$$

according to the same state space decomposition. Then the set $\mathcal{J}_S(\mathcal{H}_I)$ *is GAS if and only if there are no invariant states with support on* $\bigcap_k \ker\left(M_{k,P}\right)$.

Proof. Necessity is immediate: if there was an invariant state supported by $\bigcap_k \ker\left(M_{k,P}\right)$, it would have non trivial support on $\mathcal{H}_R$, and therefore $\mathcal{H}_S$ could not be attractive. In order to prove the other implication, consider LaSalle's theorem. By hypotesis, $\mathcal{J}_S(\mathcal{H}_I)$ is invariant and is contained in E, therefore it is contained in the largest invariant set W in the zero-difference locus E. Suppose that $\mathcal{J}_S(\mathcal{H}_I) \subset W$, but $\mathcal{J}_S(\mathcal{H}_I) \neq W$. That is, there exists a set $W \subseteq E$ which is invariant and strictly contains $\mathcal{J}_S(\mathcal{H}_I)$. Therefore its support has to be

$$\mathcal{H}_W = \mathcal{H}_S \oplus \mathcal{H}_{R'}$$

with $\mathcal{H}_{R'}$ subspace of $\mathcal{H}_R$, and by Lemma 5.3.10 $\mathcal{J}_W(\mathcal{H}_I)$ must be invariant too. Consider

then a state $\hat{\rho}$ which belongs to $\mathcal{J}_W(\mathcal{H}_I)$, with non trivial support on $\mathcal{H}_{R'}$, and define

$$\tilde{\rho} = \frac{\Pi_{R'}\hat{\rho}\,\Pi_{R'}}{\mathrm{Tr}(\Pi_{R'}\hat{\rho})} = \begin{bmatrix} 0 & 0 \\ 0 & \tilde{\rho}_R \end{bmatrix}$$

which has support on $\mathcal{H}_{R'}$ only. By construction, $\tilde{\rho}$ is in $\mathcal{J}_W(\mathcal{H}_I)$, and therefore its trajectory is contained in $\mathcal{J}_W(\mathcal{H}_I)$. It is also in E, that is $\Delta V(\tilde{\rho}) = 0$. As $V(\tilde{\rho}) = 1$, then its evolution must be also remain in $\mathcal{H}_{R'} \subseteq \mathcal{H}_R$ at any time. Therefore an invariant set with support on $\mathcal{H}_R$ exists. By reversing the implication, this means that if does not exist an invariant set with support on $\mathcal{H}_R$, then $\mathcal{J}_S(\mathcal{H}_I)$ is the largest invariant set in E. Furthermore, Proposition 5.3.11 indicates that if there is an invariant set in E with support on $\mathcal{H}_R$, its support must actually be contained in

$$\bigcap_k \ker\left(M_{k,P}\right).$$

Therefore, if no such subset exists, attractivity of $\mathcal{J}_S(\mathcal{H}_I)$ is guaranteed by LaSalle's theorem. By Lemma 5.3.9 and Lemma 5.3.10, the existence of an invariant set is equivalent to the existence of an invariant state with support on $\bigcap_k \ker\left(M_{k,P}\right)$. ■

Given the usual decomposition $\mathcal{H}_I = \mathcal{H}_S \oplus \mathcal{H}_R$, let further decompose $\mathcal{H}_R$ in

$$\mathcal{H}_{R'} = \mathcal{H}_R \ominus \bigcap_k \ker\left(M_{k,P}\right) \quad \text{and} \quad \mathcal{H}_{R''} = \bigcap_k \ker\left(M_{k,P}\right)$$

and consider the operation elements M_k in a basis induced by the decomposition $\mathcal{H}_I = \mathcal{H}_S \oplus \mathcal{H}_{R'} \oplus \mathcal{H}_{R''}$:

$$M_k = \left[\begin{array}{c|c|c} M_{k,S} & M_{k,P'} & 0 \\ \hline 0 & M_{k,R1} & M_{k,R2} \\ \hline 0 & M_{k,R3} & M_{k,R4} \end{array}\right].$$

Density operators ρ which have support on the bottom right block clearly belong to $\mathcal{J}_{R''}(\mathcal{H}_I)$. Sufficient, although not necessary, conditions to be sure that no invariant sets have support on that subspace are that $\bigcap_k \ker\left(M_{k,R2}\right) = \{0\}$ and $M_{k,R3} = 0\ \forall k$. This way, the states that have support on $\bigcap_k \ker\left(M_{k,P}\right)$ will be mapped into states which has non-trivial support on $\left[\bigcap_k \ker\left(M_{k,P}\right)\right]^{\perp}$, and therefore no invariant set will exist in $\bigcap_k \ker\left(M_{k,P}\right)$. This intuition will be further developed in Chapter 7, where a control design tool capable of achieving attractivity of a given subspace is obtained.

6

A canonical matrix form based on the QR decomposition

In this chapter some technical results about QR decomposition will be recalled, allowing the development of a new algebraic tool, namely a canonical form with respect to the left action of the unitary matrix group. With this tool it will then be possible to move from the analysis results presented in the previous chapter to an algorithm for the synthesis of stabilizing control laws.

On the uniqueness of QR decomposition

Definition 6.0.13 (QR decomposition, Horn and Johnson 1990)**.** A QR decomposition of a complex-valued square matrix A is a decomposition of A as

$$A = QR,$$

where Q is an orthogonal matrix (meaning that $Q^{\dagger}Q = I$) and R is an upper triangular matrix.

The uniqueness of the QR decomposition of a complex-valued matrix A, both in the case in which A is non-singular and in the case in which it is singular, will now be investigated.

While the real-matrix case is well known (see e.g. Horn and Johnson 1990), a little extra care is needed in the complex case. Consider first the case of A non-singular.

Lemma 6.0.14. *Let*

$$A = Q_1 R_1\,, \quad A = Q_2 R_2$$

be two QR decompositions of the same non-singular *square matrix A. Then R_1 and R_2 only differ for the phase of their rows, that is*

$$R_1 = \Phi R_2\,, \qquad Q_1 = Q_2 \Phi^{-1}$$

where $\Phi = diag\left(e^{j\phi_1}, \ldots, e^{j\phi_n}\right)$.

Proof. Using the fact that A and its factors are non singular, it results

$$Q_2^\dagger Q_1 = R_2 R_1^{-1} = \Phi$$

where Φ is upper triangular because it is the product of two upper triangular matrices. The matrix Φ must also be orthonormal, because it is the product of two orthonormal matrices. Therefore, starting from the first column of Φ, it results $|\Phi_{11}| = 1$, where Φ_{ij} is the element of Φ in position (i, j).

Proceed now by induction on the column index j. Assume that all the columns Φ_k with $k < j$ satisfy $\Phi_{lk} = 0$ for any $l \neq k$. In order for Φ_j to satisfy $\Phi_k^\dagger \Phi_j = 0\ \forall k < j$, it must be

$$\Phi_{1j} = \cdots = \Phi_{j-1,j} = 0.$$

Moreover, as Φ is upper triangular, it must be $\Phi_{j+1,j} = \cdots = \Phi_{n,j} = 0$. Therefore, by othonormality of Φ, it has to be $|\Phi_{jj}| = 1$. ∎

When A is singular, on the other hand, this is not true, as the following example shows.

Example 1. Consider the following matrix:

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

Since it is already upper triangular, a valid QR decomposition is given by $Q = I$, $R = M$. On the other hand, consider

$$Q = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \; R = \begin{bmatrix} 0 & \sqrt{2} \\ 0 & 0 \end{bmatrix},$$

which clearly also give $QR = M$, with $Q^\dagger Q = I$.

However, introducing some conditions on the R matrix, it is possible to obtain a *canonical form* for the QR decomposition in a sense that will be explained later in this section. The following lemma suggests that the columns of Q might be chosen e.g. in a way that its first r columns would span the whole range of A, r being the rank of A.

Lemma 6.0.15. *Consider a QR decomposition of a square matrix A of dimension n, and an index $\bar{j}$ in $[1,n]$, such that*

$$r_{ij} = 0 \quad \forall j \leq \bar{j}, \forall i > \rho_j \tag{6.1}$$

where ρ_j is the rank of the first j columns of A. Let a_i and q_i, be the i-th column of A and Q respectively. Then

$$< a_1, \dots, a_j > = < q_1, \dots, q_{\rho_j} > \quad \forall j = 1, \dots, \bar{j}.$$

Proof. Consider the expression for the j-th column of A, $a_j = Q r_j$. By the hypothesis, the last $n - \rho_j$ elements of r_j are zeros, hence it results

$$a_j \in < q_1, \dots, q_{\rho_j} > \quad \forall j = 1, \dots, \bar{j}$$

and therefore

$$< a_1, \dots, a_j > \subseteq < q_1, \dots, q_{\rho_j} > \quad \forall j = 1, \dots, \bar{j}.$$

As the rank of the first j columns is ρ_j, which is also the dimension of $< q_1, \dots, q_{\rho_j} >$, equality of the two subspaces holds. ∎

A QR decomposition will now be constructed via a Gram-Schmidt orthonormalization process, fixing the degrees of freedom of the upper-triangular factor R and verifying that the resulting decomposition satifies the hypotesis of Lemma 6.0.15 for $\bar{j} = n$. This particular choice of the QR decomposition gives a canonical form on $\mathbb{C}^{n \times n}$ with respect to left-multiplication for elements of the unitary matrix group $\mathcal{U}(n)$.

Construction of the QR decomposition by orthonormalization

Theorem 6.0.16. *Given any (complex) square matrix A of dimension n, it is possible to derive a QR decomposition $A = QR$ such that hypotheses of Lemma 6.0.15 are satisfied for $\bar{j} = n$, and such that the first nonzero element of each row of R is real and positive.*

Proof. The QR decomposition of A will be explicitly constructed column by column. Denote by A, Q, R the matrices, with a_i, q_i, r_i their i-th columns and with $a_{i,j}, q_{i,j}, r_{i,j}$ their elements,

respectively. Start from the first non zero column of $A \in \mathbb{C}^{n\times n}$, a_{i_0}, and define

$$q_1 = \frac{a_{i_0}}{\|a_{i_0}\|}, \qquad r_{1,i_0} = \|a_{i_0}\|, \qquad r_{2,i_0} = \ldots = r_{n,i_0} = 0. \tag{6.2}$$

Also fix $r_j = 0$ for all $j < i_0$.

The next columns of Q, R are constructed by an iterative procedure. Define ρ_{i-1} as the rank of the first $i-1$ columns of A. It can be assumed (by induction) to have the first ρ_{i-1} columns of Q and the first $i-1$ columns of R constructed in such a way that $r_{k,j} = 0$ for $k > \rho_j$ and $j \leq i-1$.

Consider the next column of A, a_i. Assume as a first case that a_i is linearly dependent with the previous columns of A, that is $\rho_i = \rho_{i-1}$. Since Lemma 6.0.15 applies, a_i can be written as

$$a_i = \sum_{j=1}^{i-1} \alpha_j a_j = \sum_{j=1}^{i-1} \alpha_j \sum_{\ell=1}^{\rho_j} r_{\ell,j} q_\ell$$

and therefore, being a_i a linear combination of the columns $\{q_1, \ldots, q_{\rho_{i-1}}\}$, the elements of r_i are defined as

$$r_{\ell,i} = q_\ell^\dagger a_i\,, \quad \text{for } \ell = 1, \ldots, \rho_i.$$

On the other hand, if the column a_i is linearly independent from the previous columns of A, then the rank $\rho_i = \rho_{i-1} + 1$. As before, the first ρ_{i-1} coefficients of r_i must be defined as

$$r_{\ell,i} = q_\ell^\dagger a_i\,, \quad \text{for } \ell = 1, \ldots, \rho_i - 1.$$

Let us also introduce

$$\tilde{a}_i := a_i - \sum_{\ell=1}^{\rho_i} r_{\ell,i} q_\ell \neq 0 \tag{6.3}$$

and define

$$q_{\rho_i} = \frac{\tilde{a}_i}{\|\tilde{a}_i\|} \qquad r_{\rho_i,i} = \|\tilde{a}_i\|. \tag{6.4}$$

In both cases, let $r_{\ell,i} = 0$ for $\ell = \rho_i + 1, \ldots, n$. It is immediate to verify that the obtained q_{ρ_i} is orthonormal to the columns $q_1, \ldots, q_{\rho_i - 1}$, and that $a_i = Q r_{\rho_i}$.

After iterating until the last column of R is defined, the remaining columns of Q have to be chosen, so that the set $\{q_1, \ldots, q_n\}$ is an orthonormal basis for $\mathbb{C}^{n\times n}$. By construction, $A = QR$. ∎

R is a canonical form

Let $\mathcal{G}$ be a group acting on $\mathbb{C}^{n\times n}$. Let $A, B \in \mathbb{C}^{n\times n}$. If there exists a $g \in \mathcal{G}$ such that $g(A) = B$, then A and B are said to be $\mathcal{G}$-equivalent, denoted by $A \sim_{\mathcal{G}} B$.

Definition 6.0.17. A canonical form with respect to $\mathcal{G}$ is a function $\mathcal{F} : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ such that for every $A, B \in \mathbb{C}^{n\times n}$:

i. $\mathcal{F}(A) \sim_{\mathcal{G}} A$;

ii. $\mathcal{F}(A) = \mathcal{F}(B)$ if and only if $A \sim_{\mathcal{G}} B$.

Consider the unitary matrix group $\mathcal{U}(n) \subset \mathbb{C}^{n\times n}$ and consider its action on $\mathbb{C}^{n\times n}$ through left-multiplication, that is, for any $U \in \mathcal{U}(n)$, $M \in \mathbb{C}^{n\times n}$:

$$U(M) = UM.$$

The following result can thus be proved.

Theorem 6.0.18. *Define $\mathcal{F}(A) = R$, with R the upper-triangular matrices obtained by the procedure described in the proof of Theorem 6.0.16. Then $\mathcal{F}$ is a canonical form with respect to $\mathcal{U}(n)$ (and its action on $\mathbb{C}^{n\times n}$ by left multiplication).*

Proof. By construction $A = QR$, with unitary Q, so $\mathcal{F}(A) \sim_{\mathcal{U}(n)} A$. If $A, B \in \mathcal{C}^{n\times n}$ are such that $\mathcal{F}(A) = \mathcal{F}(B) = R$, thus $A = QR$ and $B = VR$ for some $Q, V \in \mathcal{U}(n)$, and hence $A = QV^{-1}B$.

On the other hand, if $A = UB$, $U \in \mathcal{U}(n)$, it has to be proved that the upper-triangular matrix in the canonical QR decompositions $A = QR^{(A)}$ and $B = VR^{(B)}$ is the same. If the first non zero column of B is b_{i_0}, then the first column nonzero column of A is, being U unitary, $a_{i_0} = U b_{i_0}$. One then finds from (6.2)

$$v_1 = \frac{U^\dagger a_{i_0}}{\|U^\dagger a_{i_0}\|} = U^\dagger q_1 \quad r^{(B)}_{1,i_0} = \|U^\dagger a_{i_0}\| = r^{(A)}_{1,i_0}. \tag{6.5}$$

Hence the first i_0 columns of $R^{(A)}$ and $R^{(B)}$ are identical. Proceed then by induction. Assume that $r^{(A)}_j = r^{(B)}_j$, $q_j = U v_j$ for $j = 1, \ldots, i-1$. If the column a_i is linearly dependent from the previous $i-1$ so it must be b_i. The elements of $r^{(A)}_i$ are defined as

$$r^{(A)}_{k,i} = q_k^\dagger a_i = q_k^\dagger U U^\dagger a_i = v_k^\dagger b_i = r^{(B)}_{k,i},$$

for $k = 1, \ldots, \rho_i - 1$. On the other hand, if the column a_i is linearly independent from the previous columns of A, then the rank $\rho_i = \rho_{i-1} + 1$. As before, the first $\rho_i - 1$ coefficients of

r_i are defined as

$$r_{k,i}^{(A)} = q_k^\dagger a_i = q_k^\dagger U U^\dagger a_i = v_k^\dagger b_i = r_{k,i}^{(B)},$$

for $k = 1, \ldots, \rho_i - 1$, and $r_{k,i}^{(A)} = r_{k,i}^{(B)} = 0$ for $k = \rho_i + 1, \ldots, n$. Consider as before

$$\tilde{a}_i := a_i - \sum_{k=1}^{\rho_i - 1} r_{ki}^{(A)} q_k \neq 0.$$

By using the equivalent definition and the inductive hypothesis it follows that $\tilde{b}_i = U^\dagger \tilde{a}_i$ and

$$v_{\rho_i} = \frac{U^\dagger \tilde{a}_i}{\|U^\dagger \tilde{a}_i\|} = U^\dagger q_{\rho_i} \quad r_{\rho_i,i}^{(B)} = \|U^\dagger \tilde{a}_i\| = r_{\rho_i,i}^{(A)}.$$

Hence $r_i^{(A)} = r_i^{(B)}$, and by induction $R^{(A)} = R^{(B)}$.

■

7

Engineering attractivity via closed-loop control

This chapter deals with the problem of stabilization of a given quantum subspace by discrete-time measurements and unitary control. The employed control scheme follows the ideas of Lloyd and Viola (2001); Ticozzi and Viola (2006), and is in fact an instance of the Markovian feedback models studied in e.g. Belavkin (1983); James (2004). Suppose that a generalized measurement operation can be performed on the system at times $t = 1, 2, \ldots$, resulting in an open system, discrete-time dynamics described by a given Kraus map, with associated Kraus operators $\{M_k\}$. This can be realized, for example, when the system is coupled to an auxiliary measurement apparatus, it is manipulated coherently, and then a projective measurement is performed on the auxiliary system (see Section 5.1). Suppose moreover that the state of the system can be unitarily controlled, i.e. $\rho_{\text{controlled}} = U\rho U^\dagger$, $U \in \mathcal{U}(\mathcal{H}_I)$. Assume that the control is fast with respect to the measurement time scale, or the measurement and the control acts in distinct time slots.

The generalized measurement outcome k can then be used to condition the control choice, that is, a certain coherent transformation U_k is applied after the k-th output is recorded. In other words, a Markovian feedback control can be implemented, consisting in a map from the set of measurement outcomes to the set of unitary matrices, $U(k) : k \mapsto U_k \in \mathcal{U}(\mathcal{H}_I)$. The measurement-control loop is then iterated: by averaging over the measurement results at each step, a different TPCP map results, which depends on the design of the set of unitary controls $\{U_k\}$ and describes the evolution of the state *immediately after* each application of

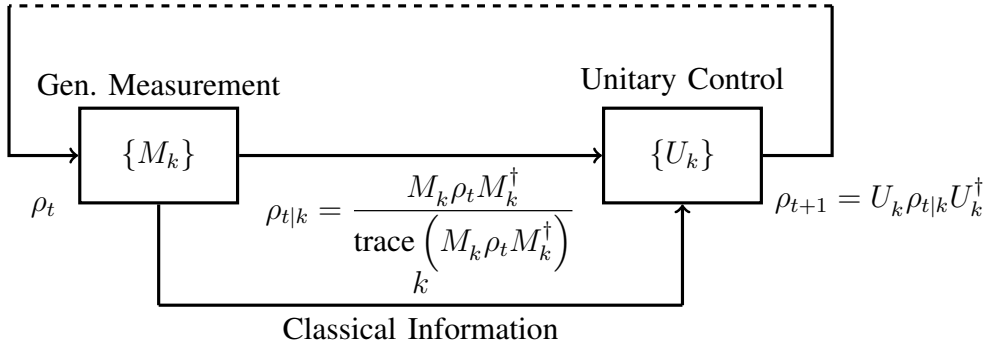


Figure 7.1: A measurement-dependent unitary control scheme.

the controls:

$$\rho(t+1) = \sum_k U_k M_k \rho(t) M_k^\dagger U_k^\dagger.$$

Figure 7.1 depicts the feedback control loop (before the averaging).

In the next section it will be considered the problem of characterizing the set of open loop dynamics that can be engineered through this feedback setup *by designing the set* $\{U_k\}$ *with fixed measurement operator* $\{M_k\}$.

Then, on the basis of the analysis results of Chapter 5, it will be derived an algorithm that allows to design the set of unitary controls such that global asymptotical stability of a *given* subspace is achieved. Notice that the desired result is going to be achieved for the averaged time-evolution that describes the system immediately after the control step. It is easy to show that if a certain subspace is GAS for the averaged dynamics, it must be so also for the conditional ones.

7.1 Simulating generalized measurements

A first straightforward application of the canonical form derived in the previous section is to establish which quantum operation can be realized applying a given generalized measurement and feedback control. Assume that a generalized measurement can be done, with associated operators $\{M_k\}_{k=1}^m$, and consider the problem of implementing a different state transformation, or measurement, with associated operators $\{N_k\}_{k=1}^m$, by using the unitary control loop as above. Notice that the employed control scheme allows to modify only the conditioned states, not the probability of the outcomes, since $\text{Tr}(M_k^\dagger M_k \rho) = \text{Tr}(M_k^\dagger U_k^\dagger U_k M_k \rho)$. The following result holds.

Proposition 7.1.1. *A measurement with associated operators* $\{N_k\}_{k=1}^m$ *can be simulated by a certain choice of unitary controls from a measurement* $\{M_k\}_{k=1}^m$*, if and only if there exist a*

reordering $j(k)$ of the first m integers such that:

$$\mathcal{F}(N_k) = \mathcal{F}(M_{j(k)}),$$

where $\mathcal{F}$ returns the canonical R factor of the argument, as described in Chapter 6.

Proof. Assume that for a given reordering $j(k)$ it holds $\mathcal{F}(N_k) = \mathcal{F}(M_{j(k)}) = R_k$. Therefore the canonical QR decomposition of N_k and $M_{j(k)}$ gives

$$N_k = U_k R_k\,, \qquad M_{j(k)} = V_{j(k)} R_k.$$

Let then $U_k V_{j(k)}^\dagger$ be the unitary control associated with the measurement outcome k. Then

$$\mathcal{T}_{\text{closed loop}}[\rho] = \sum_k U_k V_{j(k)}^\dagger M_{j(k)} \rho M_{j(k)}^\dagger V_{j(k)} U_k^\dagger = \sum_k U_k R_k \rho R_k^\dagger U_k^\dagger = \sum_k N_k \rho N_k^\dagger$$

and therefore it simulates the measurement associated with the operators $\{N_k\}_{k=1}^m$.

On the other hand, suppose that there exists a set of unitary controls $\{Q_k\}_{k=1}^m$ and there is a reordering $j(k)$ of the first m integers such that

$$Q_{j(k)} M_{j(k)} = N_k.$$

According to Theorem 6.0.18, $\mathcal{F}$ is a canonical form with respect to $\mathcal{U}(n)$ and its action on $\mathbb{C}^{n\times n}$ by left multiplication, and therefore if $Q_{j(k)} M_{j(k)} = N_k$ then $\mathcal{F}(M_{j(k)}) = \mathcal{F}(N_k)$. ∎

7.2 Global asymptotic stabilization of a quantum subspace

Suppose that the operators $\{M_k\}$ are given, corresponding to a measurement that is performed on the quantum system, with corresponding outcomes $\{k\}$. Consider the problem of finding a set of unitary transformations $\{U_k\}$ such that, once they are applied to the system, the resulting semigroup generator

$$\mathcal{T}[\rho] = \sum_k U_k M_k \rho M_k^\dagger U_k^\dagger$$

makes a given set $\mathcal{J}_S(\mathcal{H}_I)$ GAS.

The following preliminary, technical result is needed.

Lemma 7.2.1. *Let R be the upper triangular factor of a* canonical QR decomposition *in the*

form

$$R = \begin{bmatrix} R_S & R_P \\ 0 & R_R \end{bmatrix}$$

(according to the block structure induced by (5.4)) and suppose $R_P = 0$. Consider the matrix N obtained by left multiplying R by a unitary matrix V:

$$N = VR = \begin{bmatrix} V_S & V_P \\ V_Q & V_R \end{bmatrix} \begin{bmatrix} R_S & 0 \\ 0 & R_R \end{bmatrix} = \begin{bmatrix} N_S & N_P \\ N_Q & N_R \end{bmatrix}.$$

Then $N_Q = 0$ implies $N_P = 0$.

Proof. Consider first the case in which R_S is full rank. Let $r \times m$ be the dimension of V_Q, and $m \times m$ be the dimension of R_S. Since it must be $N_Q = V_Q R_S = 0$ and R_S is full rank, it must be $V_Q = 0$.

As V is unitary, its column must be orthonormal. Being $V_Q = 0$, V_S must be itself an orthonormal block in order to have orthonormality of the first m columns of V. It then follows that $V_P = 0$, because any j-th column, $j > m$, must be orthonormal to all the first m columns. It then follows that

$$N_P = V_P R_R = 0.$$

Consider now the other case in which R_S is singular. This implies that $\rho_m < m$ (ρ_m being the rank of the first m columns of R); therefore, as R is a triangular factor of a canonical QR decomposition, the element $R_{\rho_m+1,m} = 0$.

Now, by construction of the canonical QR decomposition, if there were non-zero columns of index $j > m$, one of them would have a non-zero element on the row of index $\rho_m + 1$. By recalling that $R_P = 0$, it results $R_{\rho_m+1,j} = 0$, $\forall j \in [m+1, m+r]$. Therefore, all the last r columns are zero-vectors, and in particular $R_R = 0$. It then follows that

$$N_P = V_P R_R = 0.$$

■

This result will be instrumental in proving the main theorem of the section, which provides an iterative control design procedure that renders the desired subspace asymptotically stable whenever it is possible.

Theorem 7.2.2. *Consider a subspace orthogonal decomposition $\mathcal{H}_I = \mathcal{H}_S \oplus \mathcal{H}_R$ and a given generalized measurement associated to Kraus operators $\{M_k\}$.*

The task of achieving global asymptotic stability of $\mathcal{J}_S(\mathcal{H}_I)$ by a feedback unitary control

policy is feasible if and only if the following algorithm is completed successfully. In this case, the algorithm returns a control law $\{U_k\}$ that makes $\mathcal{J}_S(\mathcal{H}_I)$ GAS for the controlled dynamics.

Control design algorithm

Let $\{|\phi\rangle_j^S\}_{j=1}^m$, $\{|\phi\rangle_k^R\}_{k=1}^r$ denote orthonormal bases for $\mathcal{H}_S$ and $\mathcal{H}_R$, and represent each M_k as a matrix with respect to the basis $\{|\phi\rangle_j^S\}_{j=1}^m \bigcup \{|\phi\rangle_k^R\}_{k=1}^r$. Compute a QR decomposition $M_k = Q_k R_k$ with canonical R_k for each k. Call $\mathcal{H}_R^{(0)} = \mathcal{H}_R$, $\mathcal{H}_S^{(0)} = \mathcal{H}_S$, $U_k^{(0)} = Q_k^\dagger$ and rename the matrix blocks $R_{S,k}^{(0)} = R_{S,k}$, $R_{P,k}^{(0)} = R_{P,k}$ and $R_{R,k}^{(0)} = R_{R,k}$.

If $R_{P,k} = 0\ \forall k$, then the problem is not feasible and a unitary control law cannot be found. Otherwise define $V^{(0)} = I$, $Z^{(0)} = I$, and consider the following iterative procedure, starting from $i = 0$:

1. Define $\mathcal{H}_R^{(i+1)} = \bigcap_k \ker R_{P,k}^{(i)}$:
   If $\mathcal{H}_R^{(i+1)} = \{0\}$ then the iteration is successfully completed. Go to step 8).
   If $\mathcal{H}_R^{(i+1)} \subsetneq \mathcal{H}_R^{(i)}$, define $\mathcal{H}_S^{(i+1)} = \mathcal{H}_R^{(i)} \ominus \mathcal{H}_R^{(i+1)}$ and $Y^{(i+1)} = I$.
   If $\mathcal{H}_R^{(i+1)} = \mathcal{H}_R^{(i)}$ (i.e. $R_{P,k}^{(i)} = 0\ \forall k$) then, if $\dim(\mathcal{H}_R^{(i)}) \geq \dim(\mathcal{H}_S^{(i)})$:

   (a) Choose a subspace $\mathcal{H}_S^{(i+1)} \subseteq \mathcal{H}_R^{(i)}$ of the same dimension of $\mathcal{H}_S^{(i)}$. (Re)-define $\mathcal{H}_R^{(i+1)} = \mathcal{H}_R^{(i)} \ominus \mathcal{H}_S^{(i+1)}$.
   (b) Let $\mathcal{H}_T^{(i)} = \bigoplus_{j=0}^{i-1} \mathcal{H}_S^{(j)}$. Construct a unitary matrix Y with the following block form, according to a Hilbert space decomposition $\mathcal{H}_I = \mathcal{H}_T^{(i)} \oplus \mathcal{H}_S^{(i)} \oplus \mathcal{H}_S^{(i+1)} \oplus \mathcal{H}_R^{(i+1)}$:

$$Y^{(i+1)} = \left[\begin{array}{c|c|c|c} I & 0 & 0 & 0 \\ \hline 0 & 1/\sqrt{2}I & 1/\sqrt{2}I & 0 \\ \hline 0 & 1/\sqrt{2}I & -1/\sqrt{2}I & 0 \\ \hline 0 & 0 & 0 & I \end{array}\right]. \tag{7.1}$$

   If instead $\dim(\mathcal{H}_R^{(i)}) < \dim(\mathcal{H}_S^{(i)})$:

   (a) Choose a subspace $\mathcal{H}_S^{(i+1)} \subseteq \mathcal{H}_S^{(i)}$ of the same dimension of $\mathcal{H}_R^{(i)}$.
   (b) Let $\mathcal{H}_T^{(i)} = \left(\bigoplus_{j=0}^{i-1} \mathcal{H}_S^{(j)}\right) \oplus \left(\mathcal{H}_S^{(i)} \ominus \mathcal{H}_S^{(i+1)}\right)$. Construct a unitary matrix Y with the following block form, according to a Hilbert space decomposition $\mathcal{H}_I = \mathcal{H}_T^{(i)} \oplus \mathcal{H}_S^{(i+1)} \oplus \oplus \mathcal{H}_R^{(i+1)}$:

$$Y^{(i+1)} = \left[\begin{array}{c|c|c} I & 0 & 0 \\ \hline 0 & 1/\sqrt{2}I & 1/\sqrt{2}I \\ \hline 0 & 1/\sqrt{2}I & -1/\sqrt{2}I \end{array}\right]. \tag{7.2}$$

(c) Define $Z^{(i+1)} = Z^{(i)}Y^{(i+1)}$ and go to step 8).

2. Define $Z^{(i+1)} = Z^{(i)}Y^{(i+1)}$.

3. Rewrite $\tilde{R}^{(i)}_{R,k} = W^{(i+1)}R^{(i)}_{R,k}W^{(i+1)\dagger}$ in a basis according to the $\mathcal{H}^{(i)}_R = \mathcal{H}^{(i+1)}_S \oplus \mathcal{H}^{(i+1)}_R$ decomposition.

4. Compute the canonical QR decomposition of $\tilde{R}^{(i)}_{R,k} = Q^{(i+1)}_k R^{(i+1)}_k$. Compute the matrix blocks $R^{(i+1)}_{P,k}, R^{(i+1)}_{R,k}$ of $R^{(i+1)}_k$, again according to the decomposition $\mathcal{H}^{(i)}_R = \mathcal{H}^{(i+1)}_S \oplus \mathcal{H}^{(i+1)}_R$.

5. Define $U^{(i+1)} = \left[\begin{array}{c|c} I & 0 \\ \hline 0 & W^{(i+1)\dagger}\left(Q^{(i+1)}_k\right)^{\dagger} W^{(i+1)} \end{array}\right] U^{(i)}$.

6. Define $V^{(i+1)} = \left[\begin{array}{c|c} I & 0 \\ \hline 0 & W^{(i+1)} \end{array}\right] V^{(i)}$.

7. Increment the counter and go back to step 1).

8. Return the unitary controls $U_k = V^{(i)\dagger}Z^{(i)}V^{(i)}U^{(i)}_k$.

Proof. Consider first the case in which the algorithm stops before the iterations. This happens if $R_{P,k} = 0$ for every k. Remember that each R_k has been put in canonical form, so it follows from Lemma 7.2.1 that any control choice that ensures invariance of the desired subspace, that is $N_k = U_k R_k$ with $N_{Q,k} = 0$, makes also $\mathcal{J}_R(\mathcal{H}_I)$ invariant, since $N_{P,k} = 0$. Hence an invariant state with support on $\mathcal{H}_R$ always exists. This, via Theorem 5.3.12, precludes the existence of a control choice that renders $\mathcal{J}_S(\mathcal{H}_I)$ GAS.

If the algorithm does not stop, then at each step of the iteration the dimension of $\mathcal{H}^{(i)}_R$ is reduced by at at least 1, hence the algorithm is completed in at most n steps. If the algorithm is successfully completed at a certain iteration j, a set of unitary controls $\{U^{(j)}_k\}$ and a unitary $V^{(j)}$ have been built, such that the controlled quantum operation element, under the change

of basis $V^{(j)}$, is of the form:

$$\tilde{N}_k = V^{(j)} U_k M_k V^{(j)\dagger} = Z^{(j)} \left[\begin{array}{c|c|c|c|c} R^{(0)}_{S,k} & \bar{R}^{(0)}_{P,k} & 0 & 0 & 0 \\ \hline 0 & R^{(1)}_{S,k} & \ddots & 0 & 0 \\ \hline 0 & 0 & \ddots & \bar{R}^{(j-1)}_{P,k} & 0 \\ \hline 0 & 0 & 0 & R^{(j)}_{S,k} & \bar{R}^{(j)}_{P,k} \\ \hline 0 & 0 & 0 & 0 & R^{(j)}_{R,k} \end{array}\right] \tag{7.3}$$

where the block structure is consistent with the decomposition $\bigoplus_{i=0}^{j+1} \mathcal{H}_S^{(i)}$ (where to simplify the notation it has been set $\mathcal{H}_S^{(j+1)} = \mathcal{H}_R^{(j)}$). Let $\bar{R}_k$ be the block matrix above and consider its upper-triangular part. The rows have the form $\left[\bar{R}^{(i)}_{P,k} \quad 0 \quad \dots \quad 0\right]$ because at each step of the iteration a basis $W^{(i)}$ is chosen according to the decomposition $\mathcal{H}_S^{(i+1)} \oplus \mathcal{H}_R^{(i+1)}$, where $\mathcal{H}_R^{(i+1)} \subseteq \bigcap_k \ker R^{(i)}_{P,k}$, hence obtaining $R^{(i)}_{P,k} W^{(i)\dagger} = \left[\bar{R}^{(i)}_{P,k} \quad 0 \quad \dots \quad 0\right]$. It is easy to verify that the subsequent unitary transformations have no effects on the blocks $\bar{R}^{(i)}_{P,k}$.

The upper-triangular form of each $\bar{R}_k$ and the form of $Z^{(j)}$ and $V^{(j)}$, both block-diagonal with respect to the orthogonal decomposition $\mathcal{H}_S \oplus \mathcal{H}_R$, ensure invariance of $\mathcal{H}_S$.

By construction, for all $i = 0, \dots, j$, either $\bigcap_k \ker \bar{R}^{(i)}_{P,k} = \{0\}$ and $Y^{(i)} = I$, or $\bar{R}^{(i)}_{P,k} = 0$ for all k and $Y^{(i)}$ differs from the identity matrix and has the form (7.1) or (7.2).

Let us prove that no invariant state can have support on $\bigoplus_{i=1}^{j+1} \mathcal{H}_S^{(i)}$ by induction. First consider a state with support on $\mathcal{H}_S^{(j+1)} = \mathcal{H}_R^{(j)}$ alone:

$$\bar{\rho} = \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & \bar{\rho}_R \end{array}\right].$$

If $\bigcap_k \ker \bar{R}^{(j)}_{P,k} = \{0\}$, then $\bar{\rho}$ is mapped by $\sum_k \bar{R}_k \cdot \bar{R}_k^\dagger$ into a state $\bar{\rho}'$ with non-trivial support on $\mathcal{H}_S^{(j)}$. Being in this case $Y^{(j)} = I$, $Z^{(j)}$ is block-diagonal with respect to the considered decomposition and it cannot be $Z^{(j)} \bar{\rho}' Z^{(j)\dagger} = \bar{\rho}$, for any $\bar{\rho}$ in $\mathfrak{D}(\mathcal{H}_R^{(j)})$.

On the other hand, if $\bar{R}^{(j)}_{P,k} = 0 \ \forall k$, then $Y^{(j)}$ contains off-diagonal full-rank blocks and maps the state

$$\bar{\rho}' = \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & \sum_k R^{(j)}_{R,k} \bar{\rho}_R R^{(j)\dagger}_{R,k} \end{array}\right]$$

into a state with non-trivial support on $\mathcal{H}_S^{(j)}$. The subsequent application of $Z^{(j-1)}$ will then map the state into a state with nontrivial support on $\bigoplus_{i=1}^{j} \mathcal{H}_S^{(i)}$, and therefore $\bar{\rho}$ cannot be invariant.

Let us now proceed with the inductive step, with m as the induction index. Assume that

no invariant state can have support on $\bigoplus_{i=j+1-m}^{j+1} \mathcal{H}_S^{(i)}$ alone (induction hypothesis), and consider the subspace $\bigoplus_{i=j-m}^{j+1} \mathcal{H}_S^{(i)}$. By the induction hypothesis if there were an invariant state with support on this subspace, it would be in the form

$$\bar{\rho} = \left[\begin{array}{c|c|c} 0 & 0 & 0 \\ \hline 0 & \bar{\rho}_S & \bar{\rho}_P \\ \hline 0 & \bar{\rho}_P^\dagger & \bar{\rho}_R \end{array}\right]$$

with $\bar{\rho}_S \neq 0$ having support on $\mathcal{H}_S^{(j-m)}$. Let us rewrite

$$Z^{(j)} = Z^{(j-m-1)} Y^{(j-m)} Z_{(j-m+1)}$$

where $Z_{(j-m+1)} = Y^{(j-m+1)} \cdot \ldots \cdot Y^{(j)}$.

Again, two cases may happen. If $\bigcap_k \ker \bar{R}_{P,k}^{(j-m-1)} = \{0\}$, then $Y^{(j-m)} = I$ and $\bar{\rho}$ is mapped by $\bar{R}_k$ into a state with non trivial support on $\mathcal{H}_S^{(j-m-1)}$. The subsequent application of $Z_{(j-m+1)}$ and of $Y^{(j-m)}$ does not affect this, and because of $Z^{(j-m-1)}$, the first complete iteration will map $\bar{\rho}$ into a state with non trivial support on $\bigoplus_{i=1}^{j-m-1} \mathcal{H}_S^{(i)}$. Therefore $\bar{\rho}$ cannot be invariant.

On the other hand, if $\bar{R}_{P,k}^{(j-m-1)} = 0 \ \forall k$, then $Y^{(j-m)}$ has the form (7.1) and the closed loop evolution of $\bar{\rho}$ is

$$\bar{\rho}' = \sum_k \Big(Z^{(j-m-1)} Y^{(j-m)} \underbrace{Z_{(j-m+1)} \bar{R}_k \bar{\rho} \bar{R}_k^\dagger Z_{(j-m+1)}^\dagger}_{\tilde{\rho}_k} \cdot Y^{(j-m)\dagger} Z^{(j-m-1)\dagger} \Big).$$

If $\tilde{\rho}_k$ has support on $\bigoplus_{i=j+1-m}^{j+1} \mathcal{H}_S^{(i)}$ for all k, then $\bar{\rho}'$ will have the same support, and therefore $\bar{\rho}$ is not invariant. If instead $\tilde{\rho}_k$ has non trivial support on $\mathcal{H}_S^{(j-m)}$ for some k, then because of the subsequent application of $Z^{(j-m-1)} Y^{(j-m)}$, $\bar{\rho}'$ will have non trivial support on $\bigoplus_{i=1}^{j-m-1} \mathcal{H}_S^{(i)}$, and again $\bar{\rho}$ is not invariant.

When the induction process reaches $m = j-1$, then it states that no invariant states are supported on $\mathcal{H}_S^{(1)} \oplus \cdots \oplus \mathcal{H}_S^{(j)} \oplus \mathcal{H}_R^{(j)}$, and therefore according to Theorem 5.3.12 global asymptotic stability of the subspace $\mathcal{S}$ is achieved. ■

The algorithm is clearly constructive. The following corollary holds.

Corollary 7.2.3. *A desired subspace $\mathcal{H}_S$ can be made GAS if and only if the $R_{P,k}$ blocks of the canonical R-factors, computed with respect to the decomposition $\mathcal{H}_I = \mathcal{H}_S \oplus \mathcal{H}_R$, are not all zero.*

Proof. The control design algorithm described in Theorem 7.2.2 fails to run to completion if and only if $R_{P,k} = 0 \; \forall k$. According to the same theorem, this is therefore a sufficient and necessary condition for feasibility of global asymptotic stabilization of the subspace $\mathcal{H}_S$. ■

It has been noticed in Bolognani and Ticozzi (2010b) that from Corollary 7.2.3 it follows that a pure state ρ_d can be rendered GAS for some feedback control strategy if and only if there exists a $\bar{k}$ such that:

$$[\rho_d, R_{\bar{k}}] \neq 0.$$

While this condition does resemble the one emerging from the study of the Markovian feedback master equation in continuous-time Ticozzi and Viola (2009), where the R_k's are substituted by the Hermitian part of the measurement operator M, a remarkable difference is apparent: the structure of the R_k's also depends on the choice of target state, rendering the determination of the stabilizable pure-state manifold non trivial.

7.3 Robustness of state-preparation

A potential limitation to the implementation of this feedback strategy lays in the fact it requires strong control capabilities and perfect detection. That is, it is assumed that the form of the measurement map is known exactly, and that every measurement leads to a valid outcome.

In Ticozzi and Bolognani (2010) it has been evaluated how critical this hypothesis is for the whole procedure following the approach of Ticozzi and Viola (2009). Let us choose a suitable Hermitian basis in $\mathfrak{B}(\mathcal{H}_i) \approx \mathbb{C}^{d\times d}$. This can always be done for finite d, for example by employing the natural d-dimensional extension of the Pauli matrices (Alicki and Lendi, 1987; Altafini, 2004). In such a basis, all density operators are represented by d^2-dimensional vectors $\bar{\rho} = (\rho_0, \rho_1, \ldots, \rho_{d^2-1})^T$, where the first component ρ_0, relative to $\frac{1}{\sqrt{d}}\mathbb{I}_d$, is invariant and equal to $\frac{1}{\sqrt{d}}$ for TP-dynamics. Let $\rho_v = (\rho_1, \ldots, \rho_{d^2-1})^T$. Hence any Kraus map $\mathcal{E}[\cdot]$, being a TP linear map, in this vectorized representation must take the form:

$$\bar{\rho}(t+1) = \begin{bmatrix} 1/\sqrt{d} \\ \rho_v(t+1) \end{bmatrix} = \left[\begin{array}{c|c} 1 & 0 \\ \hline C & D \end{array}\right] \begin{bmatrix} 1/\sqrt{d} \\ \rho_v(t) \end{bmatrix}.$$

Assume that the dynamics has a unique attractive state $\bar{\rho}^{(0)}$. Thus $I - D$ must be invertible and then:

$$\bar{\rho}^{(0)} = \frac{1}{\sqrt{d}} \begin{bmatrix} 1 \\ (I-D)^{-1}C \end{bmatrix}.$$

Consider now a small perturbation of the Kraus map, $\tilde{\mathcal{E}}[\cdot] = (1-\varepsilon)\mathcal{E}[\cdot]+\varepsilon\mathcal{E}'[\cdot]$ depending on the continuous parameter ε, and ε sufficiently small so that $(I - D - \varepsilon(D' - D))$ remains invertible. This may account for small detection errors, imperfect knowledge of the model and other non-idealities. The vectorized dynamics becomes:

$$\bar{\rho}(t+1) = \left((1-\varepsilon) \left[\begin{array}{c|c} 1 & 0 \\ \hline C & D \end{array}\right] + \varepsilon \left[\begin{array}{c|c} 1 & 0 \\ \hline C' & D' \end{array}\right] \right) \begin{bmatrix} 1/\sqrt{d} \\ \rho_v \end{bmatrix},$$

and the new attractive, unique equilibrium state is:

$$\bar{\rho}^{(\varepsilon)} = \frac{1}{\sqrt{d}} \begin{pmatrix} 1 \\ (I - (1-\varepsilon)D - \varepsilon D')^{-1}((1-\varepsilon)C + \varepsilon C') \end{pmatrix}.$$

Because $\bar{\rho}^{(\varepsilon)}$ is a continuous function of ε, this guarantees that for a sufficiently high detection efficiency the perturbed attractive state will be arbitrarily close to the desired one in trace norm. Therefore, if the control task is relaxed to a state preparation problem with sufficiently high fidelity, this may be accomplished with a sufficiently high detection efficiency, yet strictly less than 1.

7.4 Examples

In this section the capabilities of the proposed feedback unitary control are studied, and when possible the proposed algorithm is applied to design an effective control law.

Projective measurements

A particularly simple case is worth mentioning: when the M_k are rank one projectors, that is, *represent a non-degenerate von Neumann's measurement*, the stabilization of *any pure state* can be achieved. In fact, being a canonical form:

$$\mathcal{F}(M_k) = \mathcal{F}(U\Pi_k U^\dagger) = \mathcal{F}(\Pi_k U^\dagger) = R_k,$$

where Π_i is the rank one projector on the k-th basis element, and hence $\Pi_k U^\dagger$ is different from zero only in the k-th row, which is in turn the k-th column of U, u_k. Thus each R_k has only the first row different from zero, and it is proportional to $u_k^\dagger$. Being $\{u_k\}$ a basis, some R_{Pk} has to be non-zero as it corresponds to the last $n-1$ components of the u_k's.

Physically, at any measurement step a known pure state is obtained, which can then be driven back to desired one. While the achieved "cyclic" stabilization may appear weak, the

use of projective measurements renders it robust with respect unwanted noise effects: at each cycle a state of maximal information is deterministically determined by the measurement, virtually erasing any unwanted dynamics.

Entanglement Generation

Consider *two-qubit system*, defined on a Hilbert space $\mathcal{H}_I \simeq \mathbb{C}^2 \otimes \mathbb{C}^2$. Consider the task of stabilizing the maximally entangled state

$$\rho_d = \frac{1}{2}(|00\rangle + |11\rangle)(\langle 00| + \langle 11|), \tag{7.4}$$

which has the representation $\rho_d = \frac{1}{2}\begin{bmatrix} 1&0&0&1\\0&0&0&0\\0&0&0&0\\1&0&0&1\end{bmatrix}$ in the *computational basis* $\mathcal{C} = \{|ab\rangle = |a\rangle \otimes |b\rangle | a, b = 0, 1\}$.

In order to apply the proposed control design technique, let us consider a different basis $\mathcal{B}$ such that in the new representation $\rho_d^{\mathcal{B}} = \mathrm{diag}([\,1\;0\;0\;0\,])$. This can be achieved by considering the *Bell-basis*

$$\mathcal{B} = \left\{ \frac{|00\rangle + |11\rangle}{\sqrt{2}}, \frac{|00\rangle - |11\rangle}{\sqrt{2}}, \frac{|01\rangle + |10\rangle}{\sqrt{2}}, \frac{|01\rangle - |10\rangle}{\sqrt{2}} \right\}.$$

Let B be the unitary matrix realizing the change of basis, i.e. $\rho_d^{\mathcal{B}} = B^\dagger \rho_d B$. Consider the space decomposition

$$\mathcal{H}_I = \mathcal{H}_S \oplus \mathcal{H}_R$$

where $\mathcal{H}_S = \mathrm{span}\left\{\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)\right\}$ and $\mathcal{H}_R = \mathcal{H}_S^\perp$. The problem of stabilizing the maximally entangled state (7.4) has then been successfully casted into the problem of achieving asymptotic stability of the subspace $\mathcal{H}_S$.

Suppose that the following generalized measurement is available

$$\mathcal{T}[\rho] = \sum_{k=1}^{3} M_k \rho M_k^\dagger$$

with operators (represented in the computational basis):

$$M_1 = \frac{1}{\sqrt{4}}(\sigma_+ \otimes I), \quad M_2 = \frac{1}{\sqrt{4}}(I \otimes \sigma_+),$$
$$M_3 = \sqrt{I - M_1^\dagger M_1 - M_2^\dagger M_2},$$

where $\sigma_+ = \left[\begin{smallmatrix} 0&1\\0&0\end{smallmatrix}\right]$. These Kraus operators may be used to describe a discrete-time sponta-

neous emission process, where the event associated to $M_{1,2}$ corresponds to the decay of one qubit (with probability $\frac{1}{4}$ each), and the event of the two qubits decaying in the same time interval is neglected.

In the Bell basis, the operators take the form

$$M_1^B = \begin{bmatrix} 0 & 0 & \frac{1}{4} & -\frac{1}{4} \\ 0 & 0 & \frac{1}{4} & -\frac{1}{4} \\ \frac{1}{4} & -\frac{1}{4} & 0 & 0 \\ \frac{1}{4} & -\frac{1}{4} & 0 & 0 \end{bmatrix}, \qquad M_2^B = \begin{bmatrix} 0 & 0 & \frac{1}{4} & \frac{1}{4} \\ 0 & 0 & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & -\frac{1}{4} & 0 & 0 \\ -\frac{1}{4} & \frac{1}{4} & 0 & 0 \end{bmatrix},$$

$$M_3^B = \begin{bmatrix} 0.8536 & 0.1464 & 0 & 0 \\ 0.1464 & 0.8536 & 0 & 0 \\ 0 & 0 & 0.8660 & 0 \\ 0 & 0 & 0 & 0.8660 \end{bmatrix}.$$

Let us then apply the proposed algorithm. The canonical QR decomposition of the matrices M_k^B returns the following triangular factors (the corresponding orthogonal matrices Q_k are not reported here, see (7.6) for the final form of the controls):

$$R_1 = \begin{bmatrix} \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad R_2 = \begin{bmatrix} \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & 0 & 0 \\ 0 & 0 & \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$R_3 = \begin{bmatrix} 0.8660 & 0.2887 & 0 & 0 \\ 0 & 0.8165 & 0 & 0 \\ 0 & 0 & 0.8660 & 0 \\ 0 & 0 & 0 & 0.8660 \end{bmatrix}.$$

According to the proposed approach, by inspection of the upper triangular factors R_i it is possible to decide about the feasibility of the stabilization task. Indeed, as the blocks $R_{P,k}$, $k = 1, \ldots, 3$ are non-zero blocks, namely

$$\begin{aligned} R_{P,1} &= \begin{bmatrix} -\frac{\sqrt{2}}{4} & 0 & 0 \end{bmatrix}, \qquad R_{P,2} = \begin{bmatrix} -\frac{\sqrt{2}}{4} & 0 & 0 \end{bmatrix}, \\ R_{P,3} &= \begin{bmatrix} 0.2887 & 0 & 0 \end{bmatrix}, \end{aligned} \tag{7.5}$$

then the stabilization problem is feasible.

Moreover, notice that at this step no further transformation is needed on the matrices, as

the obtained R factors are already decomposed according to

$$\mathcal{H}_I = \mathcal{H}_S \oplus \mathcal{H}_S^{(1)} \oplus \mathcal{H}_R^{(1)}.$$

where $\mathcal{H}_R^{(1)} = \bigcap_k \ker R_{P,k}$. Continuing with the iteration, the subspace $\mathcal{H}_R^{(2)} = \bigcap_k \ker R_{P,k}^{(1)}$ has to be determined. By inspection one can see that this space is empty, and therefore the iteration stops successfully. The set of unitary controls that have to be applied when the corresponding outcome k is measured is then

$$U_k = BQ_k^\dagger B^\dagger,$$

that is:

$$\begin{aligned} U_1 &= \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & 0 & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & 0 & 0 & \frac{\sqrt{2}}{2} \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}, \qquad U_2 = \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & 0 & -\frac{\sqrt{2}}{2} \\ 0 & 1 & 0 & 0 \\ \frac{\sqrt{2}}{2} & 0 & 0 & \frac{\sqrt{2}}{2} \\ 0 & 0 & -1 & 0 \end{bmatrix}, \\ U_3 &= \begin{bmatrix} 0.9856 & 0 & 0 & 0.1691 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -0.1691 & 0 & 0 & 0.9856 \end{bmatrix}. \end{aligned} \tag{7.6}$$

It can be shown by direct computation that the Hamiltonians needed to implement these unitary transformation (using ideally unbounded control pulses in order to make the dissipation effect negligible on when the control is acting) form a 3-dimensional control algebra (D'Alessandro, 2007).

Figure 7.2 illustrates the typical behavior of closed loop evolution, where the initial density matrix is constructed from a finite ensemble of 10 randomly chosen pure states. The four curves represent the probability of finding the system in each one of the four Bell states. The solid line corresponds to the target state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ and it approaches 1 as the control law is iterated, confirming that $\mathcal{H}_S$ is rendered asymptotically stable by the designed feedback control. The dashed curve corresponds to the second Bell state $\frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$, i.e. to the probability of finding the system in $\mathcal{H}_S^{(1)}$, while the two dotted lines represent the probability of finding the system in the two remaining Bell states belonging to $\mathcal{H}_R^{(1)}$. Their behavior is consistent with the block structure of the closed loop operators reported in (7.5): since $\bigcap_k \ker R_{P,k}^{(1)}$ is empty, the trace of the projection of the $\rho(t)$ onto $\mathcal{H}_R^{(1)}$ must decrease monotonically as the control protocol is iterated. On the other hand, the dashed

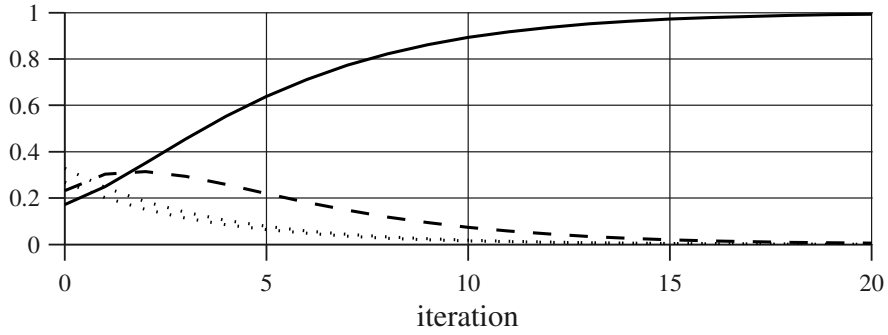


Figure 7.2: Evolution of the probability of finding the simulated system in one of the Bell's basis states, as the control algorithm iterates. The system has been initialized as an ensemble of $N = 10$ randomly picked states, i.e. $\rho(0) = \frac{1}{N}\sum_{i=1}^{N} |\psi_i\rangle\langle\psi_i|$ where $|\psi_i\rangle$'s belong to $\mathcal{H}_I$ and are randomly generated as linear combinations of the four Bell states. The solid line correspond to the probability of finding the system in the desired state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

curve exhibits an initial growth: in fact, $\mathcal{H}_S^{(1)}$ acts as a "transition" subspace which collects states from $\mathcal{H}_R^{(1)}$, and it is then attracted by $\mathcal{H}_S^{(0)}$.

A non-stabilizable case

As a third example, consider the case in which the problem of achieving global asymptotic stability of a given pure state by a feedback unitary control law is not feasible.

Consider a system of dimension $d = 2$, and consider the problem of stabilizing $\rho_0 = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right]$. Suppose that the following set of two measurements is given:

$$M_1 = \begin{bmatrix} \sqrt{p} & 0 \\ 0 & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} \sqrt{1-p} & 0 \\ 0 & 1 \end{bmatrix}.$$

Both M_1 and M_2 are already in the canonical upper triangular form prescribed by Theorem 6.0.16. Following Theorem 7.2.2, feasibility of the control problem can be checked by inspecting

$$[\rho_0, R_1] = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right]\left[\begin{smallmatrix} \sqrt{p} & 0 \\ 0 & 0 \end{smallmatrix}\right] - \left[\begin{smallmatrix} \sqrt{p} & 0 \\ 0 & 0 \end{smallmatrix}\right]\left[\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right] = \left[\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right]$$
$$[\rho_0, R_2] = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right]\left[\begin{smallmatrix} \sqrt{1-p} & 0 \\ 0 & 1 \end{smallmatrix}\right] - \left[\begin{smallmatrix} \sqrt{1-p} & 0 \\ 0 & 1 \end{smallmatrix}\right]\left[\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right] = \left[\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right].$$

As both the terms are zero, the problem is not feasible (there is no feedback unitary control that makes ρ_0 GAS).

8

Conclusions

Completely positive, trace-preserving maps represent general quantum dynamics for open systems, and if the environment is memoryless, also represent generators of discrete-time quantum Markov semigroups. Theorem 5.3.12 provides a characterization of the semigroup dynamics that render a certain pure state, or the set of states with support on a subspace, attractive, by employing LaSalle's Invariance Principle. In order to exploit this result for constructive design of stabilizing unitary feedback control strategies, a suitable linear algebraic tool has been developed, which holds some interest *per se*. It has been proved that a *canonical* QR decomposition can be derived by specializing the well-known orthonormalization approach, and that it is key to study the potential of the simple feedback control scheme presented in Chapter 7. The applicability of the derived synthesis results is twofold, since they go in the direction of both establishing the potential of discrete-time feedback strategies for pure state preparation, overcoming some intrinsic limitations that pure open-loop strategies present, and investigating the possibility of engineering invariant subspaces for quantum information encodings by feeding back information leaking into the environment. It has been determined which quantum generalized measurements, and hence non-unitary dynamics, can be simulated by unitarily controlling a given one, and whether pure states or subspaces can be rendered globally asymptotically stable. Theorem 7.2.2 gives a constructive procedure to build the unitary control, and also a test on the existence of such controls: if the algorithm does not stop on the first step, then the control problem has a solution.

References

Alicki R. and Lendi K. *Quantum Dynamical Semigroups and Applications*. Springer-Verlag, Berlin, 1987.

Altafini C. Coherent control of open quantum dynamical systems. *Physical Review A: Atomic, Molecular, and Optical Physics*, 70(6):062321:1–8, 2004.

Altafini C. Feedback stabilization of isospectral control systems on complex flag manifolds: application to quantum ensembles. *IEEE Transactions on Automatic Control*, 11(52): 2019–2028, 2007.

Bauso D., Giarré L., and Pesenti R. Nonlinear protocols for optimal distributed consensus in networks of dynamic agents. *Systems and Control Letters*, 55:918–928, 2006.

Belavkin V. P. Theory of control of observable quantum systems. *Automatica and Remote Control*, 44(2):178–188, 1983.

Bertsekas D. P. and Tsitsiklis J. N. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

Bolognani S., Carli R., and Zampieri S. A PI consensus controller with gossip communication for clock synchronization in wireless sensors networks. In *Proceedings of NECSYS'09*, Venice, Italy, 2009.

Bolognani S., Del Favero S., Schenato L., and Varagnolo D. Distributed sensor calibration and least-square parameter identification in WSNs using consensus algorithms. In *Proceedings of the 46th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, 2008.

Bolognani S., Del Favero S., Schenato L., and Varagnolo D. Consensus-based distributed sensor calibration and least-square parameter estimation in WSNs. *International Journal of Robust and Nonlinear Control*, 20(2):176–193, January 2010.

Bolognani S. and Ticozzi F. Engineering stable discrete-time quantum dynamics via a canonical QR decomposition. *IEEE Transactions on Automatic Control*, 55(12):2721–2734, December 2010a.

Bolognani S. and Ticozzi F. Pure state stabilization with discrete-time quantum feedback. In *Proceedings of the 4th International Symposium on Communications, Control and Signal Processing (ISCCSP'10)*, Limassol, Cyprus, March 2010b.

Bolognani S. and Zampieri S. Distributed quasi-Newton method and its application to the optimal reactive power flow problem. In *Proceedings of NECSYS'10*, Annecy, France, September 2010.

Bolognani S. and Zampieri S. A gossip-like distributed optimization algorithm for reactive power flow control. In *Proceedings of the IFAC World Congress 2011*, Milano, Italy, August 2011.

Bouten L., van Handel R., and James M. R. A discrete invitation to quantum filtering and feedback control. *SIAM Review*, 51(2):239–316, 2009.

Bouwmeester D., Ekert A. K., and Zeilinger A., editors. *The Physics of Quantum Information: Quantum Cryptography, Quantum Teleportation, Quantum Computation*. Springer-Verlag, 2000.

Boyd S., Ghosh A., Prabhakar B., and Shah D. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530, June 2006.

Boyd S. and Vandenberghe L. *Convex Optimization*. Cambridge University Press, 2008.

Breuer H. P. and Petruccione F. *The Theory of Open Quantum Systems*. Oxford University Press, UK, 2006.

Bullo F., Carli R., and Frasca P. Gossip coverage control for robotic networks: Dynamical systems on the space of partitions. *SIAM Journal on Control and Optimization*, Submitted.

Bullo F., Cortés J., and Martínez S. *Distributed control of robotic networks*. Applied Mathematics Series. Princeton University Press, 2009.

Burgarth D. and Giovannetti V. The generalized Lyapunov theorem and its application to quantum channels. *New Journal of Physics*, 9(5):150, 2007.

Cao M., Morse A. S., and Anderson B. D. O. Reaching a consensus in a dynamically changing environment: a graphical approach. *SIAM Journal on Control and Optimization*, 47(2):575–600, February 2008.

Carli R., Fagnani F., Speranzon A., and Zampieri S. Communication constraints in the average consensus problem. *Automatica*, 44(3):671–684, March 2008a.

Carli R. and Zampieri S. Networked clock synchronization based on second order linear consensus algorithms. In *Proceedings of the 49th IEEE Conference on Decision and Control (CDC'10)*, pages 7259–7264, Atlanta, GA, December 2010.

Carli R., Chiuso A., Schenato L., and Zampieri S. A PI consensus controller for networked clocks synchronization. In *Proc. of IFAC World Congress 2008*, Seoul, Corea, July 2008b.

Carvalho A. R. R., Milman P., de Matos Filho R. L., and Davidovich L. Decoherence, pointer engineering, and quantum state protection. *Physical Review Letters*, 86(22):4988–4991, 2001.

CC2420 Datasheet . *2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver*. Chipcom products by Texas Instruments. URL http://focus.ti.com/docs/prod/folders/print/cc2420.html.

Chiang M., Low S. H., Calderbank A. R., and Doyle J. C. Layering as optimization decomposition: A mathematical theory of network architectures. *Proceedings of the IEEE*, 95(1):255–312, 2007.

Cortés J. Distributed algorithms for reaching consensus on general functions. *Automatica*, 44:726–737, 2008.

Costa O., Fragoso M., and Marques R. *Discrete-Time Markov Jump Linear Systems (Probability and its Applications)*. Springer, 2004.

D'Alessandro D. *Introduction to Quantum Control and Dynamics*. Applied Mathematics & Nonlinear Science. Chapman & Hall/CRC, 2007.

D'Alessandro D. and Dahleh M. Optimal control of two level quantum system. *IEEE Transactions on Automatic Control*, 46(6):866–876, 2001.

Davies E. B. *Quantum Theory of Open Systems*. Academic Press, London, New York, 1976.

de Matos Filho R. L. and Vogel W. Engineering the hamiltonian of a trapped atom. *Physical Review A: Atomic, Molecular, and Optical Physics*, 58(3):R1661–R1664, 1998.

Dennis J. E., Jr. and Schnabel R. B. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, 1983.

Di Vincenzo D. P. Quantum computation. *Science*, 270:255–261, October 1995.

Dobson I., Glavitsch H., Liu C.-C., Tamura Y., and Vu K. Voltage collapse in power systems. *IEEE Circuits Devices Magazine*, 8(3):40–45, May 1992.

Dominguez-Duran M., Claros D., Urdiales C., and Coslado F. Dynamic calibration and zero configuration positioning system for wsn. In *Proceedings of the 14th IEEE Mediterranean Electrotechnical Conference (MELECON 2008)*, pages 145–150, May 2008.

Doob J. L. *Stochastic Processes*. John Wiley & Sons, Inc., New York, 1953.

Dorfler F. and Bullo F. Synchronization and transient stability in power networks and non-uniform Kuramoto oscillators. In *Proceedings of the 2010 American Control Conference (ACC)*, pages 930–937, 2010.

Everitt H. O., editor. *Experimental Aspects of Quantum Computing*. Springer, Berlin, 2005.

Fagnani F. and Zampieri S. Average consensus with packet drop communication. *SIAM Journal on Control and Optimization*, 48(1):102–133, February 2009.

Fagnani F. and Zampieri S. Randomized consensus algorithms over large scale networks. *IEEE Journal on Selected Areas in Communications*, 26(4):634–649, May 2008.

Fax J. A. and Murray R. M. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9):1465–1476, September 2004.

Ferrante A., Pavon M., and Raccanelli G. Driving the propagator of a spin system: a feedback approach. In *Proc. 41st IEEE CDC*, volume 1, pages 46–50, Dec. 2002.

Feynman R. P. Simulating physics with computers. *International Journal of Theoretical Physics*, 21, 1982.

Frasca P., Carli R., and Bullo F. Multiagent coverage algorithms with gossip communication: Control systems on the space of partitions. In *Proceedings of the American Control Conference (ACC'09)*, pages 2228–2235, St. Louis, MO, June 2009a.

Frasca P., Carli R., Fagnani F., and Zampieri S. Average consensus on networks with quantized communication. *International Journal of Robust and Nonlinear Control*, 19(16): 1787–1816, November 2009b.

Frommer A. and Szyld D. B. On asynchronous iterations. *Journal of Computational and Applied Mathematics*, 123(1-2):201–216, November 2000.

Ganeriwal S., Kumar R., and Srivastava M. Timing-sync protocol for sensor networks. In *Proceedings of the first international conference on Embedded networked sensor systems (SenSys'03)*, 2003.

Gay D. M. Some convergence properties of Broyden's method. *SIAM Journal on Numerical Analysis*, 16(4):623–630, August 1979.

Goldsmith A. *Wireless Communications*. Cambridge University Press, 2005.

Grivopoulos S. and Bamieh B. Lyapunov-based control of quantum systems. In *Proc. 42th IEEE CDC*, pages 434–438, Dec. 2003.

Gudmundson M. Correlation model for shadow fading in mobile radio systems. *Electronics Letters*, 27(23):2145–2146, November 1991.

Hohlt B., Doherty L., and Brewer E. Flexible power scheduling for sensor networks. In *IEEE and ACM International Symposium on Information Processing in Sensor Networks (IPSN'04)*, April 2004.

Horn R. A. and Johnson C. R. *Matrix Analysis*. Cambridge University Press, New York, 1990.

Hu L. and Evans D. Localization for mobile sensor networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking (MobiCom '04)*, pages 45–57, New York, NY, USA, 2004. ACM.

Ipakchi A. and Albuyeh F. Grid of the future. are we ready to transition to a smart grid? *IEEE Power Energy Magazine*, 7(2):52 –62, March-April 2009.

Jadbabaie A., Lin J., and Morse A. S. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, June 2003.

James M. R. Risk-sensitive optimal control of quantum systems. *Physical Review A: Atomic, Molecular, and Optical Physics*, 69(3):032108, March 2004.

Kelly F. P., Maulloo A. K., and Tan D. K. H. Rate control for communication networks: Shadow prices, proportional fairness and stability. *J. Oper. Res. Soc.*, 49:237–252, 1998.

Khaneja N., Brockett R., and Glaser S. Time optimal control of spin systems. *Physical Review A: Atomic, Molecular, and Optical Physics*, 63:032308, 2001.

Knill E. On protected realization of quantum information. *Physical Review A: Atomic, Molecular, and Optical Physics*, 74(4):042301:1–11, 2006.

Knill E., Laflamme R., and Viola L. Theory of quantum error correction for general noise. *Physical Review Letters*, 84(11):2525–2528, 2000.

Kraus K. *States, Effects, and Operations: Fundamental Notions of Quantum Theory*. Lecture notes in Physics. Springer-Verlag, Berlin, 1983.

Kummerer B. Quantum Markov processes and application in physics. In *Quantum Independent Increment Processes*, volume II of *Lecture Notes in Mathemtics 1866*. Springer Berlin - Heidelberg, 2006.

Kundur P. *Power System Stability and Control*. McGraw-Hill, 1994.

LaSalle J. P. *The Stability and Control of Discrete Processes*, volume 62 of *Applied mathematical sciences*. Springer Verlag New York, 1980.

Lidar D. A., Chuang I. L., and Whaley K. B. Decoherence-free subspaces for quantum computation. *Physical Review Letters*, 81(12):2594–2597, 1997.

Lloyd S. and Viola L. Engineering quantum dynamics. *Physical Review A: Atomic, Molecular, and Optical Physics*, 65:010101:1–4, 2001.

Lopes J. A., Moreira C. L., and Madureira A. G. Defining control strategies for microgrids islanded operation. *IEEE Transactions Power Systems*, 21(2):916–924, May 2006.

Lorincz K. and Welsh M. MoteTrack: a robust, decentralized approach to RF-based location tracking. In **Strang T. and Linnhoff-Popien C.**, editors, *Location- and Context-Awareness*, volume 3479 of *Lecture Notes in Computer Science*, pages 63–82. Springer Berlin / Heidelberg, 2005.

Mamandur K. and Chenoweth R. D. Optimal control of reactive power flow for improvements in voltage profiles and for real power loss minimization. *IEEE Transactions on Power Apparatus and Systems*, PAS-100(7):3185–3194, July 1981.

Maròti M., Kusy B., Simon G., and Àkos Lédeczi . The flooding time synchronization protocol. In *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys'04)*, pages 39–49, 2004.

Massoud Amin S. and Schewe P. F. Preventing blackouts. *Scientific American*, pages 60–67, May 2007.

Massoud Amin S. and Wollenberg B. F. Toward a smart grid: power delivery for the 21st century. *IEEE Power and Energy Magazine*, 3(5):34–41, September-October 2005.

Mirrahimi M., Rouchon P., and Turinici G. Lyapunov control of bilinear schroedinger equations. *Automatica*, 41(11):1987–1994, November 2005.

Mitter S. and Sahai A. Information and control: Witsenhausen revisited. In *Learning, control and hybrid systems*, volume 241 of *Lecture Notes in Control and Information Sciences*, pages 281–293. Springer Berlin / Heidelberg, 1999.

Moreau L. Stability of multiagent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2):169–182, February 2005.

Nedic A. and Ozdaglar A. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

Nielsen M. A. and Chuang I. L. *Quantum Computation and Information*. Cambridge University Press, Cambridge, 2002.

Nocedal J. and Wright S. J. *Numerical Operation*. Springer, 2006.

Oh S., Schenato L., and Sastry S. A hierarchical multiple-target tracking algorithm for sensor networks. In *IEEE International Conference on Robotics and Automation*, pages 2197 – 2202, Barcelona, Spain, April 2005.

Olfati-Saber R., Fax J. A., and Murray R. M. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, January 2007.

Olfati-Saber R. and Murray R. M. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, September 2004.

Pantazis N. A. and Vergados D. D. A survey on power control issues in wireless sensor networks. *IEEE Communications Surveys Tutorials*, 9(4):86–107, 2007.

Patwari N., Hero I., A. O., Perkins M., Correal N. S., and O'Dea R. J. Relative location estimation in wireless sensor networks. *IEEE Transactions on Signal Processing*, 51(8): 2137–2148, August 2003.

Poyatos J. F., Cirac J. I., and Zoller P. Quantum reservoir engineering with laser cooled trapped ions. *Physical Review Letters*, 77(23):4728–4731, 1996.

Rabiee A., Shayanfar H. A., and Amjady N. Reactive power pricing: Problems and a proposal for a competitive market. *IEEE Power Energy Magazine*, 7(1):18–32, January 2009.

Rodrigues Gomes M. H. and Saraiva J. T. Active/reactive bid based dispatch models to be used in electricity markets. *Electric Power Systems Research*, 78:106–121, 2008.

Roozbehani M., Dahleh M., and Mitter S. Dynamic pricing and stabilization of supply and demand in modern electric power grids. In *Proceedings of the First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 543–548, 2010a.

Roozbehani M., Dahleh M., and Mitter S. On the stability of wholesale electricity markets under real-time pricing. In *Proceedings of the 49th IEEE Conference on Decision and Control*, Atlanta, GA, USA, 2010b.

Sakurai J. J. *Modern Quantum Mechanics*. Addison-Wesley, New York, 1994.

Santacana E., Rackliffe G., Tang L., and Feng X. Getting smart. with a clearer vision of the intelligent grid, control emerges from chaos. *IEEE Power Energy Magazine*, 8(2):41–48, March 2010.

Sarlette A. and Sepulchre R. Consensus optimization on manifolds. *SIAM Journal on Control and Optimization*, 48(1):56–76, 2009.

Sarlette A., Sepulchre R., and Leonard N. E. Autonomous rigid body attitude synchronization. *Automatica*, 45(2):572–577, 2009.

Schenato L. and Gamba G. A distributed consensus protocol for clock synchronization in wireless sensor network. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 2289–2294, New Orleans, U.S.A., December 2007.

Shabani A. and Lidar D. A. Theory of initialization-free decoherence-free subspaces and subsystems. *Physical Review A: Atomic, Molecular, and Optical Physics*, 72(4):042303:1–14, 2005.

Simeone O. and Spagnolini U. Distributed time synchronization in wireless sensor networks with coupled discrete-time oscillators. *EURASIP Journal on Wireless Communications and Networking*, 2007:Article ID 57054, 13 pages, 2007.

Solis R., Borkar V. S., and Kumar P. R. A new distributed time synchronization protocol for multihop wireless networks. In *Proceedings of the 45th IEEE Conference on Decision and Control (CDC'06)*, San Diego, December 2006.

Spagnolini U. and Bosisio A. V. Indoor localization by attenuation maps: model-based interpolation for random medium. In *Proceedings of the 9th International Conference on Electromagnetics in Advanced Application (ICEAA)*, pages 1–4, Torino, Italy, September 2005.

Spanos D. P., Olfati-Saber R., and Murray R. M. Distributed sensor fusion using dynamic consensus. In *Proccedings of the 16th IFAC World Congress*, July 2005.

Special issue on sensor networks and applications . *Proceedings of the IEEE*, 91(8), August 2003.

Special issue on technology of networked control systems . *Proceedings of the IEEE*, 95 (1), January 2007.

Speranzon A., Fischione C., Johansson K., and Sangiovanni-Vincentelli A. A distributed minimum variance estimator for sensor networks. *IEEE Journal on Selected Areas in Communications*, 26(4):609–621, 2008.

Strikwerda J. C. A probabilistic analysis of asynchronous iteration. *Linear Algebra and its Applications*, 349(1-3):125–154, July 2002.

Strogatz S. H. *Sync*. Penguin books New York, 2004.

Sundararaman B., Buy U., and Kshemkalyani A. D. Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks*, 3(3):281–323, May 2005.

Szewczyk R., Osterweil E., Polastre J., Hamilton M., Mainwaring A. M., and Estrin D. Habitat monitoring with sensor networks. *Communications of the ACM*, 47(6):34–40, 2004.

Tedeschi E., Tenti P., and Mattavelli P. Synergistic control and cooperative operation of distributed harmonic and reactive compensators. In *IEEE PESC 2008*, 2008.

Tenti P. and Mattavelli P. A time-domain approach to power term definitions under non-sinusoidal conditions. In *6th Intl. Work. on Power Def. and Measurem. under Non-Sinusoidal Cond.*, Milano, Italy, October 2003.

Ticozzi F. and Viola L. Single-bit feedback and quantum dynamical decoupling. *Physical Review A: Atomic, Molecular, and Optical Physics*, 74(5):052328:1–11, 2006.

Ticozzi F. and Viola L. Quantum Markovian subsystems: Invariance, attractivity and control. *IEEE Transactions on Automatic Control*, 53(9):2048–2063, 2008.

Ticozzi F. and Viola L. Analysis and synthesis of attractive quantum Markovian dynamics. *Automatica*, 45:2002–2009, 2009.

Ticozzi F. and Bolognani S. On a canonical QR decomposition and feedback control of discrete-time quantum dynamics. In *Proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems (MTNS 2010)*, Budapest, Hungary, July 2010.

Tsitsiklis J. N., Bertsekas D. P., and Athans M. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31 (9):803–812, September 1986.

van Handel R., Stockton J. K., and Mabuchi H. Feedback control of quantum state reduction. *IEEE Transactions on Automatic Control*, 50(6):768–780, 2005.

Verstraete F., Wolf M. M., and Ignacio Cirac J. Quantum computation and quantum-state engineering driven by dissipation. *Nature Physics*, 5:633–636, 2009.

Viola L., Knill E., and Laflamme R. Constructing qubit in physical systems. *Journal of Physics A: Mathematical and General*, 34(35):7067–7079, 2001.

Viola L., Knill E., and Lloyd S. Dynamical decoupling of open quantum system. *Physical Review Letters*, 82(12):2417–2421, 1999.

Wang G., Kowli A., Negrete-Pincetic M., Shafieepoorfard E., and Meyn S. A control theorist's perspective on dynamic competitive equilibria in electricity markets. In *Proc. IFAC World Congress 2011*, Milano, Italy, 2011. submitted.

Wang X. and Schirmer S. G. Analysis of Lyapunov method for control of quantum states. *IEEE Transactions on Automatic Control*, 55(10):2259–2270, October 2010.

Werner-Allen G., Tewari G., Patel A., Welsh M., and Nagpal R. Firefly-inspired sensor network synchronicity with realistic radio effects. In *ACM Conference on Embedded Networked Sensor Systems (SenSys'05)*, San Diego, November 2005.

Whitehouse K. and Culler D. Calibration as parameter estimation in sensor networks. In *Proceedings of the 1st ACM International Workshop on Wireless sensor networks and applications*, pages 59–67, 2002.

Wiseman H. M. Quantum theory of continuous feedback. *Physical Review A: Atomic, Molecular, and Optical Physics*, 49(3):2133–2150, 1994.

Wu R., Pechen A., Brif C., and Rabitz H. Controllability of open quantum systems with Kraus-map dynamics. *Journal of Physics A: Mathematical and Theoretical*, 40(21):5681–5693, 2007.

Xiao L. and Boyd S. Fast linear iterations for distributed averaging. *Systems and Control Letters*, 53(1):65–78, September 2004.

Yoon S., Veerarittiphan C., and Sichitiu M. L. Tiny-sync: Tight time synchronization for wireless sensor networks. *ACM Transactions on Sensor Networks*, 3(2), June 2007.

Zanca G. and Zorzi F. Measurements on CC2420 radio chipset. Technical report, Department of Information Engineering, University of Padova, Italy, 2008.

Zeidler E. *Applied Functional Analysis: Applications to Mathematical Physics*, volume 108 of *Applied mathematical sciences*. Springer Verlag New York, 1999.