Consensus in Multiagent Systems

Vorgelegt von
Dipl.-Ing.
Miguel Parada Contzen
geb. in Concepción

von der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.-Ing. -

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender:	Prof. Dr.-Ing. Clemens Gühmann
Gutachter:	Prof. Dr.-Ing. Jörg Raisch
Gutachter:	Dr. Robert Shorten
Gutachter:	Dr.-Ing. Johannes Schiffer

Tag der wissenschaftliche Aussprache: 28. September 2017

Berlin 2017

PhD Thesis

Consensus in Multiagent Systems

Miguel Parada Contzen

Miguel Parada Contzen: *Consensus in Multiagent Systems*
PhD Thesis, © October 2017

SUPERVISOR:
Prof. Dr.-Ing. Jörg Raisch

LOCATION:
Fachgebiet Regelungssysteme, Fakultät IV - Elektrotechnik und Informatik, Technische Universität Berlin, Einsteinufer 17, 10587 Berlin, Germany

A Carlos Elías y a Marina.

Abstract

During the last decade, the problem of *consensus* in multiagents systems has been studied with special emphasis on graph theoretical methods. Consensus can be regarded as a control objective in which is sought that all systems, or *agents*, in a network have an equivalent output value. This is achieved through a given control strategy usually referred to as consensus *algorithm*. The motivation to study such an objective comes from different areas, such as engineering, and social and natural sciences. In the control engineering field, important application examples are formation control of swarms of mobile robots and distributed electric generation. Most of the work in this area is done for agents with single or double integrator dynamics and algorithms derived as the Laplacian matrix of undirected graphs. That is, consensus is often studied as a property of particular networks and particular algorithms.

In this thesis, we tackle the problem from a Control Theory perspective in an attempt to augment the class of systems that can be studied. For that we translate the consensus problem from its classical formulation for integrator systems into a general continuous time stability problem. From here, different algorithmic strategies under several dynamical assumptions of the agents can be studied through well known control theoretical tools – as Lyapunov's theory, linear matrix inequalities (LMI), or robust control – along with graph theoretical concepts. In particular, in this work we study consensus of agents with arbitrary linear dynamics under the influence of linear algorithms not necessarily derived from graphs. Furthermore, we include the possibility that the agents are disturbed by several factors as external signals, parameter uncertainties, switching dynamics, or communication failure. The theoretical analysis is also applied to the problem of power sharing in electric grids and to the analysis of distributed formation control.

Zusammenfassung

Im Laufe der letzten zehn Jahre wurde das Problem des *Konsenses* in Multi-Agenten-Systemen mit besonderem Augenmerk auf graphen-theoretische Methoden erforscht. Ein Konsens kann als ein Regelungsziel interpretiert werden, in welchem alle Systeme, oder *Agenten*, in einem Netzwerk identische Ausgangswerte anstreben. Dies wird durch eine Regelungsstrategie erreicht, die man üblicherweise als *Konsens Algorithm* bezeichnet. Motiviert wird diese Regelungsstrategie aus verschiedenen Bereichen, wie den Ingenieurwissenschaften und den Sozial- und Naturwissenschaften. Wichtige Beispiele im Bereich der Regelungstechnik sind die Formationsregelung in Schwärmen von mobilen Robotern oder die verteilte Energieerzeugung. Die Mehrheit der Werke in diesem Bereich konzentriert sich auf Agenten mit Einfach- oder Doppel-Integratordynamiken, sowie Algorithmen abgeleitet aus der „Laplacian-Matrix" von ungerichteten Graphen. In anderen Worten, der Konsens wird oft als eine Eigenschaft besonderer Netzwerke und Algorithmen erforscht.

In dieser Dissertation wird das Problem des Konsenses aus einer regelungs-theoretischen Perspektive betrachtet. Dies wird durch die Übersetzung des Konsens-Problems aus seiner klassischen Formulierung für Integratorensysteme zu einem zeitkontinuierlichen Stabilität-Problem bewerkstelligt. Basierend auf bekannten Methoden der Regelungstechnik (zum Beispiel, auf Lyapunovs Theorie, den linearen Matrixungleichungen (LMI), oder der Robustregelung) und graph-theoretische Ideen werden unterschiedliche algorithmische Strategien und mehrere Arten von dynamischen Agenten behandelt. Im Besonderen wird in dieser Arbeit der Konsens von Agenten mit willkürlichen linearen Dynamiken betrachtet. Dies ist eine Erweiterung zu den klassischen Einfach- und Doppelt-Integratoren Dynamiken. Desweiteren betrachten wir Algorithmen, die nicht unbedingt von Graphen abgeleitet sind. Es wird weiterhin das Verhalten der Agenten unter dem Einfluss von Störungen untersucht. Dies beinhaltet zum Beispiel externe Störungssignale, parametrische Unsicherheiten, geschaltete Dynamiken oder Kommunikationfehler. Die Anwendbarkeit der hier erlangten theoretischen Ergebnisse wird am Beispiel der Formationsregelung und der Stabilisierung verteilter Energiesysteme demonstriert.

Acknowledgments

There is a number of people who have, directly or indirectly, contributed to this project and I would like to thank them greatly. I will proceed chronologically as it seems to be the least unfair of all possible permutations.

First I would like to thank my parents, Patty & Rigo, and my sister, Marce, for their unconditional support, endless laughter and loving care through all the many steps that ended with this thesis. I would also like to extend my gratitude to the rest of my family, who also have been a great support in many periods and aspects of my work.

I would like to thank Prof. Dr. Jörg Raisch for accepting to guide me through this process and believing in me, without much of a recommendation about myself but solely my word. My research at the TU-Berlin is funded by the Chilean National Agency of Technology and Scientific Research (CONICYT), through the program *Becas Chile*.

I also need to thank and acknowledge the help of many people and friends I met at the University: Doro & Johannes, Xavier, Truong, Philipp, etc. Many friends have also helped a lot during my time in Berlin. Among them I need to thank specially Christoph, Vale & Bine, Anita, Daca, and the players at Schachfreunde Berlin 1903. Most importantly, I need to thank my wife Marina for her support, love and understanding during all the time we have been together. Specially now that Carlos Elías is also with us.

Muchas gracias a todos.

Berlin, May 2017

M.P.C.

Contents

List of Figures

List of Tables

Acronyms

AA. Autonomous Agent. See Section 4.1.2.

AAN. Autonomous Agents Network. See Section 4.1.2.

BRL. Bounded Real Lemma. See Section A.4.

CA. Connected Agent. See Section 5.2.1.

CAN. Connected Agents Network. See Section 5.2.1.

IA. Integrator Agent. See Section 4.1.2.

IN. Integrator Agents Network. See Section 4.1.2.

IAD. Integral Absolute Deviation. See Section 4.2.3.

IAE. Integral Absolute Error. See Section 7.2.5.

ISD. Integral Square Deviation. See Section 4.2.3.

ISE. Integral Square Error. See Section 7.2.5.

LMI. Linear Matrix Inequality. See Section 2.2.2.

MIMO. Multiple Input Multiple Output.

PID. Proportional Integral Derivative Controller. P, PI, PD, and ID controllers are defined analogously.

RMS. Root Mean Square. See Section B.1.

SISO. Single Input Single Output.

SVD. Singular Values Decomposition. See Section A.1.

VSI. Voltage Source Inverter. See Section 6.1.2.

1 Introduction

1.1. Motivation

The idea of *consensus* in multi agents systems had gain much attention in the control society during the last decade. The analogy of a swarm of birds is a useful way to explain the main characteristic of the problem: a group of similar systems (or *agents*) agree to coordinate some important variables through a given information exchange strategy (or *algorithm*). Consensus is in this way a control objective where all systems in a network aim to have an equivalent output. Research in this topic is motivated by applications in many fields such as engineering, computer science, physics, social sciences, and biology. In engineering, attention has been paid to applications where multiple agents have to coordinate to perform common tasks. For example, manoeuvres of groups of vehicles, large array of telescopes, sensor networks, electric grids, or mobile cooperative robots.

Over the years, multiple works have driven attention to this topic. The publication of books like [22, 33, 34] shows that the field has already reached an advanced state. However the topic is still a popular area of research as shown in the review paper [29] where more than a hundred references are quoted. More recently, the review paper [5] includes over three hundred references.

Most of the work in the area is based on Graph Theoretical approaches to the problem and single or double integrators dynamics. Examples of this are the already quoted publication, the doctoral theses [14, 31] and an increasing number of papers such as [1, 2, 15, 16, 18, 32]. From a more mathematical perspective, some examples are [6, 118, 119, 17, 23, 46] and the references within. The particular dynamics with which these publications deal makes it difficult to systematically extend the results to other cases of interest. Some publications, *e.g.* [20, 21, 36, 38, 47, 53, 54, 60], extend the graph theoretical approaches to systems with more general linear dynamics by introducing other control theoretical tools like Linear Matrix Inequalities (LMI), Optimal Control Theory or Lyapunov's Theory. Consensus in multi agents systems is also defined and studied in fields far away from classical control theory. For example, Games Theoretical approaches [28, 37, 40] or Max Plus algebra systems [25, 26]. Nevertheless, these efforts have still left many unanswered questions.

One of these, which has taken much attention in the field, is related to switching algorithms. This has become an important topic in the area and can be found in many publications as, for example, [9, 19, 22, 31, 41, 47, 50–52, 55, 57–61]. The study of switching systems cannot be avoided because of the nature of the consensus problem where communication plays a fundamental role. The communication channels between the agents are, in practice, far from ideal and, therefore, robustness or persistence against temporal or permanent loss of communication needs to be addressed. Unfortunately, the limitations of the switching systems theory leave, again, many unanswered questions. Other interesting issue regarding to communication is signals delay. This is present in many of the already quoted works but also in, for example, [35, 43, 56].

A related problem to consensus is that of *formation*. This is perhaps easier to relate to practical cases and explains up to a certain point the interest in single and double integrators dynamics. Indeed, in formation control, consensus is not searched in the state variables or outputs of the agents (what can be interpreted as their "velocity"), but on the integral over time of them (what can be interpreted as the agents' "position"). Examples of this can be found in the quoted books and theses, but also in a wide spectrum of papers with emphasis on different aspects: [3, 7, 8, 10–13, 24, 27, 30, 32, 39, 42, 44, 45, 48, 49, 62]. As in the "velocity" case, the described communication issues are also present.

These various approaches are far to constitute an ordered and comprehensive body of knowledge that is flexible enough to treat complex scenarios from a control systems perspective. The view of consensus as a collection of particular problems (particular dynamics, algorithms, or communication dynamics) makes it difficult to give the subject a theoretical frame robust enough to study coordination as a control objective of the same relevance as stability, robustness or frequency response. That is why, in this work, a general view of the problem is proposed by means of classical Control Theory concepts. This approach is thought as a complement to the existing works on the subject, allowing the gradual extension of the analysis to include more complex assumptions on the systems.

To do this, consensus can be intuitively compared with the equilibrium point of a system that resumes the characteristics of the whole network. The underlaying hypothesis is that consensus can be studied through a (not unique) characterization of the whole network as a stability problem. However, this aspect is usually not expressly addressed in the quoted works. Some exceptions to this are the recent papers [4, 35, 50, 51, 60], where consensus is studied as the convergence property of a *differences vector* between the outputs of one of the agents and the rest of them. This agent that serves as comparing references is usually called the *leader* agent.

In this thesis, this idea is further exploited to formally translate the consensus problem into a classical stability one through the introduction of an analysis tool named *hierarchical organization*. This tool is neither unique nor does it have a predefined structure. Therefore, the resulting differences-vector does not necessarily represent the differences between one unique agent and the rest of them, but, for example, between each agent and the next one in line or in a pyramidal structure. From here, the problem is addressed by means of standard Control Theory and different situations are investigated by changing the assumptions about the dynamics and characteristics of the agents and their relations.

1.2. Structure

After this introduction, this thesis is divided in two main parts. The first part (Chapters 2 and 3) gives a general mathematical background for the rest of the thesis. The second part (Chapters 4 to 7) deals with the consensus problem from different aspects. Additionally, two appendices of needed results are considered. A brief summary of the contents of each chapter is as follows:

- **Chapter 2.** A general overview of Graphs Theory is presented. The main notions in this area are generalized to model, afterwards, the interaction of multiple signals between agents. In the second part, this chapter deals with general Lyapunov's Stability Theory and its relationship to Linear Matrix Inequalities (LMI).
- **Chapter 3.** Some existing results in Switched Systems stability are shown. First, polytopic systems are defined and from there time dependent switching systems are studied. This is done for system with deterministic and stochastic switching laws.
- **Chapter 4.** The problem of Consensus is explained and formally stated. First a general description of Multi Agents Systems is given and then the concept of consensus is formalized through the concept of organization and consensus error.
- **Chapter 5.** The consensus problem is analyzed through different control strategies. The first Section deals with consensus algorithms derived from loopless weighted undirected graphs, which is the standard approach to the problem, in networks composed of, first, pure integrator systems and, secondly, general lineal dynamical systems. The second section proposes other kind of algorithms to address problems derived from particular characteristics of the agents or to generalize other strategies. The last section of this chapter deals briefly with some generalizations of dynamics of the agents or their control strategies. Chapters 4 and 5 can be considered the main part of the thesis.
- **Chapter 6.** An application of the described methodology is used to treat the problem of power sharing in electric microgrids. The first section of the chapter models a microgrid as a multiagent system, then the problems of active and reactive power sharing are treated separately.
- **Chapter 7.** The formation problem, as a special case of consensus, is studied in small networks. First a description of the problem and the multiagent systems is given. Then it follows a centralized solution of the problem that can be totally distributed in the agents without the need of communication. Finally, further distributed issues are addressed from a centralized perspective.
- **Chapter 8.** This chapter presents comprehensively the conclusions of the thesis.

- **Appendix A.** This appendix presents several non-original results used in different parts of the thesis. The sections are respectively focused on pseudoinverses, norms, general algebraic results, LMI results and, lastly, a linear optimization problem. This complements the mathematical background chapters.

- **Appendix B.** This appendix deals with the model of electrical grids as a complement to Chapter 6. The first section is about general definitions on the electrical field. The second section derives the active and reactive power models by means of a circuital analysis of a generic grid.

Even though this thesis is meant to be read and understood in a sequential way, there are strong relationships between certain chapters and sections. A schematic representation of these relationships is to be seen in Figure 1.1.

1.3. Notation

Through this thesis the set of real numbers is denoted $\mathbb{R}$, of positive real numbers $\mathbb{R}^+$, of non-negative real numbers $\mathbb{R}_0^+$ and of complex numbers $\mathbb{C}$. Powers of these sets are used to denote vectors and matrices sets. For example, $\mathbb{R}^n$ is the set of all n-dimensional vectors with real elements and $\mathbb{C}^{m\times n}$ is the set of all $m \times n$-dimensional matrices with complex elements. Other sets are denoted by capital Latin letters. The cardinality of a set S, *i.e.* the number of elements in the set, is denoted $|S|$. The complement of a set S is the set of all elements that are not elements of S. Union and intersection of sets are denoted by the standard symbols $\cup$ and $\cap$ respectively, while the intersection of a set S_1 and the complement of another set S_2, by $S_1 \backslash S_2$. Graphs, and graphs related sets (see Section 2.1), are denoted by calligraphic symbols as $\mathcal{G}$, $\mathcal{V}$, $\mathcal{E}$, etc.

Given a set $S \subset \mathbb{R}$, its minimum is denoted $\min\{S\}$; its maximum, $\max\{S\}$; its infimum $\inf\{S\}$; and its supremum, $\sup\{S\}$. An optimization problem is denoted with the help of these sets. For example, minimize a non-negative functional $J(\mathbf{x}) \geqslant 0$, over a vector $\mathbf{x} \in \mathbb{R}^n$ subject to a set of conditions $f(\mathbf{x}) < 0$ is denoted $J^* = \inf\{J(\mathbf{x}) \in \mathbb{R}^+ | f(\mathbf{x}) < 0\}$. Alternatively, it can be written through the conventional use in optimization texts:

$$\begin{aligned} \min_{\mathbf{x}} \quad & J(\mathbf{x}) \\ \text{s.t.} \quad & f(\mathbf{x}) < 0 \\ & J(\mathbf{x}) \geqslant 0 \end{aligned}$$

The optimal value is denoted by J^* and the argument of the problem (the vector for which the optimum value holds) as $\mathbf{x}^*$.

In general, matrices will be represented by bold capital Latin letters (*e.g.* $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times p}$, $\mathbf{C} \in \mathbb{R}^{q\times n}$, etc.), vectors by bold small Latin letters (*e.g.* $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^q$, $\mathbf{u} \in \mathbb{R}^p$, etc.), and scalars as Greek small letters (*e.g.* $\alpha > 0$, $\varepsilon \ll 1$, etc.). Exceptions to these rules are matrix

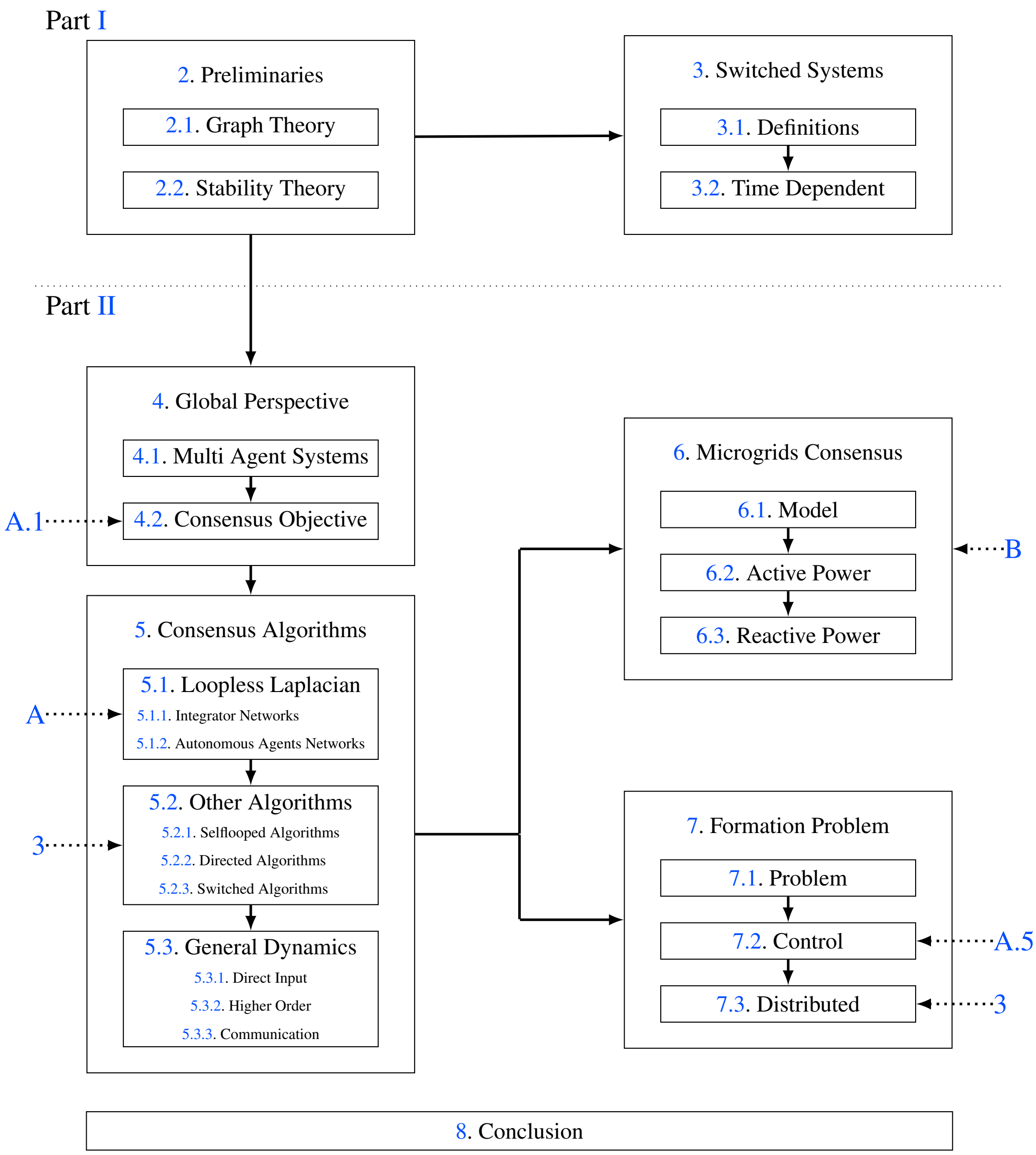


Figure 1.1.: Chapters relationships within this thesis.

operators over a graph $\mathscr{G}$ (*e.g.* $L(\mathscr{G}) \in \mathbb{R}^{N\times N}$), the so called parametrization vector $\boldsymbol{\alpha} \in \mathbb{R}^{M}$ in Section 3.1.2, matrices related to the Finsler Lemma A.14, and functions of already defined matrices within a matrix inequality denoted by Franktur font symbols. *e.g.* $\mathfrak{M}_{11} = \mathbf{A} + \mathbf{B}\mathbf{R}^{-1}\mathbf{B}'$ in

$$\mathfrak{M} := \begin{bmatrix} \mathfrak{M}_{11} & \mathbf{C} \\ \star & \mathbf{D} \end{bmatrix} < 0.$$

Matrix $\mathbf{A}' \in \mathbb{R}^{n\times m}$ is the transpose of $\mathbf{A} \in \mathbb{R}^{m\times n}$. Matrix $\mathbf{A}^* \in \mathbb{C}^{n\times m}$ is the conjugate transpose of $\mathbf{A} \in \mathbb{C}^{m\times n}$. The notation "$\star$" is used to indicate a symmetric block within a matrix, for example,

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \star & \mathbf{C} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}' & \mathbf{C} \end{bmatrix}.$$

Matrix inequalities such as $\mathbf{A} < 0$ ($\mathbf{A} > 0$) are used to indicate that matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is symmetric negative (or positive) definite (see Section 2.2.2). The inverse of a matrix $\mathbf{A} \in \mathbb{C}^{n\times n}$ is denoted as $\mathbf{A}^{-1} \in \mathbb{C}^{n\times n}$. The (Moore-Penrose) pseudoinverse (see Section A.1) of a matrix $\mathbf{A} \in \mathbb{C}^{n\times m}$ is denoted $\mathbf{A}^{+} \in \mathbb{C}^{m\times n}$. An element in the (block) position (i,j) of a matrix $\mathbf{A}$ is denoted $[\mathbf{A}]_{ij}$.

Other operations over matrix $\mathbf{A} \in \mathbb{C}^{n\times m}$ are its rank, denoted $r = \operatorname{rank}\{\mathbf{A}\}$; its dimension, $d = \dim\{\mathbf{A}\} = \min\{n,m\}$; the determinant of a square matrix, $\det\{\mathbf{A}\} = |\mathbf{A}|$; the set of eigenvalues of a square matrix of dimension n, $\operatorname{eig}\{\mathbf{A}\} = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$; and the set of singular values of a matrix $\operatorname{svd}\{\mathbf{A}\} = \{\sigma_1, \sigma_2, \ldots, \sigma_r, 0, \ldots, 0\}$. The real part of a complex matrix $\mathbf{A} = \mathbf{X} + j\mathbf{Y} \in \mathbb{C}^{n\times m}$, is denoted $\operatorname{real}\{\mathbf{A}\} = \operatorname{real}\{\mathbf{X} + j\mathbf{Y}\} = \mathbf{X} \in \mathbb{R}^{n\times m}$; and its imaginary part, $\operatorname{imag}\{\mathbf{A}\} = \operatorname{imag}\{\mathbf{X} + j\mathbf{Y}\} = \mathbf{Y} \in \mathbb{R}^{n\times m}$. In general scalars, vectors and matrices can be time dependent functions, but for simplicity's sake and when it is clear enough from the context, the time dependence will be dropped. A time delayed vector is defined as $\mathbf{v}_\tau = \mathbf{v}(t-\tau)$.

The identity matrix and the null matrix are respectively denoted by $\mathbf{I}$ and $\mathbf{0}$. A column vector of identity matrices is denoted as $\mathbf{1} = [\mathbf{I}, \mathbf{I}, \ldots \mathbf{I}]'$. If necessary, the dimensions of these matrices will be stated as an index, *e.g.* $\mathbf{I}_q$ is the identity matrix in $\mathbb{R}^{q\times q}$, $\mathbf{0}_{m\times n}$ is the zero matrix in $\mathbb{R}^{m\times n}$, and $\mathbf{1}_{Nq\times q}$ is composed of N identity matrix in $\mathbb{R}^{q\times q}$. The matrix stack operators over an ordered set $S = \{s_1, .., s_N\}$ are defined as:

$$\begin{aligned}
\mathrm{col}\{\mathbf{X}_i\}_{i\in S} &:= \begin{bmatrix} \mathbf{X}_{s_1} \\ \vdots \\ \mathbf{X}_{s_N} \end{bmatrix}, \\
\mathrm{row}\{\mathbf{X}_i\}_{i\in S} &:= \begin{bmatrix} \mathbf{X}_{s_1} & \cdots & \mathbf{X}_{s_N} \end{bmatrix}, \text{ and} \\
\mathrm{diag}\{\mathbf{X}_i\}_{i\in S} &:= \begin{bmatrix} \mathbf{X}_{s_1} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{X}_{s_N} \end{bmatrix} = \sum_{i=1}^{N} \mathbf{s}_i \mathbf{X}_{s_i} \mathbf{s}_i'.
\end{aligned}$$

Note that matrices $\mathbf{X}_i$, $i \in S$, must have compatible dimensions. We denote $\mathbf{s}_i \in \mathbb{R}^{Nq\times q}$ as the N blocks column vector where each element is $\mathbf{0}_{q\times q}$ but the i-th element that is the identity matrix $\mathbf{I}_q$, so that $\mathbf{1} = \sum_{i=1}^{N} \mathbf{s}_i$.

The probability of occurrence of an event q will be denoted as $P\{q\} \in [0,\ 1]$. The expected value of a variable $X \in \mathscr{X}$, which can be a scalar, a vector, or a matrix, in the discrete space $\mathscr{X}$, will be denoted as $E\{X\} := \sum_{x\in\mathscr{X}} xP\{x = X\}$.

1.4. Publications

Some parts of this thesis are based on the following publications, to all of which the author is the main contributor.

Miguel Parada Contzen. "Consensus Algorithm Analysis and Design For Agents With Linear Dynamics." In: *European Control Conference (ECC)*. 2015.

Miguel Parada Contzen. "Consensus in networks with arbitrary time invariant linear agents." Accepted in European Journal of Control. 2017.

Miguel Parada Contzen and Jörg Raisch. "A polytopic approach to switched linear systems." In: *IEEE Multi-Conference on Systems and Control (MSC)*. 2014.

Miguel Parada Contzen and Jörg Raisch. "Active Power Consensus in Microgrids." In: *International Symposium on Smart Electric Distribution Systems and Technologies (EDST)*. 2015.

Miguel Parada Contzen and Jörg Raisch. "Reactive Power Consensus in Microgrids." In: *European Control Conference (ECC) 2016*. 2016.

Part I.

Mathematical Background

2
Preliminaries

This chapter establishes the basic background to understand this thesis. Consensus is usually studied through graph theoretical methods and therefore a short summary on the subject is presented. A summary of Lyapunov's Theory on stability of systems with emphasis on Linear Matrix Inequalities (LMI) follows. Other relevant results are to be found in Appendix A.

2.1. Graph Theory

Traditionally, the consensus problem is strongly related to *Graph Theory*. Most of the consensus works use intensely graph theoretical methods for description and analysis of networks. Some key examples are [22, 33, 34]. In this section, basic notions of the subject are presented based on the quoted works and specialized books as [116, 117, 122]. The following definitions are modified for the purpose of this thesis from the standard notions of graph theory.

Definition 2.1.1. An *undirected graph* is a tuple $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ where

- $\mathscr{V} = \{1, 2, \ldots N\}$ is a set of N *nodes* or *vertices*, and
- $\mathscr{E} \subseteq \{(i,j) \in \mathscr{V} \times \mathscr{V}\}$ is a set of *edges*, where we interpret that the edge denoted $(i,j) \in \mathscr{E}$ is the same as the edge $(j,i) \in \mathscr{E}$.

Note that when referring to an unordered edge we slightly abuse notation by representing it by an ordered pair (i,j). With this notation we mean that an unordered edge between nodes i and j of an undirected graph can be equivalently specified either by the pair (i,j) or the pair (j,i). Which is not the same as the graph having two different ordered edges.

In the context of this thesis, the nodes correspond to *agents*, and the existence of an edge labeled $e_k = (i_k, j_k) \in \mathscr{E} = \{e_1, e_2, \cdots, e_{|\mathscr{E}|}\}$ means that agent i_k and agent j_k interact with each other either through input and output signals, or by a hierarchical relationship. Note that this defines implicitly an arbitrary indexation of the edges which is independent of the labeling of the nodes. We will refer to edges $(i,i) \in \mathscr{E}$ indistinctly as *loops* or *selfloops*. A graph without selfloops will be called *loopless*.

Definition 2.1.2. A *path* is an ordered sequence of nodes in an undirected graph such that any pair of consecutive nodes is connected by an edge. An undirected graph is *connected* if there is a path between every two nodes and unconnected otherwise. A *fully connected* graph is such that there is an undirected edge between every pair of different nodes.

Definition 2.1.3. A (spanning) *tree* $\mathscr{T}$ is an undirected graph over a set of nodes $\mathscr{V}$ which is connected and has $N-1$ edges, where $N = |\mathscr{V}|$ is the number of nodes.

The usual definition of tree accepts that the graph has less than $N-1$ edges. However, in the context of this work, only spanning trees are meaningful and therefore, to abbreviate, we will refer to them simply as *trees*. Equivalent definitions for (spanning) trees can be found with ease, but Definition 2.1.3 is sufficient for our needs. Cayley's Formula states that in a set of N vertices, N^{N-2} spanning trees can be defined.

Definition 2.1.4. The *neighbor set* of a node $i \in \mathscr{V}$ in an undirected graph $\mathscr{G}$ is defined as

$$\mathscr{N}_i = \{j \in \mathscr{V} | (i,j) \in \mathscr{E} \ \wedge \ i \neq j\}$$

Definition 2.1.5. Am *undirected weighted graph* is a tuple $\mathscr{G}_w = (\mathscr{G}, w_q)$ where

- $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ is an undirected graph, and
- $w_q : \mathscr{E} \to \mathscr{M} \subseteq \mathbb{R}^{q\times q} \backslash \{\mathbf{0}\}$ is a function that associates a non-zero positive definite *weight* matrix to each edge.

This last definition is a generalization of the usual one because it considers that the weight associated with each edge is not only a positive scalar, but a $(q \times q)$ matrix. This consideration is done to model multiple input/output signals of the agents. *e.g.* three-dimensional position or speed of a vehicle; active and reactive power of an electric generation unit; etc.

Definition 2.1.6. The *dimension* of a weighted graph is the dimension of the image matrix space $\mathscr{M}$ of the weight function w_q. That is, $\dim\{\mathscr{G}_w\} = q \iff \mathscr{M} \subseteq \mathbb{R}^{q\times q} \backslash \{\mathbf{0}\}$.

Definition 2.1.7. An *unweighted* graph is a special case of weighted graphs where $w_q((i,j)) = \mathbf{I}_q$, $\forall (i,j) \in \mathscr{E}$.

Note that, contrary to the conventional definition, in an unweighted graph we do not deny the existence of the weight function, but merely restrict it to the trivial case. This is thought so that the dimension of an unweighted graph can be defined.

Definition 2.1.8. A *strictly directed graph*, or strict digraph, is an unweighted graph where the edge set $\mathscr{E} \subseteq \mathscr{V} \times \mathscr{V}$ is redefined so that each edge has an unique orientation. That is, $(i,j) \in \mathscr{E} \Rightarrow (j,i) \notin \mathscr{E}$.

In this case the notation $(i,j) \in \mathscr{V} \times \mathscr{V}$ and $(j,i) \in \mathscr{V} \times \mathscr{V}$ represent two different edges that cannot be simultaneously part of a strict digraph. Note that strict digraphs cannot have selfloops. For every loopless undirected unweighted graph, $2^{|\mathscr{E}|}$ strict digraphs can be defined by giving an orientation to every edge. An arbitrary strict digraph generated from an undirected unweighted graph $\mathscr{G}$ will be denoted by $\mathscr{G}^o$.

To represent graphs, nodes will be drawn as black dots and undirected edges as lines linking the nodes. If necessary, nodes will be labeled by a natural number over it. If several graphs in a figure are defined over the same set of nodes, the labels will be shown only in the first graph. When no label is shown over an edge, it will be assumed that it is unweighted. If the weight needs to be shown, it will be represented by the corresponding weight matrix. In some cases, the label e_k is used to name the edge. The edges of a strictly directed graph will be represented with an arrow at the end of the line showing its direction. Their labeling is as for the undirected edges. See Figure 2.1 for an example.

Because of the inclusion of matrix weights, the usual definitions of graph related matrices also need to be generalized.

Definition 2.1.9. The *Incidence Matrix*, denoted $D(\mathscr{G}^o)$, of a strict digraph $\mathscr{G}^o$ of $\dim\{\mathscr{G}^o\} = q$ is defined as a matrix where each block $o_{ik} = [D(\mathscr{G}^o)]_{ik}$ takes either the value $o_{ik} = -\mathbf{I}_q$ if the edge e_k has its origin in i, $o_{ik} = \mathbf{I}_q$ if node i is the destination of edge e_k or $o_{ik} = \mathbf{0}_{q\times q}$ otherwise.

Note that this definition assumes that the edges are labeled by the index k. Different labeling systems for the edges of a graph would lead to different incidence matrices. In total, if the graph has $|\mathscr{E}|$ edges, $|\mathscr{E}|!$ different labeling systems can be defined.

Definition 2.1.10. The *adjacency matrix*, denoted $A(\mathscr{G}_w)$, of a weighted graph $\mathscr{G}_w$ is constructed so that each block $\mathbf{W}_{ji} := [A(\mathscr{G}_w)]_{ij}$ takes the value $\mathbf{W}_{ji} = w_q((j,i)) \in \mathscr{M}$ if $(j,i) \in \mathscr{E}$ or $\mathbf{W}_{ji} = \mathbf{0}$ otherwise.

Note that this matrix is symmetric for undirected weighted graphs.

Definition 2.1.11. The matrix *degree of node i*, Δ_i, in an undirected weighted graph is defined as the sum of all elements of the respective block column or block row of the adjacency matrix plus the corresponding diagonal element (the weight of the selfloop). *i.e.* $\Delta_i = \sum_{j=1}^{N} \mathbf{W}_{ij} + \mathbf{W}_{ii} = \sum_{i=1}^{N} \mathbf{W}_{ij} + \mathbf{W}_{ii}$. The *degree matrix* is $\Delta(\mathscr{G}_w) = \operatorname{diag}\{\Delta_1, \dots, \Delta_N\}$.

Definition 2.1.12. The *Laplacian matrix* of an undirected weighted graph $\mathscr{G}_w$ is $L(\mathscr{G}_w) := \Delta(\mathscr{G}_w) - A(\mathscr{G}_w)$.

This matrix is sometimes referred to as the "loopy Laplacian" [172]. We decompose the Laplacian matrix as $L(\mathscr{G}_w) = \hat{L}(\mathscr{G}_w) + \operatorname{diag}\{\mathbf{W}_{ii}\}_{i\in\mathscr{V}}$. Then, each column and row of $\hat{L}(\mathscr{G}_w)$ sums up to zero. This can respectively be written as $\mathbf{1}'\hat{L}(\mathscr{G}_w) = \mathbf{0}$ and $\hat{L}(\mathscr{G}_w)\mathbf{1} = \mathbf{0}$ where $\mathbf{1} = \operatorname{col}\{\mathbf{I}_q\}_{i\in\mathscr{V}} \in \mathbb{R}^{Nq\times q}$ is a column of N $(q\times q)$-identity matrices. Note that $L(\mathscr{G}_w) = \hat{L}(\mathscr{G}_w)$ whenever the graph has no selfloops.

Lemma 2.1. *Given an undirected weighted graph* $\mathscr{G}_w = (\mathscr{V}, \mathscr{E} = \{e_1, \cdots, e_{|\mathscr{E}|}\}, w_q)$ *without selfloops, then*

$$\hat{L}(\mathscr{G}_w) := D(\mathscr{G}^o)\mathbf{W}D'(\mathscr{G}^o) = \Delta(\mathscr{G}_w) - A(\mathscr{G}_w),$$

where $\mathbf{W} = \operatorname{diag}\{\mathbf{W}_{i_k j_k}\}_{k=1}^{|\mathscr{E}|} = \operatorname{diag}\{w_q(e_k)\}_{k=1}^{|\mathscr{E}|}$ *and* $\mathscr{G}^o$ *is an arbitrary strict digraph defined from* $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ *with the edge labeling given by* $k \in \{1, 2, \dots, |\mathscr{E}|\}$.

Proof. The proof consists in algebraic matrix manipulations to show the equivalence. First note that considering the labeling of nodes given by k, the adjacency matrix and the degree matrix can be written in the following way:

$$A(\mathscr{G}_w) = \sum_{k=1}^{|\mathscr{E}|} \left(\mathbf{s}_{i_k} \mathbf{W}_{i_k j_k} \mathbf{s}'_{j_k} + \mathbf{s}_{j_k} \mathbf{W}_{i_k j_k} \mathbf{s}'_{i_k} \right),$$

$$\Delta(\mathscr{G}_w) = \sum_{k=1}^{|\mathscr{E}|} \left(\mathbf{s}_{i_k} \mathbf{W}_{i_k j_k} \mathbf{s}'_{i_k} + \mathbf{s}_{j_k} \mathbf{W}_{i_k j_k} \mathbf{s}'_{j_k} \right).$$

Where $\mathbf{s}_i$ is a matrix column vector composed of N square blocks and with the identity matrix in the i-th block and zeros everywhere else. Now note that the triple product in the equation of the Lemma can be decomposed as:

$$D(\mathscr{G}^o)\mathbf{W}D'(\mathscr{G}^o) = D(\mathscr{G}^o) \left(\sum_{k=1}^{|\mathscr{E}|} \hat{\mathbf{s}}_k \mathbf{W}_{i_k j_k} \hat{\mathbf{s}}'_k \right) D'(\mathscr{G}^o)$$

Where $\hat{\mathbf{s}}_k$ is a matrix column vector composed of $|\mathscr{E}|$ square blocks and with the identity matrix in the k-th block and zeros everywhere else. The product $D(\mathscr{G}^o)\hat{\mathbf{s}}_k$ can also be decomposed into $D(\mathscr{G}^o)\hat{\mathbf{s}}_k = \mathbf{s}_{i_k} - \mathbf{s}_{j_k}$ and so,

$$D(\mathscr{G}^o)\mathbf{W}D'(\mathscr{G}^o) = \sum_{k=1}^{|\mathscr{E}|} \left(\mathbf{s}_{i_k} \mathbf{W}_{i_k j_k} \mathbf{s}'_{i_k} + \mathbf{s}_{j_k} \mathbf{W}_{i_k j_k} \mathbf{s}'_{j_k} \right) - \left(\mathbf{s}_{i_k} \mathbf{W}_{i_k j_k} \mathbf{s}'_{j_k} + \mathbf{s}_{j_k} \mathbf{W}_{i_k j_k} \mathbf{s}'_{i_k} \right).$$

Which proves the equivalence. □

Example 2.1. To illustrate the proof of Lemma 2.1, consider the graphs depicted in Figure 2.1 where $\mathscr{G}$ is an undirected graph with $\mathscr{V} = \{1,2,3\}$, $\mathscr{E} = \{(1,2),(1,3)\}$; the weight function of $\mathscr{G}_w$ is such that $w_q((1,2)) = \mathbf{W}_{12} = \mathbf{W}'_{12}$ and $w_q((1,3)) = \mathbf{W}_{13} = \mathbf{W}'_{13}$; and the directed edges of $\mathscr{G}^o$ are $e_1 = (1,2)$ and $e_2 = (1,3)$. The corresponding matrices are:

$$L(\mathscr{G}_w) = \begin{bmatrix} \mathbf{W}_{12}+\mathbf{W}_{13} & -\mathbf{W}_{12} & -\mathbf{W}_{13} \\ -\mathbf{W}_{12} & \mathbf{W}_{12} & \mathbf{0} \\ -\mathbf{W}_{13} & \mathbf{0} & \mathbf{W}_{13} \end{bmatrix}, D(\mathscr{G}^o) = \begin{bmatrix} -\mathbf{I} & -\mathbf{I} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}.$$

The triple product can be decomposed as:

$$D(\mathscr{G}^o)\mathbf{W}D'(\mathscr{G}^o) = D(\mathscr{G}^o) \left(\underbrace{\begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix}}_{\hat{\mathbf{s}}_1} \mathbf{W}_{12} \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix}}_{\hat{\mathbf{s}}'_1} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}}_{\hat{\mathbf{s}}_2} \mathbf{W}_{13} \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\hat{\mathbf{s}}'_1} \right) D'(\mathscr{G}^o)$$

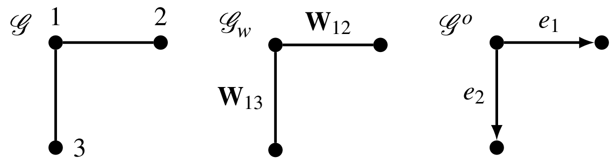


Figure 2.1.: Simple Graph in Example 2.1.

$$
\begin{aligned}
D(\mathscr{G}^o)\mathbf{W}D'(\mathscr{G}^o) &= \underbrace{\begin{bmatrix}-\mathbf{I}\\ \mathbf{I}\\ \mathbf{0}\end{bmatrix}}_{\mathbf{s}_2-\mathbf{s}_1} \mathbf{W}_{12} \underbrace{\begin{bmatrix}-\mathbf{I} & \mathbf{I} & \mathbf{0}\end{bmatrix}}_{\mathbf{s}'_2-\mathbf{s}'_1} + \underbrace{\begin{bmatrix}-\mathbf{I}\\ \mathbf{0}\\ \mathbf{I}\end{bmatrix}}_{\mathbf{s}_3-\mathbf{s}_1} \mathbf{W}_{13} \underbrace{\begin{bmatrix}-\mathbf{I} & \mathbf{0} & \mathbf{I}\end{bmatrix}}_{\mathbf{s}'_3-\mathbf{s}'_1} \\
&= \Bigg(\begin{bmatrix}-\mathbf{I}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix} \mathbf{W}_{12} \begin{bmatrix}-\mathbf{I} & \mathbf{0} & \mathbf{0}\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ \mathbf{I}\\ \mathbf{0}\end{bmatrix} \mathbf{W}_{12} \begin{bmatrix}\mathbf{0} & \mathbf{I} & \mathbf{0}\end{bmatrix} + \dots \\
&\qquad \begin{bmatrix}-\mathbf{I}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix} \mathbf{W}_{12} \begin{bmatrix}\mathbf{0} & \mathbf{I} & \mathbf{0}\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ \mathbf{I}\\ \mathbf{0}\end{bmatrix} \mathbf{W}_{12} \begin{bmatrix}-\mathbf{I} & \mathbf{0} & \mathbf{0}\end{bmatrix} \Bigg) + \dots \\
&\qquad\quad \Bigg(\begin{bmatrix}-\mathbf{I}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix} \mathbf{W}_{13} \begin{bmatrix}-\mathbf{I} & \mathbf{0} & \mathbf{0}\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\\ \mathbf{I}\end{bmatrix} \mathbf{W}_{13} \begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{I}\end{bmatrix} + \dots \\
&\qquad\qquad \begin{bmatrix}-\mathbf{I}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix} \mathbf{W}_{13} \begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{I}\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\\ \mathbf{I}\end{bmatrix} \mathbf{W}_{13} \begin{bmatrix}-\mathbf{I} & \mathbf{0} & \mathbf{0}\end{bmatrix} \Bigg) \\
&= \Bigg(\begin{bmatrix}\mathbf{I}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix} \mathbf{W}_{12} \begin{bmatrix}\mathbf{I} & \mathbf{0} & \mathbf{0}\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ \mathbf{I}\\ \mathbf{0}\end{bmatrix} \mathbf{W}_{12} \begin{bmatrix}\mathbf{0} & \mathbf{I} & \mathbf{0}\end{bmatrix} + \dots \\
&\qquad \begin{bmatrix}\mathbf{I}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix} \mathbf{W}_{13} \begin{bmatrix}\mathbf{I} & \mathbf{0} & \mathbf{0}\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\\ \mathbf{I}\end{bmatrix} \mathbf{W}_{13} \begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{I}\end{bmatrix} \Bigg) + \dots \\
&\qquad\quad - \Bigg(\begin{bmatrix}\mathbf{I}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix} \mathbf{W}_{12} \begin{bmatrix}\mathbf{0} & \mathbf{I} & \mathbf{0}\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ \mathbf{I}\\ \mathbf{0}\end{bmatrix} \mathbf{W}_{12} \begin{bmatrix}\mathbf{I} & \mathbf{0} & \mathbf{0}\end{bmatrix} + \dots \\
&\qquad\qquad \begin{bmatrix}\mathbf{I}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix} \mathbf{W}_{13} \begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{I}\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\\ \mathbf{I}\end{bmatrix} \mathbf{W}_{13} \begin{bmatrix}\mathbf{I} & \mathbf{0} & \mathbf{0}\end{bmatrix} \Bigg)
\end{aligned}
$$

$$D(\mathscr{G}^o)\mathbf{W}D'(\mathscr{G}^o) = \begin{bmatrix} \mathbf{W}_{12}+\mathbf{W}_{13} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_{12} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{W}_{13} \end{bmatrix} - \begin{bmatrix} \mathbf{0} & \mathbf{W}_{12} & \mathbf{W}_{13} \\ \mathbf{W}_{12} & \mathbf{0} & \mathbf{0} \\ \mathbf{W}_{13} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$
$$= \Delta(\mathscr{G}_w) - A(\mathscr{G}_w) = L(\mathscr{G}_w).$$

■

In the case of undirected graphs with scalar weights ($\dim\{\mathscr{G}\} = 1 \iff \mathscr{M} = \mathbb{R}^+$) and without selfloops, Lemma 2.1 is usually given as an alternative definition of the Laplacian matrix. From this property is immediate that when $\mathbf{W}_{i_k j_k} = \mathbf{W}'_{i_k j_k} > 0$, $\forall e_k \in \mathscr{E}$, then the Laplacian matrix is positive semi-definite, *i.e.* all its eigenvalues are real and non negative.

Lemma 2.2. *Let $\mathscr{G}_w$ be a loopless undirected weighted graph,* $\operatorname{rank}\left\{\hat{L}(\mathscr{G}_w)\right\} = (N-1)q$ *if and only if $\mathscr{G}_w$ is connected.*

Proof. As $\hat{L}(\mathscr{G}_w) = D(\mathscr{G}^o)\mathbf{W}D'(\mathscr{G}^o)$ and $\operatorname{rank}\{\mathbf{W}\} = |\mathscr{E}|q$, then $\operatorname{rank}\left\{\hat{L}(\mathscr{G}_w)\right\} = \operatorname{rank}\{D(\mathscr{G}^o)\}$. Note that $D'(\mathscr{G}^o)\mathbf{1} = \mathbf{0}$ which implies that the columns of $\mathbf{1} \in \mathbb{R}^{Nq\times q}$ are vectors in the null space of $D'(\mathscr{G}^o)$ and therefore $\operatorname{rank}\{D(\mathscr{G}^o)\} \leqslant (N-1)q$. From here, two cases can be distinguished:

- If the graph is not connected, then there is at least one pair of nodes, i and j, between which there is no path. Let $\mathscr{C}_i \subset \mathscr{V}$ be the set of all nodes that are connected with i (excluding the node i itself), and therefore not connected to j, then $D'(\mathscr{G}^o)(\mathbf{s}_i + \sum_{l\in\mathscr{C}_i}\mathbf{s}_l) = \mathbf{0}$ and therefore the columns of $(\mathbf{s}_i + \sum_{l\in\mathscr{C}_i}\mathbf{s}_l) \in \mathbb{R}^{Nq\times q}$ are also, along with $\mathbf{1}$, vectors in the null space of $D'(\mathscr{G}^o)$. This shows that for the unconnected case, $\operatorname{rank}\{D(\mathscr{G}^o)\} \leqslant (N-2)q$.
- If the graph is connected, then the q columns of $\mathbf{s}_i + \sum_{l\in\mathscr{C}_i}\mathbf{s}_l = \mathbf{1}$ are the only vectors in the null space of $D'(\mathscr{G}^o)$, implying that $\operatorname{rank}\{D'(\mathscr{G}^o)\} = (N-1)q$.

□

From this result, if the eigenvalues of the Laplacian matrix of an undirected weighted graph are ordered in an increasing order, it is clear that the first q of them are identically zero.

Definition 2.1.13. Let $\mathscr{G}_w$ be a weighted undirected loopless graph of $\dim\{\mathscr{G}_w\} = q$, the $(q+1)$-th element of the increasing ordered set $\operatorname{eig}\left\{\hat{L}(\mathscr{G}_w)\right\}$ is the *algebraic connectivity* of the graph denoted $a(\mathscr{G}_w)$.

$$\operatorname{eig}\left\{\hat{L}(\mathscr{G}_w)\right\} = \{\lambda_1 = 0, \lambda_2 = 0, \cdots, \lambda_q = 0, a(\mathscr{G}_w) := \lambda_{q+1}, \lambda_{q+2}, \cdots, \lambda_{Nq}\}$$

The algebraic connectivity is a measure of how well a graph is connected. If it is zero, then the graph is not connected. A useful study on the matter is the paper [127]. The following result is modified from [119].

Lemma 2.3 ([119]). *If $\mathscr{G}_1 = (\mathscr{V}, \mathscr{E}_1)$ and $\mathscr{G}_2 = (\mathscr{V}, \mathscr{E}_2)$ are two undirected unweighted loopless graphs such that $\mathscr{E}_1 \cap \mathscr{E}_2 = \{\}$, then $a(\mathscr{G}_1) + a(\mathscr{G}_2) \leqslant a(\mathscr{G}_1 \oplus \mathscr{G}_2)$, where $\mathscr{G}_1 \oplus \mathscr{G}_2 = (\mathscr{V}, \mathscr{E}_1 \cup \mathscr{E}_2)$.*

Proof. Define the set $W = \left\{\mathbf{x} \in \mathbb{R}^{Nq} | \|\mathbf{x}\| = 1 \ \wedge \ \mathbf{x}'\mathbf{1} = \mathbf{0}\right\}$. As $\mathscr{E}_1 \cap \mathscr{E}_2 = \{\}$, $\hat{L}(\mathscr{G}_1 \oplus \mathscr{G}_2) = \hat{L}(\mathscr{G}_1) + \hat{L}(\mathscr{G}_2)$. Thus,

$$\begin{aligned} a(\mathscr{G}_1 \oplus \mathscr{G}_2) &= \min_{\mathbf{x} \in W} \left\{\mathbf{x}'\hat{L}(\mathscr{G}_1)\mathbf{x} + \mathbf{x}'\hat{L}(\mathscr{G}_2)\mathbf{x}\right\} \\ &\geqslant \min_{\mathbf{x} \in W} \left\{\mathbf{x}'\hat{L}(\mathscr{G}_1)\mathbf{x}\right\} + \min_{\mathbf{x} \in W} \left\{\mathbf{x}'\hat{L}(\mathscr{G}_2)\mathbf{x}\right\} \\ &= a(\mathscr{G}_1) + a(\mathscr{G}_2). \end{aligned}$$

□

Corollary 2.4. *Given a connected undirected unweighted loopless graph $\mathscr{G}_1 = (\mathscr{V}, \mathscr{E})$ with $|\mathscr{E}| < N(N-1)$ edges, adding an additional edge $e_{|\mathscr{E}|+1} \notin \mathscr{E}$ results in a graph $\mathscr{G}_2 = (\mathscr{V}, \mathscr{E} \cup \{e_{|\mathscr{E}|+1}\})$ with an algebraic connectivity larger than that of the $\mathscr{G}_1$. i.e., $a(\mathscr{G}_1) < a(\mathscr{G}_2)$. The algebraic connectivity is maximum when the graph is fully connected. In that case, $\hat{L}(\mathscr{G}_w) = N\mathbf{I} - \mathbf{1}\mathbf{1}'$ and $a(\mathscr{G}_w) = N$.*

Note that a similar result can be proposed for undirected weighted graphs when the weights associated with the possible edges of the graphs are fixed.

2.2. Stability of Systems

2.2.1. Lyapunov's Stability

Lyapunov's Theory on stability of dynamical systems is fundamental to understand the contributions of this work. Nowadays the so called *Lyapunov's Second Method* is standard in control theory and can be easily found in non linear control books as [69, 74]. These concepts were developed by Aleksandr Mikhailovich Lyapunov in his doctoral dissertation in 1892. They were translated to French and immediately attracted the attention of the scientific community. However, these results were long forgotten by the western scientific community until the mid 1950's when researchers as R. E. Kalman [67, 68] and J. P. LaSalle [70, 71] drew attention to them. In the centenary of its first publication, the *International Journal of Control* republished, in English, Lyapunov's Doctoral thesis in a special issue of the journal [72].

In this section we state the main definitions and results of Lyapunov's theory without proofs. The following definitions and theorems are mainly taken from [67] and [69] with some minor changes in notation.

Definition 2.2.1. A *non autonomous continuous time system* is such that:

$$\dot{\mathbf{x}} = f(t, \mathbf{x}) \tag{2.1}$$

where $f : [0,\infty) \times D \to \mathbb{R}^n$ is piecewise continuous in t and locally Lipschitz in $\mathbf{x}$ on $[0,\infty) \times D$, and $D \subset \mathbb{R}^n$ is a domain that contains the origin.

Definition 2.2.2. The origin $\mathbf{x}^* = \mathbf{0}$ is an equilibrium point of (2.1) if and only if $\forall t \geqslant 0$,

$$\mathbf{x}^* = f(t,\mathbf{x}^*).$$

Additionally, it is

- *stable* if, for each $\varepsilon > 0$, there is $\delta = \delta(\varepsilon,t_0) > 0$ such that

$$||\mathbf{x}(t_0)|| < \delta \Rightarrow ||\mathbf{x}(t)|| < \varepsilon, \forall t \geqslant t_0 \geqslant 0.$$

- *uniformly stable* if, for each $\varepsilon > 0$, there is $\delta = \delta(\varepsilon) > 0$, independent of t_0, such that the previous stability condition is fulfilled.
- *unstable* if not stable.
- *asymptotically stable* if it is stable and there is $c = c(t_0) > 0$ such that $\mathbf{x}(t) \to \mathbf{0}$ as $t \to \infty$, for all $||\mathbf{x}(t_0)|| < c$.
- *uniformly asymptotically stable* if it is uniformly stable and there is $c > 0$, independent of t_0, such that for all $||\mathbf{x}(t_0)|| < c$, $\mathbf{x}(t) \to \mathbf{0}$ as $t \to \infty$, uniformly in t_0; that is, for each $\varepsilon > 0$, there is $T = T(\varepsilon) > 0$ such that

$$||\mathbf{x}(t)|| < \varepsilon, \forall t \geqslant t_0 + T(\varepsilon), \forall ||\mathbf{x}(t_0)|| < c.$$

- *globally uniformly asymptotically stable* if it is uniformly stable and, for each pair of positive numbers ε and c, there is $T = T(\varepsilon,c) > 0$ such that

$$||\mathbf{x}(t)|| < \varepsilon, \forall t \geqslant t_0 + T(\varepsilon,c), \forall ||\mathbf{x}(t_0)|| < c.$$

For simplicity of language, in many cases we will describe system (2.1) as *stable*, if the origin is a stable equilibrium point. Analogous for the other cases in Definition 2.2.2.

Definition 2.2.3. A function $v : \mathbb{R}^n \to \mathbb{R}$ is *positive definite* iff $v(\mathbf{0}) = 0$ and $v(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathbb{R}^n \backslash \{0\}$. Analogously, it is *negative definite* when the inequality is reversed, and *semidefinite* when weak inequalities ($\geqslant$ or $\leqslant$) are used.

Lyapunov's Second Method for non autonomous systems is described by the following theorem:

Theorem 2.5 (Non autonomous Lyapunov)**.** *Let* $\mathbf{x}^* = \mathbf{0}$ *be an equilibrium point for (2.1) and* $D \subset \mathbb{R}^n$ *be a domain containing the origin. Let* $v : [0,\infty) \times D \to \mathbb{R}$ *be a continuous function such that* $\forall t \geqslant 0$ *and* $\forall \mathbf{x} \in D$*,*

$$w_1(\mathbf{x}) \leqslant v(t,\mathbf{x}) \leqslant w_2(\mathbf{x}), \tag{2.2}$$

$$\frac{d}{dt} v(t,\mathbf{x}) \leqslant -w_3(\mathbf{x}), \tag{2.3}$$

where $w_1(\mathbf{x})$*,* $w_2(\mathbf{x})$ *and* $w_3(\mathbf{x})$*, are continuous positive definite functions on D, and* $v(t,\mathbf{0}) = 0$*, then* $\mathbf{x}^* = \mathbf{0}$ *is uniformly asymptotically stable.*

Corollary 2.6 (Global Nonantonomous Lyapunov)**.** *Suppose that all the assumptions of Theorem 2.5 are satisfied globally (for all* $\mathbf{x} \in \mathbb{R}^n$*) and* $w_1(\mathbf{x})$ *is radially unbounded, i.e.* $\|\mathbf{x}\| \to \infty \implies w_1(\mathbf{x}) \to \infty$*, then* $\mathbf{x}^* = \mathbf{0}$ *is globally uniformly asymptotically stable.*

Corollary 2.7 (Exponential Nonantonomous Lyapunov)**.** *Suppose that all assumptions of Theorem 2.5 are satisfied with*

$$w_1(\mathbf{x}) \geqslant k_1||\mathbf{x}||^c, \;\; w_2(\mathbf{x}) \leqslant k_2||\mathbf{x}||^c, \;\; w_3(\mathbf{x}) \geqslant k_3||\mathbf{x}||^c.$$

for some positive constants k_1*,* k_2*,* k_3 *and c, then* $\mathbf{x}^* = \mathbf{0}$ *is exponentially stable. Moreover, if the assumptions hold globally, then* $\mathbf{x}^* = \mathbf{0}$ *is globally exponentially stable.*

The previous theorems can be relaxed for the case of autonomous systems in the form

$$\dot{\mathbf{x}} = f(\mathbf{x}). \tag{2.4}$$

That is, by considering Lyapunov functions that depend only on $\mathbf{x}$ and not explicitly on time. In that case, the theorems are simplified as there is no need to use the auxiliary autonomous functions w_1, w_2 and w_3.

Corollary 2.8 (Autonomous Lyapunov)**.** *Let* $\mathbf{x}^* = \mathbf{0}$ *be an equilibrium point for (2.4). Let* $v : \mathbb{R}^n \to \mathbb{R}$ *be a continuous and radially unbounded positive definite function, such that* $\forall \mathbf{x} \neq \mathbf{0}$*,*

$$\dot{v}(\mathbf{x}) < 0,$$

then $\mathbf{x}^* = \mathbf{0}$ *is globally asymptotically stable.*

Lyapunov's theory has its parallel to discrete time systems in the form of

$$\mathbf{x}(k+1) = h(k,\mathbf{x}(k)), \tag{2.5}$$

where $h : \mathbb{N}_0 \times D \to \mathbb{R}^n$ is piecewise continuous in $\mathbf{x} \in D$. See, for example, [68]. All other definitions in the discrete time case are identical as in the continuous case considering the discrete time variable k instead of the continuous t. Stability for this kind of systems can be verified with the help of the following theorem.

Theorem 2.9 (Discrete Lyapunov). *Let $\mathbf{x}^* = \mathbf{0}$ be an equilibrium point for (2.5) and let $v : \mathbb{N}_0 \times \mathbb{R}^n \to \mathbb{R}$ be a continuous function in $\mathbf{x} \in D$ such that $\forall k \in \mathbb{N}_0$ and $\forall \mathbf{x} \in \mathbb{R}^n$,*

$$w_1(\mathbf{x}(k)) \leqslant v(k, \mathbf{x}(k)) \leqslant w_2(\mathbf{x}(k)), \tag{2.6}$$

$$\Delta v_k := v(k+1, \mathbf{x}(k+1)) - v(k, \mathbf{x}(k)) \leqslant -w_3(\mathbf{x}(k)), \tag{2.7}$$

where $w_1(\mathbf{x})$, $w_2(\mathbf{x})$ and $w_3(\mathbf{x})$, are continuous positive definite functions on $\mathbb{R}^n$, $w_1(\mathbf{x})$ is radially unbounded, and $v(k, \mathbf{0}) = 0$. Then, $\mathbf{x}^* = \mathbf{0}$ is globally uniformly asymptotically stable. Furthermore, if

$$w_1(\mathbf{x}) \geqslant k_1||\mathbf{x}||^c, \;\; w_2(\mathbf{x}) \leqslant k_2||\mathbf{x}||^c, \;\; w_3(\mathbf{x}) \geqslant k_3||\mathbf{x}||^c,$$

for some positive constants k_1, k_2, k_3 and c, then $\mathbf{x}^* = \mathbf{0}$ is globally exponentially stable.

2.2.2. Linear Matrix Inequalities

Linear Matrix Inequalities (LMIs) are an active research topic in control theory and a powerful tool for the analysis of system stability. During the last decades LMI techniques have been successfully applied in many control problems including, for example, dynamic feedback design, uncertainty analysis and robust control design. The book [80] is often quoted as the basic reference in the field. Immediately after the publication of this book, a Matlab® Toolbox for solving LMI problems was published [88]. The book [84] gives an interesting insight on different implementation and application aspects from both, control and optimization, perspectives. Other introductory references on the matter are [86, 87, 98, 101, 108, 109, 113]. Additionally, many interesting results and a broad bibliographic review can be found in the course material [103] which is available online (in Portuguese).

In this work, matrix inequalities will be used to indicate that a certain matrix is positive (or negative) definite (or semi-definite). The expression *linear matrix inequality* refers to a matrix inequality where all variables are linear with respect to each other. LMIs are convex optimization problems and therefore they can be numerically solved with the help of several available software. During the 1980's many algorithms with guaranteed global convergence were developed under what is now known as semi-definite programming. Because of this, in practice, to formulate a problem in terms of LMIs is sufficient to compute a numeric solution. In this work we prefer the solver SeDuMi [111] parsed by YALMIP [96] which have become the *de facto* standard in the area. Definiteness of matrices is a property defined in the following way.

Definition 2.2.4. A matrix $\mathbf{M} = \mathbf{M}' \in \mathbb{R}^{n \times n}$ is *positive definite*, denoted $\mathbf{M} > 0$, if and only if its quadratic form is *positive definite*. That is, if $v(\mathbf{x}) = \mathbf{x}'\mathbf{M}\mathbf{x} > 0$, $\forall \mathbf{x} \neq \mathbf{0}$. It is *positive semidefinite*, denoted $\mathbf{M} \geqslant 0$, if $v(\mathbf{x}) = \mathbf{x}'\mathbf{M}\mathbf{x} \geqslant 0$, $\forall \mathbf{x} \neq \mathbf{0}$ and *negative (semi) definite*, denoted $\mathbf{M} < 0$ (respectively $\mathbf{M} \leqslant 0$), if $-\mathbf{M} > 0$ ($-\mathbf{M} \geqslant 0$).

Positive (semi) definite matrices have only positive (non negative) real eigenvalues and can be decomposed into $\mathbf{M} = \mathbf{N}'\mathbf{N}$ where $\mathbf{N} \in \mathbb{R}^{n\times n}$.

The development of LMIs is strongly related to Lyapunov's theory. Indeed, a very simple LMI is obtained from applying Lyapunov's second method to the linear time invariant system:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} \tag{2.8}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$. It is a well known fact that a linear system in this shape is asymptotically stable if and only if matrix $\mathbf{A}$ is Hurwitz, *i.e.* if all its eigenvalues have negative real parts. This condition can be equivalently expressed as an LMI.

Theorem 2.10. *System (2.8) is asymptotically stable, if and only if it exists* $\mathbf{P} > 0$ *such that*

$$\mathbf{P}\mathbf{A} + \mathbf{A}'\mathbf{P} < 0. \tag{2.9}$$

Proof. Consider the autonomous quadratic Lyapunov function $v(\mathbf{x}) = \mathbf{x}'\mathbf{P}\mathbf{x}$ with $\mathbf{P} = \mathbf{P}' > 0$. It can be easily shown that this function fulfills all the assumptions of Corollary 2.8. Therefore if $\dot{v}(\mathbf{x}) = \mathbf{x}'\mathbf{P}\dot{\mathbf{x}} + \dot{\mathbf{x}}'\mathbf{P}\mathbf{x} = \mathbf{x}'\left(\mathbf{P}\mathbf{A} + \mathbf{A}'\mathbf{P}\right)\mathbf{x} < 0$, the origin is asymptotically stable and $\mathbf{A}$ is Hurwitz. As this condition has to be fulfilled for all $\mathbf{x} \in \mathbb{R}^n$, this leads to LMI (2.9). Furthermore, if $\mathbf{A}$ is Hurwitz, then $\mathbf{P} = \int_0^{+\infty} e^{\mathbf{A}'t}e^{\mathbf{A}t}dt$ satisfies (2.9):

$$\begin{aligned}
\mathbf{P}\mathbf{A} + \mathbf{A}'\mathbf{P} &= \int_0^{+\infty}\left(e^{\mathbf{A}'t}e^{\mathbf{A}t}\mathbf{A} + \mathbf{A}'e^{\mathbf{A}'t}e^{\mathbf{A}t}\right)dt \\
&= \int_0^{+\infty}\left(\frac{d}{dt}e^{\mathbf{A}'t}e^{\mathbf{A}t}\right)dt \\
&= e^{\mathbf{A}'t}e^{\mathbf{A}t}\Big|_0^{+\infty} = -\mathbf{I} < 0
\end{aligned}$$

□

Remark 2.1. Note that if λ is an eigenvalue of $\mathbf{A}$, then $\lambda + \delta$, with $\delta \in \mathbb{R}$, is an eigenvalue of $\mathbf{A} + \delta\mathbf{I}$. From here, $\mathbf{P}(\mathbf{A} + \delta\mathbf{I}) + (\mathbf{A}' + \delta\mathbf{I})\mathbf{P} < 0$ if and only if the eigenvalues of $\mathbf{A}$ have real parts strictly smaller than $-\delta$. Furthermore, $\mathbf{P}(\mathbf{A} + \delta\mathbf{I}) + (\mathbf{A}' + \delta\mathbf{I})\mathbf{P} \leqslant 0$ if the real parts of the eigenvalues of $\mathbf{A}$ are smaller or equal than $-\delta$.

Similar, from the discrete Lyapunov theorem, a discrete time linear system $\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k)$, with $k \in \mathbb{N}_0$, is stable if there exist $v(\mathbf{x}) = \mathbf{x}'\mathbf{P}\mathbf{x} > 0$ such that $\forall k \in \mathbb{N}_0$, $\Delta v_k := v(\mathbf{x}(k+1)) - v(\mathbf{x}(k)) = \mathbf{x}'(k)\left(\mathbf{A}'\mathbf{P}\mathbf{A} - \mathbf{P}\right)\mathbf{x}(k) < 0$. Which is equivalent to study the feasibility of $\mathbf{A}'\mathbf{P}\mathbf{A} - \mathbf{P} < 0$ with $\mathbf{P} > 0$. A stable discrete time system has all its eigenvalues in the unitary circle. If that is the case, then $\mathbf{P} = \sum_{k=0}^{\infty}(\mathbf{A}')^k\mathbf{A}^k$ fulfills the inequality.

These results, which are modified versions of Lyapunov's original statements in his thesis, are important because they show the basic procedure followed by most applications of LMIs to control theory. That is, to propose a Lyapunov function to check stability and from there develop LMI conditions that can be numerically verified. Other LMI related results useful to this thesis can be found in Appendix A.4.

Some remarks on the numeric solution of LMI need to be taken into account. It is sometimes important to count the number of "scalar variables" that defines the problem. A scalar variable is a variable entry in a "matrix variable". A full variable matrix $\mathbf{X} \in \mathbb{R}^{n\times m}$ is composed of $n \times m$ scalar variables. A symmetric matrix $\mathbf{P} \in \mathbb{R}^{n\times n}$ (typically referred to as *Lyapunov Matrix*) has $(n^2 - n)/2 + n = (n+1)n/2$ variables. If more demanding structural restrictions on the matrix variables are imposed, the number of variables can be diminished. For example, a diagonal matrix $\mathbf{D} \in \mathbb{R}^{n\times n}$ has only n variables.

In general, any available LMI software is capable of solving most of common problems (feasibility or optimization) with acceptable efficiencies in any standard software/hardware configuration. For example, all numeric evaluations of LMI in this thesis are done with a Dell notebook with an Intel Core i5 CPU at 2.50GHz and with 6.00GB RAM, over Matlab R2011b on Windows 7 of 64-bits. However, the complexity of the restrictions have a direct impact on the time needed to compute a solution and on its accuracy.

By complexity we refer mainly to 1) number of scalar variables, 2) number of inequalities, and 3) dimensions of the involved matrices that define the inequalities. Of course, these three elements are related to each other as, for example, larger matrices will imply larger numbers of variables and more inequalities will usually imply also more matrix variables. A fourth component that determines how fast a problem can be solved is related to "how large" is the possible solutions region. If the solutions region of a given problem is too "narrow", in general it is difficult to reach it and therefore it takes longer to solve the problem. Therefore, minimization problems are in general more demanding than feasibility problems. Furthermore, having "too many" restrictions, which directly impacts the number of variables, is also often impractical. However, the performance of any algorithm can be improved by more powerful hardware configurations and better software implementations. Typically, all LMI computations in this thesis are in the range of fractions of seconds to seconds.

3 Stability of Switched Systems

Hybrid Systems are a very active research area where many disciplines merge in the study of complex dynamical behaviors with continuous and discrete states, *e.g.* [153]. Switched linear systems can be interpreted as a special case of hybrid systems. They have been widely studied and a key reference is the book [151]. Many results rely on the finding of Lyapunov functions to ensure stability. A good summary of this idea may be found in survey papers such as [137, 152, 157], which give a wide spectrum of the topic.

The study of switched systems typically makes the differences between continuous and discrete time. Some examples for the continuous case are [133, 141], where conditions for global asymptotic stability are developed considering dwell time. In [158], stability for a particular case with two discrete states is studied, while in [154] the problem of continuous state feedback and pole allocation is addressed by imposing a common Lyapunov function to every discrete mode. Piecewise Continuous Lyapunov functionals have also been proposed, some examples are [132, 137, 161, 163, 166, 167]. This approach suggests a relationship between continuous and discrete time switched systems as it associates different Lyapunov functions to the time intervals corresponding to each mode.

In discrete time, besides the case where switching is arbitrary [77], we can distinguish when the switching sequences are known and fixed, and the case where only jumping probabilities are known. In the first case the stability analysis is done either by computing the spectral radius of a matrix that represents the whole cycle of the system, *e.g.* [152], or by using Lyapunov functions as in [136, 140, 170]. For the probabilistic case, strongly related with Markov Chains Theory, the definition of stability needs to be modified in order to consider the stochastic nature of the switching mechanisms. However, similar analysis tools can be used. Some examples are the recent survey paper [156] and other references such as [142, 148, 149, 162, 164, 168]. Because of the formality and completeness of the analysis, the paper [135] and the book [134] of the same authors deserve special attention. A more general approach is followed in publications such as [143–145, 150, 176, 159, 160, 165, 169, 171], where stochastic stability is studied through the expected behavior of a Lyapunov function. The analysis of Markov jumping systems have been also extended to the continuous case, see for example [139, 146, 147, 156, 168].

In this chapter an application of Lyapunov's Second Method and LMIs to switched systems is presented. We represent a continuous time switched system by a discrete time equivalent system and from there we deriver stability conditions based on discrete time Lyapunov conditions. We consider the case where the jumps between discrete modes are deterministic and where they are the result of a Markov stochastic process. Most of the results presented here can be found in the quoted references on discrete time switched systems with exception of the result that consider robustness against parametric uncertainties at the end of the chapter.

3.1. Preliminary Definitions

3.1.1. Switched Systems

A switched system can be defined in the following way.

Definition 3.1.1. A *switched linear system* has discrete mode dynamics given by a relation of time $q : \mathbb{R}^+ \mapsto Q = \{q_1, q_2, ..., q_M\}$, and continuous dynamics such that

$$\dot{\mathbf{x}} = f(q_i, \mathbf{x}) = \mathbf{A}_i \mathbf{x} \tag{3.1}$$

where $\mathbf{A}_i \in \mathbb{R}^{n \times n}$, $i \in \{1, 2, ...M\}$. The possible transitions between discrete modes are modeled by a set of edges $E = \{e_{ij} = (q_i, q_j) | q_i, q_j \in Q \ \wedge \ q_i \neq q_j\} \subseteq Q \times Q$. These transitions occur at *switching instants* τ_k with $k \in \mathbb{N}$. $\mathbf{x}$ is continuous at the switching instants.

Definition 3.1.2. An *Infinite Switching Time Set* is an infinite sequence of switching instants: $S_\infty = \{\tau_0, \tau_1, ..., \tau_k, ...\}$.

We assume that for some sufficiently small $\delta > 0$, $\delta < \tau_{k+1} - \tau_k$ so that within any finite time interval, there is only a finite number of switching instants. Note that in this description of switched systems, the value k is associated with the jumps as a function of time. In other words, k is not a relation of the continuous states, nor of the current or future discrete modes. E may be a strict subset of $Q \times Q$ so that certain discrete modes transitions may be forbidden.

A first case of interest is when arbitrary switching sequences are allowed. Here, asymptotic stability can be defined as the asymptotic stability of the steady state $\mathbf{x}_s = \mathbf{0}$ for any switching signal q allowed by E. It is trivial to show that a necessary condition is that all $\mathbf{A}_i$ are Hurwitz. If that is not the case for an $\mathbf{A}_j$, we can choose $q = q_j = const$ to show that the switched system is not asymptotic stable for every switching signal. In the references, *e.g.* [151], a decreasing common Lyapunov function $v(\mathbf{x}) = \mathbf{x}'\mathbf{P}\mathbf{x}$ for all modes is shown as a sufficient condition for the stability of the switched system. The author has extended this approach in [155] to a more general class of Lyapunov functions derived from *Homogeneous Polynomials*.

Because of space limitations and relevancy to the rest of the thesis, in this chapter we will not discuss the arbitrary switching case. We will concentrate on a second class of switching structure driven solely by time dependent restrictions.

Stability under time dependent switching is defined as the (global uniform asymptotic) stability of the steady state $\mathbf{x}_s = \mathbf{0}$ for any switching signal q allowed by E over an infinite switching time set. The trivial case when the switching time set is finite is not of interest as then, after a successive chain of discrete jumps, the switched system rests in a final mode where the continuous states evolve. It is clear that if this final mode is stable, then every possible execution of the system will be stable and there is no need of further analysis. On the contrary, if the system has the ability of switching endlessly, then stability must be studied considering all infinite possible jumps.

3.1.2. Polytopic Systems

Since the beginning of the last decade, many results have been published on polytopic systems. See for example [82, 90, 91, 95, 105], etc. This kind of systems is often used to model uncertain linear systems or time variant systems with smooth parametric changes. Different approaches with LMIs as the main analysis tool have been proposed to characterize stability through quadratic Lyapunov functions.

Definition 3.1.3. The *Unit Simplex* of $\mathbb{R}^M$ is the set

$$\Lambda_M = \left\{ \boldsymbol{\alpha} \in \mathbb{R}^M | \sum_{i=1}^{M} \alpha_i = 1 \ \wedge \ \alpha_i \geqslant 0, \ \ \forall i \in \{1,2,...,M\} \right\},$$

where α_i is the i-th element of the *parametrization vector* $\boldsymbol{\alpha}$.

Definition 3.1.4. A *polytopic system* in continuous time is such that

$$\dot{\mathbf{x}} = \mathbf{A}(\boldsymbol{\alpha})\mathbf{x}, \tag{3.2}$$

where $\boldsymbol{\alpha} \in \Lambda_M$ and $\mathbf{A}(\boldsymbol{\alpha}) = \sum_{i=1}^{M} \alpha_i \mathbf{A}_i$ with known matrices $\mathbf{A}_i \in \mathbb{R}^{n\times n}$, $i = 1,...,M$, called *vertices*.

The parametrization vector $\boldsymbol{\alpha}$ can be an unknown constant $\boldsymbol{\alpha} \in \Lambda_M$, in which case the polytopic structure is commonly used to model uncertainties in the parameters of a linear system; or a function of time $\boldsymbol{\alpha}(t) : \mathbb{R}^+ \to \Lambda_M$. Stability of systems described in this way has been widely studied and the complexity of the problem increases when $\boldsymbol{\alpha}$ is time dependent.

To investigate whether there are conditions that assure that the system is asymptotically stable regardless of the unknown value of $\boldsymbol{\alpha}$, only by considering the known information on the vertices, the existence of a common quadratic Lyapunov function ($v = \mathbf{x}'\mathbf{P}\mathbf{x}$) for all vertices is a sufficient condition. This situation might however be too restrictive as it imposes a very particular kind of stability condition that has to be fulfilled by all the vertices. This is why other Lyapunov functions are proposed. For example [105, 106] propose the use of a Lyapunov matrix which is a linear function of the parametrization vector.

Definition 3.1.5. A *Linear Polynomial Lyapunov Function* is such that

$$v = \mathbf{x}'\mathbf{P}(\boldsymbol{\alpha})\mathbf{x}, \tag{3.3}$$

where

$$\mathbf{P}(\boldsymbol{\alpha}) = \sum_{i=1}^{M} \alpha_i \mathbf{P}_i,$$

$\boldsymbol{\alpha} \in \Lambda_M$, and $\mathbf{P}_i = \mathbf{P}_i' > 0$.

In this case the original common stability condition is relaxed by considering a functional that also depends in a linear way on the unitary simplex Λ_M. Note that common quadratic Lyapunov stability is a special case of (3.3) when $\mathbf{P}_i = \mathbf{P}$, $\forall i \in \{1,\ldots,M\}$.

Further assumptions about the structure of the Lyapunov function can be made. Particularly, *homogeneous matrix polynomials* can be used to test stability. The use of such Lyapunov functions has been documented in, for example, [78, 79, 81, 82, 99, 100, 107]. However, for our objectives, we will not need these more complex Lyapunov functions.

3.1.3. Polytopic Approximation of a Switched System

In the switched linear system (3.1), each of the matrices $\mathbf{A}_i$ can be interpreted as a vertex of the polytopic system described by (3.2). Considering the time dependence of q, if it is imposed that $\forall t : q = q_i \implies \alpha_i(t) > 0$, then the parametrization vector $\boldsymbol{\alpha}$ is a function of time that interprets the evolution of the discrete state q.

In particular, the behavior of a switched linear system can be approximated by using parametrization vectors in the following function class:

$$K_h(\varepsilon) = \left\{ \boldsymbol{\alpha} : \mathbb{R}^+ \to \Lambda_M | \alpha_i(t) = c_i + \sum_{k=1}^{N} n_i(k)\Theta_\varepsilon(t - \tau_k) \right\},$$

with N the number of switching instants considered (that might be infinite), $c_i = 1$ if the i-th mode is active at $t = 0$ and $c_i = 0$ otherwise, and $n_i(k) \in \{-1, 0, 1\}$ selected according to the edge active at τ_k. If the jump is from mode q_i to q_j, then $n_i(k) = -1$; if the jump is from q_j to q_i, then $n_i(k) = 1$; $n_i(k) = 0$ otherwise. $\Theta_\varepsilon(t)$ is an analytical smooth approximation of the Heaviside step function:

$$\Theta(t) = \lim_{\varepsilon \to 0^+} \Theta_\varepsilon(t) = \begin{cases} 0 & , \quad t < 0 \\ 1/2 & , \quad t = 0 \\ 1 & , \quad t > 0 \end{cases}$$

The parameter $\varepsilon > 0$ characterizes the accuracy of the approximation. A possible choice for this function would be $\Theta_\varepsilon(t) = 1/(1 + e^{-2t/\varepsilon})$.

Note that this approximation of a switched system is done through a smooth function $\boldsymbol{\alpha}(t)$ that depends on parameter $\varepsilon > 0$. In the limit $\varepsilon \to 0$, this approximation becomes an exact representation of the switched system.

3.1.4. Discrete Time Representation

A Switched System can be represented as a non-linear discrete time system. As the execution of the system between switching instants is deterministic, there is an algebraic expression for the values of the continuous states in any interval defined between consecutive switching. In this sense, a switched system is similar to an asynchronous sampled-data system and stability can be studied by observing the system at switching instants only.

The transition matrix $\mathbf{\Phi}(t,t_0)$ of system (3.2) is such that $\forall t > t_0$,

$$\frac{d}{dt}\mathbf{\Phi}(t,t_0) = \mathbf{A}(\boldsymbol{\alpha})\mathbf{\Phi}(t,t_0).$$

Note that the transition matrix depends explicitly on the approximation parameter ε when $\boldsymbol{\alpha} \in K_h(\varepsilon)$. Unfortunately, when $\boldsymbol{\alpha} \in K_h(\varepsilon)$ and in the limit $\varepsilon \to 0^+$, as $\mathbf{A}(\boldsymbol{\alpha})$ is a function of time, in general it is not possible to find an expression for the transition matrix. However, between switching instants, the system behaves as a linear system, and therefore, a piece wise expression for the trajectory of the system can be found. From here, the states at switching instants can be found as the states of a time-variant discrete time system.

Indeed, the solution of a linear dynamic system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$ with initial condition $\mathbf{x}(t_0) = \mathbf{x}_0$ is given by $\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{x}_0$. (See Proposition A.11 in the Appendix.) Therefore, we can write the following relationship between switching instants τ_k and τ_{k+1}:

$$\begin{aligned}\mathbf{x}(\tau_{k+1}) &= \lim_{\varepsilon\to 0^+}[\mathbf{\Phi}(\tau_{k+1},\tau_k)]\,\mathbf{x}(\tau_k)\\ &= e^{\mathbf{A}(\boldsymbol{\alpha}_k)T_k}\mathbf{x}(\tau_k)\\ &= \mathbf{\Phi}_k\mathbf{x}(\tau_k),\end{aligned}$$

with residence time $T_k = \tau_{k+1} - \tau_k$ and

$$\boldsymbol{\alpha}_k = \lim_{\varepsilon\to 0^+}\boldsymbol{\alpha}(t_m),$$

for some $t_m \in (\tau_k, \tau_{k+1})$. The vector $\boldsymbol{\alpha}_k$ is a vector in the respective unit simplex with zeros in every element but in the associated with the discrete mode active between τ_k and τ_{k+1}. Note that in general, $\boldsymbol{\alpha}(\tau_k) \neq \boldsymbol{\alpha}_k \neq \boldsymbol{\alpha}(\tau_{k+1})$. The discrete time system matrix is implicitly defined as $\mathbf{\Phi}_k = e^{\mathbf{A}(\boldsymbol{\alpha}_k)T_k}$. As T_k is not constant and $\boldsymbol{\alpha}_k$ changes after every switching instant, the above described discrete time system is not time-invariant. For a given signal q, it is clear that for $h > k \in \mathbb{N}_0$,

$$\mathbf{x}(\tau_h) = \left(\prod_{i=1}^{h-k}\mathbf{\Phi}_{h-i}\right)\mathbf{x}(\tau_k) = (\mathbf{\Phi}_{h-1}\mathbf{\Phi}_{h-2}...\mathbf{\Phi}_k)\,\mathbf{x}(\tau_k). \tag{3.4}$$

We will define for notation simplicity:

$$\mathbf{\Psi}_k^h = \prod_{i=1}^{h-k}\mathbf{\Phi}_{h-i}.$$

Note that this discrete time representation of the switched system could also be done without introducing the polytopic approximation of Section 3.1.3. However, we choose to maintain the polytopic representation in order to describe the Lyapunov functions to be used in the sequel in a more clear way.

3.2. Time Dependent Switching Results

In this section we introduce the key result that allow us to study stability of a continuous time switched system through its discrete time representation. From here we state several known results for the case of deterministic and probabilistic switching and we extend the conditions to consider uncertainty in the residence time at each discrete mode. The results on deterministic switching are equivalent to those to be found in several publications as [136, 140, 170]. The probabilistic results are based greatly in [134, 135].

3.2.1. Sampled Lyapunov Stability Criteria

In most publications (*e.g.* [67–69, 74]), it is usually considered that a Lyapunov function has to have continuous partial derivatives with respect to time and the states. However, as noted by LaSalle in [70, 71], this requirement is mainly for ease of calculation of the time derivative of the Lyapunov function in most practical cases where the function f is "well behaved". It is however, only required for a Lyapunov function in LaSalle's definition to have a "right hand" derivative at any time. That is, the Lyapunov function does not need to be smooth at every time instant.

A different topic is treated in [65, 73], where a methodology based on hybrid (discrete and continuous time) Lyapunov functions is proposed to treat asynchronous sampled time systems. That is, systems that depend continuously on the value of the states sampled at irregular intervals[1]. Taking into account these two ideas, stability of time depending switching systems can be studied by observing their behavior at the switching instant, similarly as what is done in publications such as [132, 137, 161, 163, 166, 167].

Lemma 3.1 (Sampled Lyapunov)**.** *For the switched system (3.1), if there exists an infinite sequence of switching instants $S = \{\tau_{l_1}, \tau_{l_2}, \ldots, \tau_{l_k}, \ldots\} \subseteq S_\infty$ and a quadratic positive definite scalar function $w : \mathbb{R}^n \times \mathbb{R}_0^+ \to \mathbb{R}$, $(\mathbf{x}, t) \mapsto w(\mathbf{x}, t) = \mathbf{x}'(t)\mathbf{P}(t)\mathbf{x}(t)$, with $\mathbf{P}(t) > 0$, $\forall t \geqslant 0$, such that $\forall k \in \mathbb{N}$,*

$$\Delta w_k := w(\mathbf{x}(\tau_{l_{k+1}}), \tau_{l_{k+1}}) - w(\mathbf{x}(\tau_{l_k}), \tau_{l_k}) < 0, \tag{3.5}$$

then the system is asymptotically stable towards the origin.

Proof. For any $\mathbf{x} \neq \mathbf{0}$ that is a solution of (3.1), from the discrete Lyapunov theorem, $\Delta w_k < 0$ implies that the implicit *discrete* time system that results from observing the polytopic system at specified instants, is stable. To prove the stability of the *continuous* time system, the inter sampling behavior needs to be analyzed.

[1]For example, a continuous time system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$ with a feedback law $\mathbf{u}(t) = \mathbf{K}\mathbf{x}(t_k)$, $\forall k \in \mathbb{N}$ that only considers the value of the states at certain instants t_k and keeps the input constant in the inter-sampling interval so that the continuous closed loop dynamics of the system become $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{K}\mathbf{x}(t_k)$.

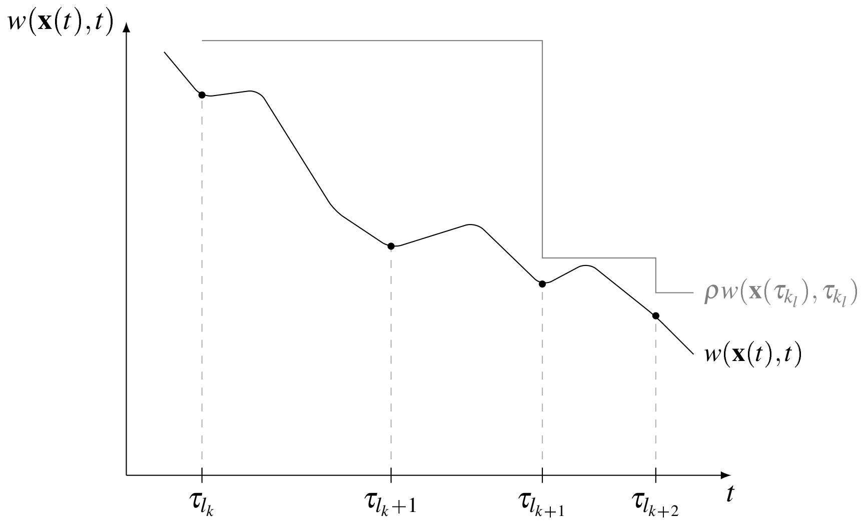


Figure 3.1.: Graphical representation of proof of Lemma 3.1

For all $k \in \mathbb{N}$, define $I_k = [\tau_{l_k}, \tau_{l_{k+1}})$ and $w_{max,k} := \max\{w(\mathbf{x}(t),t) | t \in I_k\}$. Because (3.5) holds, then $w_{max,k}$ happens either at $t_{max,k} = \tau_{l_k}$ or at a time $t_{max,k} \in (\tau_{l_k}, \tau_{l_{k+1}})$. In either case, we can write that

$$\mathbf{x}(t_{max,k}) = \mathbf{\Psi}_{max,k}\mathbf{x}(\tau_{l_k}),$$

where

$$\mathbf{\Psi}_{max,k} = e^{\mathbf{A}(\alpha_{l_k+h})(t_{max,k}-\tau_{l_k+h})}\mathbf{\Psi}_{l_k}^{l_k+h},$$

for some $h \in \mathbb{N}_0$ such that $l_k + h < l_{k+1}$.

Therefore, $w_{max,k} = \mathbf{x}'(\tau_{l_k})\left(\mathbf{\Psi}_{max,k}\right)'\mathbf{P}(t_{max,k})\left(\mathbf{\Psi}_{max,k}\right)\mathbf{x}(\tau_{l_k})$, which is a quadratic form on the vector $\mathbf{x}(\tau_{l_k})$. Then, it always exists a scalar $\rho_k \geqslant 1$ such that $w_{max,k} \leqslant \rho_k w(\mathbf{x}(\tau_{l_k}),\tau_{l_k})$, because $w(\mathbf{x}(\tau_{l_k}),\tau_{l_k}) = \mathbf{x}'(\tau_{l_k})\mathbf{P}(\tau_{l_k})\mathbf{x}(\tau_{l_k})$ is a quadratic form on the same vector and $\mathbf{P}(t) > 0$, $\forall t \in \mathbb{R}$. Defining $\rho \geqslant \sup\{\rho_k | k \in \mathbb{N}\}$, we conclude that $\forall k \in \mathbb{N}$, $\forall t \in I_k$, $w(\mathbf{x},t) \leqslant \rho w(\mathbf{x}(\tau_{l_k}),\tau_{l_k})$. That is, a piece wise constant upper bound for $w(\mathbf{x},t)$ can found and, by hypothesis (3.5), this bound approach to zero as time increases. Note that the bound cannot approach a value different than zero, as that would imply that there are trajectories of the system where (3.5) does not hold.

Asymptotic stability of the continuous time system is then proved by the existence of the Lyapunov function $w(\mathbf{x},t) = \mathbf{x}'(t)\mathbf{P}(t)\mathbf{x}(t)$ which is confined to a region that approximates to zero asymptotically. A graphical sketch of the proof can be seen in Figure 3.1. □

This theorem has the obvious inconvenience that Δw_k needs to be evaluated in an infinite number of instants and therefore it can only be used for systems that present some kind of cyclical behavior. From here on, the task is to find suitable discrete time Lyapunov functions $w(\mathbf{x},t)$ such that the conditions of Lemma 3.1 can be verified, leading to results that are similar to those in the quoted publications.

Example 3.1. One could propose as a counter example for Lemma 3.1 a simple linear system (a trivial case of switched system) with a conjugate pair of unstable eigenvalues. Take for example the linear system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$ where,

$$\mathbf{A} = \begin{bmatrix} -2.00 & 0.00 & 0.00 \\ 0.00 & 0.50 & -6\pi \\ 0.00 & 6\pi & 0.50 \end{bmatrix},$$

with eigenvalues $\lambda \in \{-2.00, 0.50 + j6\pi, 0.50 - j6\pi\}$. The conjugate states associated to the conjugate unstable eigenvalues oscillate at a frequency of $f = 3$[Hz]. The system is clearly unstable as can be seen in the simulation shown in Figure 3.2 a).

It can be then argued that sampling the system at exactly the instants where a non negative function of both unstable conjugate states vanishes, then this information will not be mapped into the Lyapunov functional and therefore, a strictly decreasing discrete Lyapunov function can be found if the other states are stable. This is not the case, because the non negative function that cancels the effect of the conjugated states would only be positive semidefinite and not positive definite as Lyapunov's theory requires.

The quadratic function $w(\mathbf{x},\varepsilon) = \mathbf{x}'\mathbf{P}(\varepsilon)\mathbf{x}$ can be proposed, with a parameter $\varepsilon \geqslant 0$ such that,

$$\mathbf{P}(\varepsilon) = \frac{1}{100}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1-\varepsilon \\ 0 & 1-\varepsilon & 1 \end{bmatrix}.$$

If $\varepsilon = 0$, then $\det\{\mathbf{P}\} = 0$ and the function is only positive semidefinite and therefore cannot be used to prove stability of the system. Indeed, in Figure 3.2 b) the continuous evolution in time of this function is shown. Sampling at a rate of $f_r = 6$[Hz] when the sum of both unstable states vanishes (the marked points in Figure 3.2 a)), leads to a decreasing sequence of values given by the local minima of the function (the red dashed line in the Figure). However, stability cannot be concluded from this sequence. For any $\varepsilon > 0$, $w(\mathbf{x},\varepsilon)$ becomes positive definite, but no infinite decreasing sequence exists. See Figure 3.2 c), where $\varepsilon = 0.1$.

This can be further studied through the discrete time system $\mathbf{x}(t_{k+1}) = e^{\mathbf{A}\frac{1}{f_r}}\mathbf{x}(t_k)$. As some eigenvalues of the discrete system matrix are outside the unitary circle, it can be numerically verified that there does not exist a positive definite matrix $\mathbf{P} > 0$ such that the discrete Lyapunov inequality $(e^{\mathbf{A}\frac{1}{f_r}})'\mathbf{P}e^{\mathbf{A}\frac{1}{f_r}} - \mathbf{P} < 0$ holds. This is not surprising as the matrix exponential

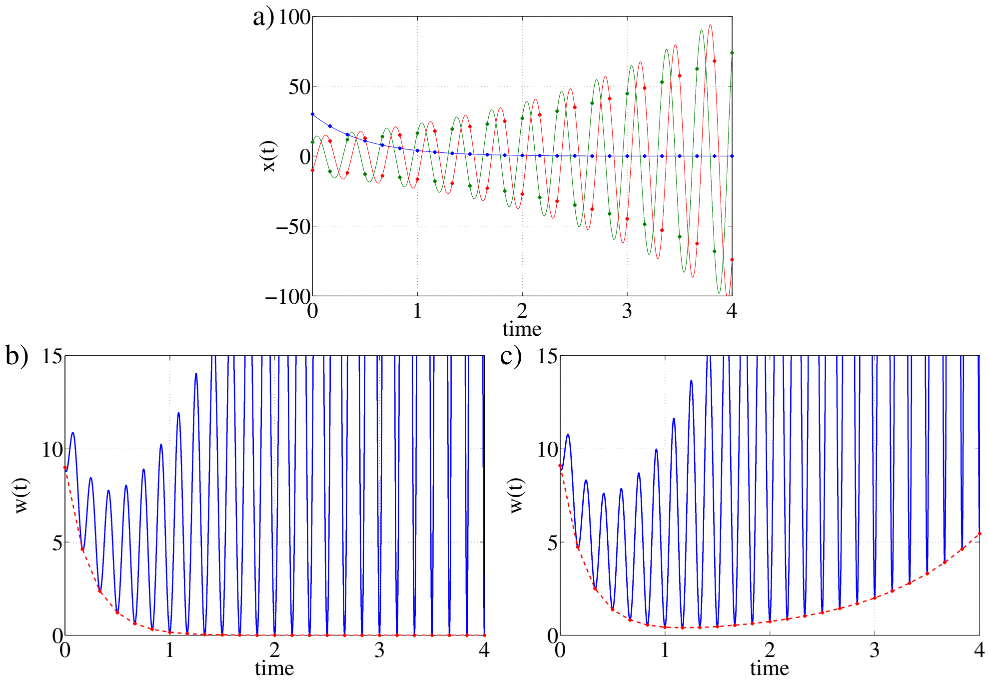


Figure 3.2.: Evolution of a) the states of system, b) a positive semidefinite quadratic function of the states, and c) a positive definite quadratic function of the states in Example 3.1.

operator maps only the stable eigenvalues of matrix $\mathbf{A}\frac{1}{f_r}$ in the unitary circle, leaving the unstable ones out of the discrete stability region and making the inequality unfeasible. Thus, showing that the system is unstable. ■

3.2.2. Deterministic Switching

Common Lyapunov Function

Assume that the switching sequence is deterministic and known. That is, the infinite switching times set $S_\infty = \{\tau_0, \tau_1, \ldots\}$ is known and the discrete modes transitions active at each switching instants are also known. Lemma 3.1 allows us to study stability of a linear switched system only by observing it at the switching instants. Furthermore, it is not required to observe the system at *every* switching instant τ_k but only in an infinite sequence of them. This means that we can arbitrarily skip some switching instants in the stability analysis when the discrete jump to perform at this moment is known. For example, if we know that at τ_l the system jumps from state q_i to state q_j, we can skip τ_l and use only τ_{l-1} and τ_{l+1}, as an expression for the behavior of the continuous states during both time intervals can be found. In this case, a sufficient stability condition is given by the following theorem.

Theorem 3.2 (Discrete Common Lyapunov Function)**.** *Given an infinite subset of the switching instants set, this is $S = \{\tau_{k_0}, \tau_{k_1}, \ldots, \tau_{k_h}, \ldots\} \subseteq S_\infty = \{\tau_0, \tau_1, \ldots\}$, switched system (3.1) is asymptotically stable towards the origin if there exists a symmetric matrix $\mathbf{P} > 0$ such that $\forall h \in \mathbb{N}_0$,*

$$\left(\boldsymbol{\Psi}_{k_h}^{k_{h+1}}\right)' \mathbf{P} \left(\boldsymbol{\Psi}_{k_h}^{k_{h+1}}\right) - \mathbf{P} < 0, \tag{3.6}$$

with $k_{h+1} > k_h \in \mathbb{N}_0$, $\forall h \in \mathbb{N}_0$.

Proof. Consider Lemma 3.1 and a Lyapunov function $w(\mathbf{x}(\tau_k), k) = \mathbf{x}(\tau_k)'\mathbf{P}\mathbf{x}(\tau_k) > 0$, with $\mathbf{P} = \mathbf{P}' > 0$. Then, a sufficient condition for stability of the system (3.2) with $\boldsymbol{\alpha} \in K_h(\varepsilon)$ is:

$$\Delta w_{k_h} = w(\mathbf{x}(\tau_{k_{h+1}})) - w(\mathbf{x}(\tau_{k_h})) < 0.$$

In the limit $\varepsilon \to 0^+$, equation (3.4) leads to

$$\Delta w_{k_h} = \mathbf{x}(\tau_{k_h})' \left[\left(\boldsymbol{\Psi}_{k_h}^{k_{h+1}}\right)' \mathbf{P} \left(\boldsymbol{\Psi}_{k_h}^{k_{h+1}}\right) - \mathbf{P} \right] \mathbf{x}(\tau_{k_h}) < 0.$$

With $\mathbf{x}(\tau_{k_h})$ arbitrary, the previous condition holds if and only if (3.6) is fulfilled $\forall h \in \mathbb{N}_0$. □

This stability condition presents some important restrictions regarding its practical use. The most obvious one is that condition (3.6) must hold for an infinite number of matrices $\boldsymbol{\Psi}_{k_h}^{k_{h+1}}$ as $h \in \mathbb{N}_0$. Therefore, if the switched system does not present a cyclic behavior, where only a limited number of switching sequences is possible, the previous theorem is not applicable.

Other important issue to be taken into account is that it is required that the switching instants and all possible switching sequences are known during the entire infinite hybrid time set. This means that the switching rules are predefined as a function of time only. A third aspect to be taken into account is that when a switching subset S cannot be used to prove stability, it does not mean that another subset S' can also not be used. It can well be the case that a given sequence of switching instants does not possess a common Lyapunov function, but other particular sequence does.

Linear Polynomial Lyapunov Function

By changing the Lyapunov function to be considered, less restrictive stability conditions can be found. Particularly, a linear polynomial function of the parametrization vector as (3.3) can be used to test stability.

Theorem 3.3 (Linear Discrete Lyapunov)**.** *Given an infinite subset of the switching instants set, this is $S = \{\tau_{k_0}, \tau_{k_1}, \ldots, \tau_{k_h}, \ldots\} \subseteq S_\infty = \{\tau_0, \tau_1, \ldots\}$, switched system (3.1) is asymptotically stable towards the origin if there exists a linear function of the parametrization vector $\mathbf{P}(\boldsymbol{\alpha}) = \sum_{i=1}^{M} \alpha_i \mathbf{P}_i$ such that $\mathbf{P}_i > 0$ and $\forall h \in \mathbb{N}_0$,*

$$\left(\boldsymbol{\Psi}_{k_h}^{k_{h+1}}\right)' \mathbf{P}(\boldsymbol{\alpha}_{k_{h+1}}) \left(\boldsymbol{\Psi}_{k_h}^{k_{h+1}}\right) - \mathbf{P}(\boldsymbol{\alpha}_{k_h}) < 0, \tag{3.7}$$

with $k_{h+1} > k_h \in \mathbb{N}_0$, $\forall h \in \mathbb{N}_0$.

Proof. The proof is the same as for Theorem 3.2 but considering a linear Lyapunov function $w(\mathbf{x}(\tau_k), k) = \mathbf{x}(\tau_k)'\mathbf{P}(\boldsymbol{\alpha}(\tau_k + \delta))\mathbf{x}(\tau_k) > 0$, where $\delta > 0$ is chosen sufficiently small so that $\tau_k + \delta \in (\tau_k, \tau_{k+1})$. The vector $\boldsymbol{\alpha}(t + \delta)$ represents the discrete modes in an immediate future. Note that, in the limit, this vector evaluated at a switching instant τ_k becomes $\lim_{\varepsilon \to 0^+} \boldsymbol{\alpha}(\tau_k + \delta) = \boldsymbol{\alpha}_k$. □

Note that Theorem 3.2 is a special case of Theorem 3.3 when $\mathbf{P}(\boldsymbol{\alpha}) = \mathbf{P}$. With this formulation we observe the same application problems as in the previous theorem. Nevertheless, the inclusion of a linear polynomial on $\boldsymbol{\alpha}$ gives a higher degree of freedom that allows us to probe stability by considering a less restrictive switching instants sequences. The following corollary is immediate:

Corollary 3.4. *Switched system (3.1) is asymptotically stable towards the origin in the set of switching instants $S_\infty = \{\tau_0, \tau_1, \ldots\}$, if $\forall i \in \{1, 2, \ldots, M\}$, there exists matrices $\mathbf{P}_i > 0$ such that*

$$\boldsymbol{\Phi}_{k_h}' \mathbf{P}_j \boldsymbol{\Phi}_{k_h} - \mathbf{P}_i < 0, \tag{3.8}$$

where $\tau_{k_h} \in \{\tau_k \in S_\infty | [\boldsymbol{\alpha}_k]_i = 1\}$ (all instants at which the system switches to mode q_i), and $j \in \{1, 2, \ldots, M\}$ is such that $i \neq j$ and $[\boldsymbol{\alpha}_{k_h+1}]_j = 1$ (q_j is the mode where the system jumps from q_i).

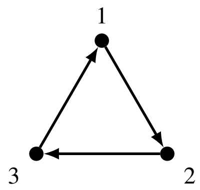


Figure 3.3.: Switched system for Sequential Process in Example 3.2.

Proof. Evaluate Theorem 3.3 at every switching instant and distinguish that each discrete mode allows one only possible jump (from q_i to q_j). □

This last result is useful because it associates with each switching instant the dynamic information of one, and only one, discrete mode. This is, at each instant, the respective LMI condition only includes one $\mathbf{A}_i$ matrix. This helps to decrease the number of LMI restrictions needed to prove stability of a switched system as the information of which transition is active at each switching instant is contained only in the Lyapunov matrix $\mathbf{P}(\boldsymbol{\alpha})$ and not implicitly in $\boldsymbol{\Psi}_{k_h}^{k_{h+1}}$. Similar results can be found in [133, 136, 140, 141, 170] and other works by the same authors.

The use of linear Lyapunov functions as described deals with the problem that it needs to be evaluated at *all* possible infinite switching instants. If the number of possible switching sequences is too large then the previous results might be not applicable in practice. Furthermore, if there is only information about the probability of occurrence of the switching between the states, this deterministic approach needs to be modified to accept some notion of stochastic stability.

Example - Periodic Switching Process

Example 3.2. Periodic switching processes are such that switching occurs on a periodical basis where a fixed sequence of discrete modes is repeated at regular intervals. Consider the sequences defined by the automaton in Figure 3.3 and the following matrices:

$$\mathbf{A}_1 = \begin{bmatrix} -1.0 & 0.2 \\ 0.0 & 0.3 \end{bmatrix}, \mathbf{A}_2 = \begin{bmatrix} 0.5 & 0.0 \\ -0.1 & 0.5 \end{bmatrix}, \mathbf{A}_3 = \begin{bmatrix} 0.2 & 0.2 \\ 0.0 & -3.0 \end{bmatrix}.$$

Note that all these matrices have at least one positive real eigenvalue. The sequence of mode changes $q_1 \to q_2$, $q_2 \to q_3$, $q_3 \to q_1$ will take place respectively at instants τ_{3n-3}, τ_{3n-2} and τ_{3n-1}, $n \in \mathbb{N}$. The system will stay in mode q_1 for $T_{3n-3} = \tau_{3n-2} - \tau_{3n-3} = 0.4$, in mode q_2 for $T_{3n-2} = \tau_{3n-1} - \tau_{3n-2} = 0.2$ and in q_3 for $T_{3n-1} = \tau_{3n} - \tau_{3n-1} = 0.4$ at each cycle. The parametrization vector as a function of time is such that $\forall n \in \mathbb{N}$:

$$\boldsymbol{\alpha}_{3n-3} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}', \ \boldsymbol{\alpha}_{3n-2} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}', \text{ and } \boldsymbol{\alpha}_{3n-1} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}',$$

so that $\mathbf{A}(\boldsymbol{\alpha}_{3n-3}) = \mathbf{A}_1$, $\mathbf{A}(\boldsymbol{\alpha}_{3n-2}) = \mathbf{A}_2$ and $\mathbf{A}(\boldsymbol{\alpha}_{3n-1}) = \mathbf{A}_3$. With this, we can define three matrices that describe the behavior of the system in all these possible changes:

$$\boldsymbol{\Psi}_{3n-3}^{3n-2} = \boldsymbol{\Phi}_{3n-3} = e^{\mathbf{A}(\boldsymbol{\alpha}_{3n-3})T_{3n-3}} = \begin{bmatrix} 0.6703 & 0.0703 \\ 0.0000 & 1.1275 \end{bmatrix},$$
$$\boldsymbol{\Psi}_{3n-2}^{3n-1} = \boldsymbol{\Phi}_{3n-2} = e^{\mathbf{A}(\boldsymbol{\alpha}_{3n-2})T_{3n-2}} = \begin{bmatrix} 1.1052 & 0.0000 \\ -0.0221 & 1.1052 \end{bmatrix},$$
$$\boldsymbol{\Psi}_{3n-1}^{3n} = \boldsymbol{\Phi}_{3n-1} = e^{\mathbf{A}(\boldsymbol{\alpha}_{3n-1})T_{3n-1}} = \begin{bmatrix} 1.0833 & 0.0489 \\ 0.0000 & 0.3012 \end{bmatrix}.$$

Note that the eigenvalues of the previous matrices are not in the unitary circle and therefore Theorem 3.2 is not applicable considering every switching instant. Furthermore, if we apply the theorem every two switching instants, *i.e.*, defining the matrices

$$\boldsymbol{\Psi}_{3n-3}^{3n-1} = \boldsymbol{\Psi}_{3n-3}^{3n-2} \cdot \boldsymbol{\Psi}_{3n-2}^{3n-1},$$
$$\boldsymbol{\Psi}_{3n-1}^{3n-2} = \boldsymbol{\Psi}_{3n-1}^{3n} \cdot \boldsymbol{\Psi}_{3n-3}^{3n-2},$$
$$\boldsymbol{\Psi}_{3n-2}^{3n} = \boldsymbol{\Psi}_{3n-2}^{3n-1} \cdot \boldsymbol{\Psi}_{3n-1}^{3n},$$

there is still no matrix $\mathbf{P}$ satisfying inequality (3.6).

However, if we consider the whole period of the sequential process, that is one every three switching instants, we can then define matrix $\boldsymbol{\Psi}_{3n-3}^{3n} = \boldsymbol{\Psi}_{3n-3}^{3n-2} \cdot \boldsymbol{\Psi}_{3n-2}^{3n-1} \cdot \boldsymbol{\Psi}_{3n-1}^{3n}$ which has all its eigenvalues inside the unitary circle and therefore there exists $\mathbf{P} > 0$ that satisfies inequality (3.6), *e.g.*,

$$\mathbf{P} = \begin{bmatrix} 1.4680 & 0.0615 \\ 0.0615 & 1.0815 \end{bmatrix}.$$

Note that when the whole sequential period is considered, the discrete time system associated with the switched system is time invariant.

Now consider Corollary 3.4. A linear polynomial $\mathbf{P}(\boldsymbol{\alpha}) = \alpha_1\mathbf{P}_1 + \alpha_2\mathbf{P}_2 + \alpha_3\mathbf{P}_3$ and the three possible jumps, lead to the following three LMIs derived from condition (3.8):

$$\boldsymbol{\Phi}_{3n-3}'\mathbf{P}_2\boldsymbol{\Phi}_{3n-3} - \mathbf{P}_1 < 0,$$
$$\boldsymbol{\Phi}_{3n-2}'\mathbf{P}_3\boldsymbol{\Phi}_{3n-2} - \mathbf{P}_2 < 0,$$
$$\boldsymbol{\Phi}_{3n-1}'\mathbf{P}_1\boldsymbol{\Phi}_{3n-1} - \mathbf{P}_3 < 0.$$

These conditions must hold simultaneously to probe the stability of the switched system under the specified switching sequence. The feasibility problem is satisfied by the following matrices whose existence proves the stability of the system under the given switching rule:

$$\mathbf{P}_1 = \begin{bmatrix} 1.2146 & 0.1436 \\ 0.1436 & 2.3796 \end{bmatrix},\ \mathbf{P}_2 = \begin{bmatrix} 2.3478 & 0.0634 \\ 0.0634 & 1.3591 \end{bmatrix},\ \mathbf{P}_3 = \begin{bmatrix} 1.6678 & 0.0843 \\ 0.0843 & 0.7039 \end{bmatrix}.$$

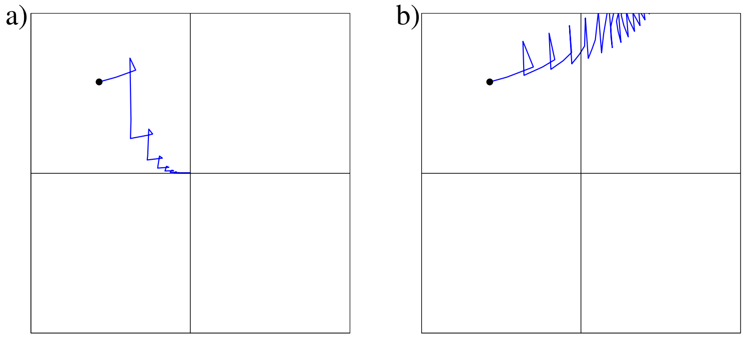


Figure 3.4.: Evolution of the states of systems in Example 3.2 for a) stable deterministic switching, b) non stable deterministic switching.

This result is confirmed by the simulation shown in Figure 3.4 a) where the evolution towards the origin of the continuous states are drawn from an arbitrary initial condition.

Note that the existence of the previous matrices only proves that the system is stable under the specified switched rule. Indeed, if other switching rule is defined, the switched system might not present a stable behavior. For example, with $T_{3n-3} = 0.5$, $T_{3n-2} = 0.4$ and $T_{3n-1} = 0.1$, the system becomes unstable as shown in Figure 3.4 b). ■

3.2.3. Probabilistic Switching

Up to here, only deterministic switching sequences have been addressed. That is, sequences where at every switching instant, the discrete modes associated to the jump are exactly known. If this assumption is relaxed to consider only the probability of switching from one mode to others, the problem of stability becomes stochastic in nature. Therefore, the definition of stability needs to be slightly modified. Equivalent statements to the following definitions can be found in several works as, for example, [150] which gives a easy to follow introduction to the topic. Other examples are [134, 135, 142–145, 148, 149, 176, 159, 160, 162, 164, 165, 168, 169, 171].

Definition 3.2.1. The stochastic discrete time system

$$\mathbf{x}(k+1) = h(k, w(k), \mathbf{x}(k)), \tag{3.9}$$

where $w(k)$ is a scalar stochastic process, and with initial condition $\mathbf{x}_0 = \mathbf{x}(k_0)$, is said to have an equilibrium point $\mathbf{x}^* = \mathbf{0}$ if, $\forall k \in \mathbb{N}$, $\mathbf{x}^* = h(k, w(k), \mathbf{x}^*) = \mathbf{0}$. This equilibrium point is said to be

- *Almost surely stable* if, for every $\varepsilon > 0$ and $h > 0$, there exists $\delta = \delta(\varepsilon, h, k_0) > 0$, such that

  $$P\{\|\mathbf{x}(k)\| < h\} \geqslant 1 - \varepsilon,\ k \geqslant k_0,$$

when $\|\mathbf{x}_0\| < \delta$.

- *Almost surely globally asymptotically stable* if it is almost surely stable and for all $\mathbf{x}_0 \in \mathbb{R}^n$,

$$P\left\{\lim_{k\to\infty} \|\mathbf{x}(k)\| = 0\right\} = 1.$$

In other words, $\mathbf{x}^* = \mathbf{0}$ is almost surely stable when for a small initial condition, the evolution of the discrete variable $\mathbf{x}(k)$ stays within a small region around the origin with a high probability. It is almost surely globally asymptotically stable if, additionally, for any initial condition, the states of the system evolve to the origin with probability one. This definitions can be directly extended to the continuous time case. In the quoted references, several equivalent names are given for the concept of "almost surely", *e.g.*, "with probability one" (w.p.1), "in probability", "stochastically", etc.

With this definitions, a stochastic Lyapunov theorem can be stated. The result is presented without proof. For a detailed explanation of the theorem see [150].

Theorem 3.5 (*e.g.* [150]). *Let $\mathbf{x}^* = \mathbf{0}$ be an equilibrium point for (3.9) and let $w : \mathbb{N}_0 \times \mathbb{R}^n \to \mathbb{R}$ be a continuous function such that $\forall k \in \mathbb{N}_0$ and $\forall \mathbf{x} \in \mathbb{R}^n$,*

$$w_1(\mathbf{x}(k)) \leqslant w(k, \mathbf{x}(k)) \leqslant w_2(\mathbf{x}(k)), \tag{3.10}$$

$$E\{\Delta w_k\} := E\{w(k+1, \mathbf{x}(k+1)) - w(k, \mathbf{x}(k))\} < 0, \tag{3.11}$$

where $w_1(\mathbf{x})$ and $w_2(\mathbf{x})$, are continuous positive definite functions on $\mathbb{R}^n$, $w_1(\mathbf{x})$ is radially unbounded, and $w(k, \mathbf{0}) = 0$. Then, $\mathbf{x}^* = \mathbf{0}$ is almost surely globally asymptotically stable.

From this result, a stochastic analysis of the switched system can be done if information on the switching probability at each state is known. For this, first we need to define the following.

Definition 3.2.2. The *probability vector* $\boldsymbol{\pi}_i^+ \in \Lambda_M$ associated to mode $q_i \in Q$, is such that each element $[\boldsymbol{\pi}_i^+]_j$ is the probability of finding the system in mode q_j immediately after switching from mode q_i at instant τ_{k_h+1}. It follows that for all $\tau_{k_h} \in \{\tau_k \in S_\infty | [\boldsymbol{\alpha}_k]_i = 1\}$,

$$\boldsymbol{\pi}_i^+ := E\{\boldsymbol{\alpha}_{k_h+1}\}.$$

We assume that this vector is always known for the studied systems. Note that, as we consider only switching between different modes, $[\boldsymbol{\pi}_i^+]_i = 0$. This assumption could be however relaxed to admit "switching" from one mode to itself. Furthermore, it can be considered that deterministic switching is a special case of stochastic switching that holds when $\boldsymbol{\pi}_i^+ = \boldsymbol{\alpha}_{k_h+1}$. That is, when the probability of the future state is zero for all modes but one.

The set of all M probability vectors defines implicitly a Markov Chain where the Markov matrix is given by

$$\mathbf{\Pi} = \operatorname{col}\left\{(\boldsymbol{\pi}_i^+)'\right\}_{i=1}^{M}.$$

The diagonal of this matrix is always zero as there cannot be a jump from one mode to itself. Therefore, the Markov chain cannot be absorbing, *i.e.* it cannot recursively jump into one final discrete mode. It depends on the structure of the edges that define the possible jumps if the chain is ergodic (or irreducible), *i.e.* if any mode can be reached by successive jumps from any other mode.

Fixed residence time

If the residence time T_{k_h} at each mode is known and constant, the following theorem can be stated:

Theorem 3.6. *Given a set of switching instants $S_\infty = \{\tau_0, \tau_1, \ldots\}$ and a set of probability vectors $\{\boldsymbol{\pi}_i^+\}_{i=1}^M$, switched system (3.1) is almost surely globally asymptotically stable towards the origin if there exists a symmetric homogeneous matrix polynomial $\mathbf{P}(\boldsymbol{\alpha}) = \sum_{i=1}^M \alpha_i \mathbf{P}_i$ such that $\forall i \in \{1,2,\ldots,M\}$, $\tau_{k_h} \in \{\tau_k \in S_\infty | [\boldsymbol{\alpha}_k]_i = 1\}$ (all instants at which the system switches to mode q_i), $\mathbf{P}_i > 0$ and*

$$\boldsymbol{\Phi}'_{k_h}\mathbf{P}(\boldsymbol{\pi}_i^+)\boldsymbol{\Phi}_{k_h} - \mathbf{P}_i < 0. \tag{3.12}$$

Proof. The proof is similar to the previous cases with a Lyapunov function $w(\mathbf{x}(\tau_k),k) = \mathbf{x}(\tau_k)'\mathbf{P}(\boldsymbol{\alpha}(\tau_k+\delta))\mathbf{x}(\tau_k) > 0$, with $\delta > 0$ small so that $\tau_k + \delta \in (\tau_k, \tau_{k+1})$. Considering a switching instant τ_{k_h} where the system switches to mode q_i, condition (3.12) follows from imposing $E\{\Delta w_{k_h}\} < 0$ and taking the limit $\varepsilon \to 0^+$. As the Lyapunov function is linear with respect to the elements of $\boldsymbol{\alpha}$, we obtain that $E\{\mathbf{P}(\boldsymbol{\alpha}_{k_h+1})\} = \mathbf{P}(E\{\boldsymbol{\alpha}_{k_h+1}\}) = \mathbf{P}(\boldsymbol{\pi}_i^+)$. Furthermore, $E\{\mathbf{P}(\boldsymbol{\alpha}_{k_h})\} = \mathbf{P}(E\{\boldsymbol{\alpha}_{k_h}\}) = \mathbf{P}(\boldsymbol{\alpha}_{k_h}) = \mathbf{P}_i$. □

Note that Theorem 3.6 associates exactly one LMI condition to each mode, independently of the number of possible switching sequences.

This result, or slightly different versions of it, is often found in the quoted references. In particular, in [134, 135] it is shown that conditions (3.12) are not only sufficient for almost surely stability, but necessary and sufficient for *mean square stability* (MSS), *i.e.* for $\lim_{k\to\infty} E\left\{\|\mathbf{x}(\tau_k)\|^2\right\} = 0$. In [134, 135] it is further shown that MSS implies almost surely stability but the reverse, that almost surely stability implies MSS, as far as we know, does not always hold. Therefore, MSS can be seen as a more restrictive condition than almost surely stability.

In [138] it is argued that the difficulty of finding suitable Lyapunov functions in general makes it hard to apply this methodology to stochastic systems. In particular, the computation of the expected value of the gradient of the Lyapunov function in Theorem 3.5 is not always possible and inhibits the use of more complex functions as, for example, homogeneous polynomials.

Uncertain residence time

In the cases where the residence time is only partially known, one can write $T_{k_h} = T_{k_h}^{min} + \Delta T_{k_h} \leqslant T_{k_h}^{min} + \Delta T_{k_h}^{max}$, where $T_{k_h}^{min} \in \mathbb{R}^+$ is a lower bound for the residence time and $\Delta T_{k_h} \in [0, \Delta T_{k_h}^{max}]$ an unknown deviation. The stability of a system under this kind of uncertainty can also be addressed by the previous result by considering a bound for the Euclidean norm of the exponential matrix. Given a time interval $I = \left[T^{min}, T^{max}\right]$, define the upper bound of the norm as

$$\delta(\mathbf{A}, I) := \max\{\|e^{\mathbf{A}t}\| \in \mathbb{R}^+ | t \in I\}.$$

Then it follows that $\forall t \in I$,

$$e^{\mathbf{A}'t} e^{\mathbf{A}t} \leqslant \|e^{\mathbf{A}t}\|^2 \mathbf{I} \leqslant \delta^2(\mathbf{A}, I)\mathbf{I}. \tag{3.13}$$

Note that the scalar bound always exits as $\|e^{\mathbf{A}t}\|$ is a continuous function of the parameter t within a closed interval. Even in cases where it might be difficult to numerically compute this value, an upper bound can be easily found, for example, by using the so-called *log norm* defined in Appendix A.2 to find the bound in equation (A.9). Another suitable option would be a bound typically used to compute approximations of the exponential matrix, derived from the Schur decomposition of matrix $\mathbf{A}$, see for example [14, Ch. 9.3.2, pp. 532]. These quadratic bounds can be used to modify the almost surely stability result of the previous section. Naturally, this implies a (possibly large) conservatism derived from the uncertain nature of the residence time and the quadratic bound.

Theorem 3.7. *In Theorem 3.6, if for some $i \in \{1, 2, \ldots, M\}$ the residence time is such that $T_{k_h} = T_{k_h}^{min} + \Delta T_{k_h} \leqslant T_{k_h}^{min} + \Delta T_{k_h}^{max}$, with $T_{k_h}^{min} \in \mathbb{R}^+$ and an unknown deviation $\Delta T_{k_h} \in [0, \Delta T_{k_h}^{max}]$, condition (3.12) can be replaced by:*

$$\begin{bmatrix} -\mathbf{P}_i + \eta_{k_h}\delta_{k_h}^2 \mathbf{I} & -\mathbf{X}_{k_h} & \mathbf{X}_{k_h} \\ \star & (\mathbf{\Phi}_{k_h}^{min})' \mathbf{P}(\boldsymbol{\pi}_i^+)\mathbf{\Phi}_{k_h}^{min} - \mathbf{Y}_{k_h} - \mathbf{Y}_{k_h}' & \mathbf{Y}_{k_h} \\ \star & \star & -\eta_{k_h}\mathbf{I} \end{bmatrix} < 0 \tag{3.14}$$

with additional variables $\eta_k \in \mathbb{R}^+$, $\mathbf{X}_{k_h} \in \mathbb{R}^{n \times n}$ and $\mathbf{Y}_{k_h} \in \mathbb{R}^{n \times n}$; and where $\mathbf{\Phi}_{k_h}^{min} := e^{\mathbf{A}(\boldsymbol{\alpha}_{k_h})T_{k_h}^{min}}$ and $\delta_{k_h} \geqslant \delta(\mathbf{A}(\boldsymbol{\alpha}_{k_h}), [0, \Delta T_{k_h}^{max}])$.

Proof. This proof uses the Finsler Lemma A.14 and the Crossed Products Proposition A.15 in Appendix A.4. Define,

$$\mathbf{\Upsilon} = \begin{bmatrix} e^{\mathbf{A}(\boldsymbol{\alpha}_{k_h})\Delta T_{k_h}} & -\mathbf{I} \end{bmatrix}, \mathbf{\Upsilon}^{\perp} = \begin{bmatrix} \mathbf{I} \\ e^{\mathbf{A}(\boldsymbol{\alpha}_{k_h})\Delta T_{k_h}} \end{bmatrix},$$

$$\mathbf{\Xi} = \begin{bmatrix} -\mathbf{P}_i & \mathbf{0} \\ \mathbf{0} & (\mathbf{\Phi}_{k_h}^{min})' \mathbf{P}(\boldsymbol{\pi}_i^+)\mathbf{\Phi}_{k_h}^{min} \end{bmatrix}, \mathbf{\Gamma} = \begin{bmatrix} \mathbf{X}_{k_h} \\ \mathbf{Y}_{k_h} \end{bmatrix}.$$

Where $\mathbf{X}_{k_h} \in \mathbb{R}^{n\times n}$ and $\mathbf{Y}_{k_h} \in \mathbb{R}^{n\times n}$ are new variables. With this and considering that

$$e^{\mathbf{A}(\boldsymbol{\alpha}_{k_h})(T_{k_h}^{min}+\Delta T_{k_h})} = \boldsymbol{\Phi}_{k_h}^{min} e^{\mathbf{A}(\boldsymbol{\alpha}_{k_h})\Delta T_{k_h}} = e^{\mathbf{A}(\boldsymbol{\alpha}_{k_h})\Delta T_{k_h}} \boldsymbol{\Phi}_{k_h}^{min},$$

condition ② of Finsler Lemma becomes

$$\begin{aligned}
(\boldsymbol{\Upsilon}^{\perp})'\boldsymbol{\Xi}\boldsymbol{\Upsilon}^{\perp} &= \begin{bmatrix} \mathbf{I} \\ e^{\mathbf{A}(\boldsymbol{\alpha}_{k_h})\Delta T_{k_h}} \end{bmatrix}' \begin{bmatrix} -\mathbf{P}_i & \mathbf{0} \\ \mathbf{0} & \left(\boldsymbol{\Phi}_{k_h}^{min}\right)' \mathbf{P}(\boldsymbol{\pi}_i^+)\boldsymbol{\Phi}_{k_h}^{min} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ e^{\mathbf{A}(\boldsymbol{\alpha}_{k_h})\Delta T_{k_h}} \end{bmatrix} \\
&= \boldsymbol{\Phi}_{k_h}' \mathbf{P}(\boldsymbol{\pi}_i^+)\boldsymbol{\Phi}_{k_h} - \mathbf{P}_i < 0 \\
&\iff (3.12).
\end{aligned}$$

The equivalent condition ④ of the Lemma can be conveniently written as

$$\begin{aligned}
\boldsymbol{\Xi} + \boldsymbol{\Gamma}\boldsymbol{\Upsilon} + \boldsymbol{\Upsilon}'\boldsymbol{\Gamma}' = &\begin{bmatrix} -\mathbf{P}_i & -\mathbf{X}_{k_h} \\ \star & \left(\boldsymbol{\Phi}_{k_h}^{min}\right)' \mathbf{P}(\boldsymbol{\pi}_i^+)\boldsymbol{\Phi}_{k_h}^{min} - \mathbf{Y}_{k_h} - \mathbf{Y}_{k_h}' \end{bmatrix} + \dots \\
&\dots + \boldsymbol{\Gamma}\begin{bmatrix} e^{\mathbf{A}(\boldsymbol{\alpha}_{k_h})\Delta T_{k_h}} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} e^{\mathbf{A}'(\boldsymbol{\alpha}_{k_h})\Delta T_{k_h}} \\ \mathbf{0} \end{bmatrix}\boldsymbol{\Gamma}' \\
&< 0.
\end{aligned}$$

As ④ $\iff$ ② , the last inequality is also a sufficient condition for stability of the system. It presents the advantage that the additional variables relax its numeric feasibility when considering bounds for the uncertainties given by the unknown quantity ΔT_{k_h}. From the Crossed Products Proposition and the bound of the exponential matrix norm (3.13), it follows that for any $\eta_{k_h} > 0$,

$$\begin{aligned}
\boldsymbol{\Gamma}\begin{bmatrix} e^{\mathbf{A}(\boldsymbol{\alpha}_{k_h})\Delta T_{k_h}} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} e^{\mathbf{A}'(\boldsymbol{\alpha}_{k_h})\Delta T_{k_h}} \\ \mathbf{0} \end{bmatrix}\boldsymbol{\Gamma}' &\leqslant \eta_{k_h} \begin{bmatrix} e^{\mathbf{A}'(\boldsymbol{\alpha}_{k_h})\Delta T_{k_h}} \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} e^{\mathbf{A}(\boldsymbol{\alpha}_{k_h})\Delta T_{k_h}} & \mathbf{0} \end{bmatrix} + \frac{1}{\eta_{k_h}}\boldsymbol{\Gamma}\boldsymbol{\Gamma}' \\
&= \begin{bmatrix} \eta_{k_h} e^{\mathbf{A}'(\boldsymbol{\alpha}_{k_h})\Delta T_{k_h}} e^{\mathbf{A}(\boldsymbol{\alpha}_{k_h})\Delta T_{k_h}} & \mathbf{0} \\ \star & \mathbf{0} \end{bmatrix} + \frac{1}{\eta_{k_h}}\boldsymbol{\Gamma}\boldsymbol{\Gamma}' \\
&\leqslant \begin{bmatrix} \eta_{k_h}\delta_{k_h}^2 \mathbf{I} & \mathbf{0} \\ \star & \mathbf{0} \end{bmatrix} + \frac{1}{\eta_{k_h}}\boldsymbol{\Gamma}\boldsymbol{\Gamma}'.
\end{aligned}$$

Therefore, an upper bound sufficient condition for ④ is

$$\begin{bmatrix} -\mathbf{P}_i + \eta_{k_h}\delta_{k_h}^2 \mathbf{I} & -\mathbf{X}_{k_h} \\ \star & \left(\boldsymbol{\Phi}_{k_h}^{min}\right)' \mathbf{P}(\boldsymbol{\pi}_i^+)\boldsymbol{\Phi}_{k_h}^{min} - \mathbf{Y}_{k_h} - \mathbf{Y}_{k_h}' \end{bmatrix} + \frac{1}{\eta_{k_h}}\boldsymbol{\Gamma}\boldsymbol{\Gamma}' < 0.$$

Condition (3.14) is equivalent by Schur complement to the last inequality. □

Note that a similar result could be proposed for the deterministic switching case.

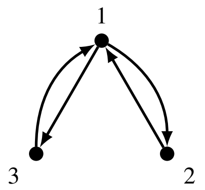


Figure 3.5.: Switched system in Fault Operation for Example 3.3.

Example - Operation under Faults

Example 3.3. Consider the switching sequences defined by the automaton in Figure 3.5 and the following matrices:

$$\mathbf{A}_1 = \begin{bmatrix} -2 & -1 \\ -1 & -2 \end{bmatrix}, \mathbf{A}_2 = \begin{bmatrix} 1 & 0 \\ -1 & -2 \end{bmatrix}, \mathbf{A}_3 = \begin{bmatrix} -2 & -1 \\ 0 & 1 \end{bmatrix}.$$

Note that the first matrix is Hurwitz, but not the others. This represents a system that operates in a nominal safe mode q_1 but where two different fault modes q_2 and q_3 may occur. In this example, the indexes associated with jumps into mode q_1 are denoted as $k_{h,1}$, to jumps into q_2 as $k_{h,2}$, and into q_3 as $k_{h,3}$. In that way, the switching instants associated with a jump to q_1 are $\tau_{k_{h,1}}$, to q_2 are $\tau_{k_{h,2}}$, and to q_3 are $\tau_{k_{h,3}}$. Furthermore, $\mathbf{A}(\boldsymbol{\alpha}_{k_{h,1}}) = \mathbf{A}_1$, $\mathbf{A}(\boldsymbol{\alpha}_{k_{h,2}}) = \mathbf{A}_2$, and $\mathbf{A}(\boldsymbol{\alpha}_{k_{h,3}}) = \mathbf{A}_3$.

Assume that the residence times at each discrete mode are known and constant with $T_{(1)} := T_{k_{h,1}} = 0.2$ for q_1, $T_{(2)} := T_{k_{h,2}} = 0.5$ for q_2 and $T_{(3)} := T_{k_{h,3}} = 0.4$ for q_3. Note that the infinite repetitive sequence $\ldots \to q_1 \to q_2 \to q_1 \to q_2 \to \ldots$ is not stable as the matrix

$$\boldsymbol{\Psi}_{k_{h,2}}^{k_{h,1}} = \boldsymbol{\Phi}_{k_{h,1}} \boldsymbol{\Phi}_{k_{h,2}} = e^{\mathbf{A}(\boldsymbol{\alpha}_{k_{h,1}})T_{k_{h,1}}} e^{\mathbf{A}(\boldsymbol{\alpha}_{k_{h,2}})T_{k_{h,2}}}$$

has one eigenvalue, $\lambda = 1.2116$, outside of the unitary circle. The same happens with the sequence $\ldots \to q_1 \to q_3 \to q_1 \to q_3 \to \ldots$. Therefore the system is not stable for all allowed switching sequences.

However, if we additionally know that the fault described by q_2 occurs once every three faults, the previous sequences are very unlikely to happen. In this case, the future probability vectors are:

$$\boldsymbol{\pi}_1^+ = \begin{bmatrix} 0 & 1/3 & 2/3 \end{bmatrix}', \ \boldsymbol{\pi}_2^+ = \boldsymbol{\pi}_3^+ = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}'.$$

For this system, the LMI conditions of Theorem 3.6 are fulfilled by $\mathbf{P}(\boldsymbol{\alpha}) = \mathbf{P}_1\alpha_1 + \mathbf{P}_2\alpha_2 + \mathbf{P}_3\alpha_3 > 0$ where

$$\mathbf{P}_1 = \begin{bmatrix} 0.8895 & -0.4591 \\ -0.4591 & 1.3860 \end{bmatrix}, \mathbf{P}_2 = \begin{bmatrix} 3.6648 & -0.2205 \\ -0.2205 & 0.4813 \end{bmatrix}, \mathbf{P}_3 = \begin{bmatrix} 0.4832 & -0.2270 \\ -0.2270 & 3.8499 \end{bmatrix}.$$

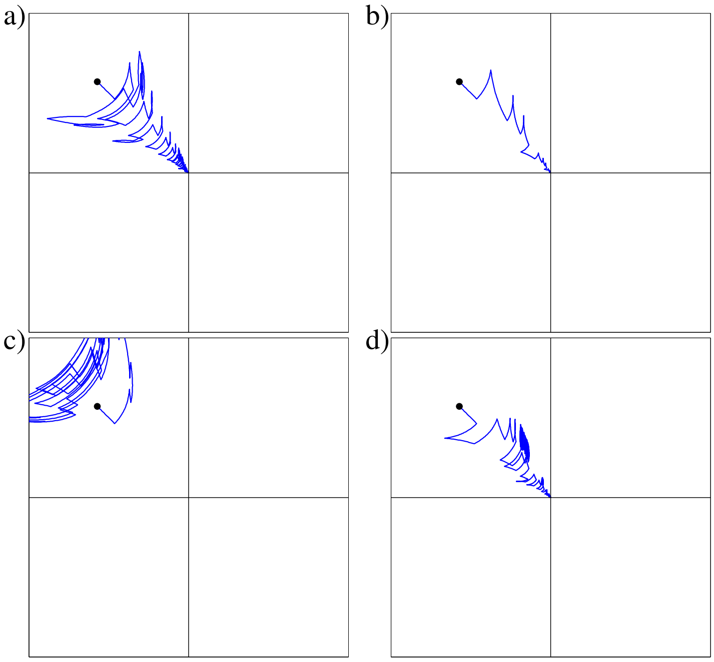


Figure 3.6.: Evolution of the states of systems in Example 3.3 for a) almost surely stable probabilistic switching, b) almost surely stable uncertain probabilistic switching, c) non almost surely stable uncertain probabilistic switching, d) presumable almost surely stable uncertain probabilistic switching.

This shows that the specified system is almost surely stable. This can be seen in Figure 3.6 a) for a random switching sequence with the switching probabilities stated before.

Now consider that residence time for each mode is not exactly known. For each mode, the uncertainty is however bounded and given by

$$
\begin{aligned}
q_1 :& T^{min}_{k_{h,1}} = 0.50 \quad \text{and } \Delta T^{max}_{k_{h,1}} = 0.05,\\
q_2 :& T^{min}_{k_{h,2}} = 0.30 \quad \text{and } \Delta T^{max}_{k_{h,2}} = 0.20,\\
q_3 :& T^{min}_{k_{h,3}} = 0.30 \quad \text{and } \Delta T^{max}_{k_{h,3}} = 0.10.
\end{aligned}
$$

A bound $\delta_{k_{h,i}} \geqslant \delta(\mathbf{A}(\boldsymbol{\alpha}_{k_{h,i}}), [0, \Delta T^{max}_{k_{h,i}}])$, $i \in \{1,2,3\}$, can be easily obtained by computing the norm of the exponential map for all values in the interval as shown in Figure 3.7. As mode q_1 presents stable non-oscillating dynamics, the maximum norm is obtained at $\Delta T_{k_{h,1}} = 0.00$ and therefore we can choose $\delta_{k_{h,1}} = 1.0000$. In the fault modes, q_2 and q_3, because of their unstable non-oscillating dynamics, the maximum is given at $\Delta T_{k_{h,i}} = \Delta T^{max}_{k_{h,i}}$ and so we can choose, $\delta_{k_{h,2}} = 1.2408$ and $\delta_{k_{h,3}} = 1.1141$.

The application of Theorem 3.7 results in this case in three LMIs, each one with the variables associated to the Lyapunov function with one additional scalar variable (η-type

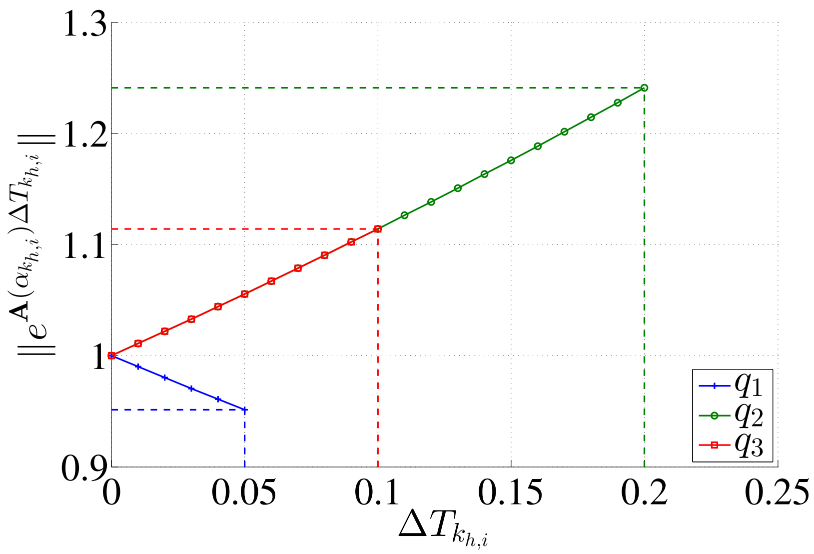


Figure 3.7.: Norm of the exponential map as a function of the uncertainty in the residence time, $\Delta T_{k_{h,i}} \mapsto \|e^{\mathbf{A}(\boldsymbol{\alpha}_{k_{h,i}})\Delta T_{k_{h,i}}}\|$, for the three modes, $i \in \{1,2,3\}$, of the system in Example 3.3.

variable) and two additional matrix variables (**X**- and **Y**-type variables). That is, there are $3 \times (2+1)(2)/2 = 9$ Lyapunov scalar variables, and $3 \times (1+2^2+2^2) = 27$ additional scalar variables. In the case of the specified uncertainty, the corresponding LMIs can be proven feasible, thus showing that the system is stable. We do not show these matrices for sake of space. This is corroborated by the simulation shown in Figure 3.6 b).

If we consider a more restrictive uncertainty given by

$$\begin{aligned}
q_1 &: T^{min}_{k_{h,1}} = 0.0667 \quad \text{and } \Delta T^{max}_{k_{h,1}} = 0.05,\\
q_2 &: T^{min}_{k_{h,2}} = 0.3000 \quad \text{and } \Delta T^{max}_{k_{h,2}} = 0.20,\\
q_3 &: T^{min}_{k_{h,3}} = 0.3000 \quad \text{and } \Delta T^{max}_{k_{h,3}} = 0.10,
\end{aligned}$$

the inequalities resulting of Theorem 3.7 are not feasible. Because this result only gives a sufficient condition for stability, the non feasibility of the LMIs does not mean that the system is unstable. However, it leaves space for well-founded doubts. Indeed, the simulation in Figure 3.6 c) shows that the switched system under these conditions does not approximate the origin.

However, as Theorem 3.7 is only a conservative condition, because an upper bound is considered in order to deal with the uncertain parameters, the cases where the inequalities are not feasible need to be interpreted carefully. For example, if we choose the following parameters

$$
\begin{aligned}
q_1 :& T_{k_{h,1}}^{min} = 0.20 \quad \text{and } \Delta T_{k_{h,1}}^{max} = 0.05, \\
q_2 :& T_{k_{h,2}}^{min} = 0.30 \quad \text{and } \Delta T_{k_{h,2}}^{max} = 0.20, \\
q_3 :& T_{k_{h,3}}^{min} = 0.30 \quad \text{and } \Delta T_{k_{h,3}}^{max} = 0.10,
\end{aligned}
$$

the respective LMIs are not feasible, although this values represent a relaxed situation with respect to the system without uncertainty at the beginning of the example (the system stays during less or equal time in the unstable modes as in the original case but longer in the stable mode). Simulations show that the system converges in all considered cases as in Figure 3.6 d).
■

Part II.

Consensus Systems

4

The Global Perspective

4.1. Multi Agents Systems

Even though consensus based control is formulated for Multi Agents Systems, it is not easy to find a general description of such a system in the related works. In this section, the general model used in this thesis is explained and some important restrictions are stated. As a result, this section summarizes the characteristics of the plant over which control is performed.

4.1.1. General Description of Multi Agents Systems

In Figure 4.1 a general distributed control scheme for a multi-agent system is shown. This representation considers a realistic scenario in a control theoretical framework, where the different components of the network are defined according to their physical characteristics or functions. The subsystems are detailed in the following list.

Subsystems:

- **Agent** i: N controlled machines. Possible non linear dynamics. The agents dynamics can be extended to other cases as discrete time or discrete states. The actuators are assumed to be part of the model of the agent. Example: Electric generators in a grid; Mobile vehicles.
- **Controller** i: each agent is equipped with an on board local controller that considers control tasks, input and output filtering, and communication management.
- **Communication Channels**: Is a communication interface between the on board components. Note that there is no control logic allocated in this block. This block includes the feedback signals from each block to itself. In the case that the dynamics of such a feedback are not of importance, the corresponding communication channel can always be modeled as a unitary matrix gain.
- **Hardware Interconnections**: All physical links existing between the agents that are not part of the control strategy. This block might be unknown, partially unknown or even not exist. Example: The electrical grid where generators are connected; terrain constrains where the vehicles move.

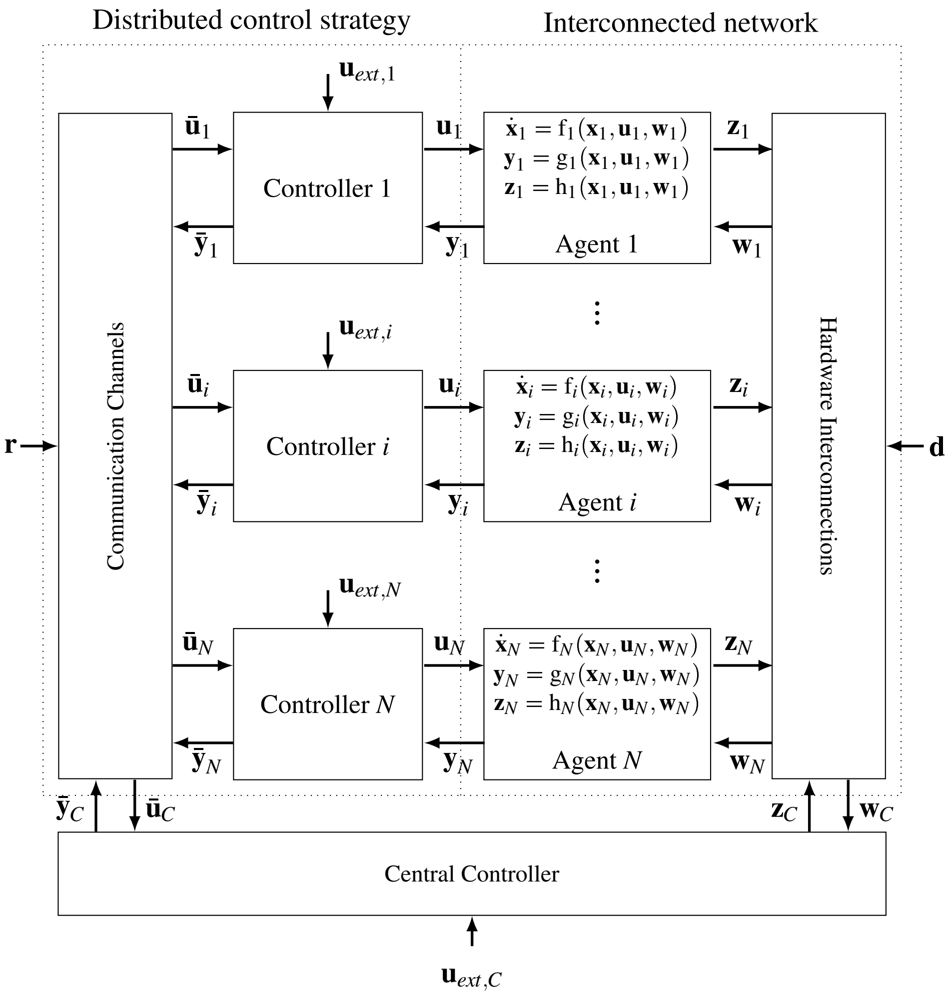


Figure 4.1.: General scheme of a multi-agents system.

- **Distributed control strategy**: Is the union set of on board controllers with the communication channels. If the control objective is consensus, then this block is the *Consensus Algorithm*.

- **Interconnected network**: Is the union set of all agents and the hardware interconnections between them.

- **Central controller**: Is an additional agent that interacts through communication channels with the on board logic of the rest of the agents or through hardware interconnections directly with the agents. It is considered an extra agent due to its importance for the control strategy, its great influence on hardware interconnections or its different nature. While the agents are some kind of similar systems (for example aircrafts or electric generators), the central controller can be special hardware designed for specific tasks.

Even though not expressly shown, all these blocks can be dynamic systems. Figure 4.1 shows the agents as continuous dynamical systems, however the dynamics of different blocks can well be in continuous time, discrete time, discrete event or combinations of the previous. The main variables associated with these blocks are described in the following list.

Variables:

- $\mathbf{x}_i$: Vector of states of agent i.

- $\mathbf{y}_i$: Vector of physical outputs of agent i.

- $\bar{\mathbf{y}}_i$: Vector of communicated outputs of agent i.

- $\mathbf{u}_i$: Vector of control inputs of agent i. The actuators are assumed to be included in the agents.

- $\bar{\mathbf{u}}_i$: Vector of data that the controller of agent i obtains from communication with others.

- $\mathbf{u}_{ext,i}$: Vector of external inputs of agent i to allow the possibility of manual operation of the agents.

- $\mathbf{z}_i$: Vector of the physical variables of agent i that interact with other agents.

- $\mathbf{w}_i$: Vector of physical variables that affect agent i as a result of the physical interaction between agents.

- $\mathbf{d}$: Vector of perturbations.

- $\mathbf{r}$: Vector of communication noise.

4.1.2. Model Restrictions

The described model is very widespread in the sense that it allows many different dynamical models for its components, making the analysis difficult. For this reason, only linear continuous dynamics will be considered. Due to the methodology followed here, the results to be presented can be extended to other more complex scenarios. However, not making the linearity assumption of the models would lead to complications that are not due to the multi agent plant or the consensus problem, but due to the modeling of the components. With this in mind, the following definitions will be considered in most parts of this thesis.

Definition 4.1.1. A linear *autonomous agent* (AA) is an agent $i \in \mathscr{V}$ that does not have any hardware interconnections with any other agent and presents individual dynamics given by:

$$\begin{aligned}\dot{\mathbf{x}}_i &= \mathbf{A}_i\mathbf{x}_i + \mathbf{B}_i\mathbf{u}_i \\ \mathbf{y}_i &= \mathbf{C}_i\mathbf{x}_i\end{aligned} \tag{4.1}$$

With $\mathbf{A}_i \in \mathbb{R}^{n_i \times n_i}$, $\mathbf{B}_i \in \mathbb{R}^{n_i \times p_i}$, and $\mathbf{C}_i \in \mathbb{R}^{q \times n_i}$.

Note that the number of outputs does not depend on the agent but is always q. We assume that $\mathbf{C}_i\mathbf{B}_i$ is full rank. Unless specifically stated otherwise, we also assume that each agent has the same number of outputs as inputs, that is $q = p_i$.

A typical special case in the consensus field are agents modeled as integrators.

Definition 4.1.2. An *integrator agent* (IA) is an AA that presents individual dynamics given by:

$$\begin{aligned}\dot{\mathbf{x}}_i &= \mathbf{B}_i\mathbf{u}_i \\ \mathbf{y}_i &= \mathbf{C}_i\mathbf{x}_i\end{aligned} \tag{4.2}$$

That is, an AA with $\mathbf{A} = \mathbf{0}$, so that $\dot{\mathbf{y}}_i = \mathbf{C}_i\mathbf{B}_i\mathbf{u}_i$.

Definition 4.1.3. An *autonomous agents network* (AAN) is the aggregation of all N autonomous agents in a set $\mathscr{V}$. The dynamics of such a network are described by:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x}\end{aligned} \tag{4.3}$$

Where,

$$\begin{aligned}&\mathbf{A} = \operatorname{diag}\{\mathbf{A}_i\}_{i=1}^N, \ \mathbf{B} = \operatorname{diag}\{\mathbf{B}_i\}_{i=1}^N, \ \mathbf{C} = \operatorname{diag}\{\mathbf{C}_i\}_{i=1}^N, \\ &\mathbf{x} = \operatorname{col}\{\mathbf{x}_i\}_{i=1}^N, \ \mathbf{u} = \operatorname{col}\{\mathbf{u}_i\}_{i=1}^N, \ \mathbf{y} = \operatorname{col}\{\mathbf{y}_i\}_{i=1}^N\end{aligned}$$

so that $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times p}$, $\mathbf{C} \in \mathbb{R}^{Nq \times n}$, $n = \sum_{i=1}^N n_i$ and $p = \sum_{i=1}^N p_i$.

Definition 4.1.4. An *integrators network* (IN) is an AAN composed only of IA, so $\mathbf{A} = \mathbf{0}$ and

$$\dot{\mathbf{y}} = \mathbf{CBu}$$

where matrix $\mathbf{CB} = \operatorname{diag}\{\mathbf{C}_i\mathbf{B}_i\}_{i=1}^{N}$ is full rank.

A network described in such a way can be classified according to its size.

Definition 4.1.5. A *small* network is such that:

i It can be analyzed in a centralized way.

ii The agents are capable of computing variables as a function of the information they know about the others.

If any of these assumptions is dropped, the network will be called *medium sized*. If none of the assumptions is true, the network is *big*.

Unless otherwise stated, this thesis deals with small networks, or at least with medium networks where a centralized analysis can be performed. In these cases it is possible to have information about the whole network and the analysis can be done from a global point of view, considering all possible relationships between agents. Note that a centralized analysis of a network does not mean that its control is done from a centralized position. Control actions and hardware can be distributed among the agents and still be analyzed from a central position.

4.2. The Consensus Objective

4.2.1. The Idea of Consensus

Given a Multi Agent System described as before, consensus can be defined as a control objective in the same way as stability or robustness in classical control. That is, the definition of consensus is independent of the agent's dynamics or methodology that the agents follow to reach this objective. It is however not an exception in the field, *e.g.* [14, 29, 34, etc.], to find definitions not only in terms of the output signals but also in terms of specific dynamics (usually integrators) and specific consensus algorithms (usually Laplacian algorithms). That is, not as a control objective for synthesis of controllers in an arbitrary plant, but as a property of particular control plants with particular controllers that can be analyzed.

Informally, to *reach consensus* is understood in this thesis as the outputs of different agents having an equivalent value. This value is usually the same, where some publications talk about *point consensus*, but it can also be defined as the difference with a given known vector. In that sense consensus can be intuitively compared with the equilibrium point of a system that resumes the characteristics of the whole network. However, explicitly reducing consensus to a stability problem, is not typically addressed in the existing works. Nevertheless, some recent conference papers, *e.g.* [4, 35, 50], have shown that consensus can be explained through the idea of a unique "leader" agent.

Consensus is usually defined as $\lim_{t\to+\infty} \|\mathbf{y}_i - \mathbf{y}_j\| = 0, \forall i, j \in \mathscr{V}$ [14, 29, 34, etc.]. That is, in the limit, the output signals of *every* agent $i \in \mathscr{V}$ need to be the same. However, due to the symmetry ($a = b \iff b = a$) and transitivity ($a = b \wedge b = c \implies a = c$) properties of the equality relationship, in the limit, this definition becomes redundant and therefore expensive to test.

Indeed, if it is true that $\mathbf{y}_i - \mathbf{y}_j = \mathbf{0}$, it is also true that $\mathbf{y}_j - \mathbf{y}_i = \mathbf{0}$ and therefore only one (and not both!) of these relationships needs to be computed in order to check that agent i and agent j reached consensus. Furthermore, if it is also true that $\mathbf{y}_i - \mathbf{y}_k = \mathbf{0}$, there is no need to compute the differences $\mathbf{y}_j - \mathbf{y}_k = \mathbf{y}_k - \mathbf{y}_j = \mathbf{0}$ to verify that agent k reached consensus with agent j, as the transitivity property assures it already. The previous definition also implies that, to check if consensus is achieved, the trivial differences $\mathbf{y}_i - \mathbf{y}_i = \mathbf{0}$ also needs to be computed.

In the case of three agents i, j and k, the quoted definition implies that six non trivial relationships ($\mathbf{y}_i - \mathbf{y}_j = \mathbf{0}$, $\mathbf{y}_i - \mathbf{y}_k = \mathbf{0}$, $\mathbf{y}_j - \mathbf{y}_i = \mathbf{0}$, $\mathbf{y}_j - \mathbf{y}_k = \mathbf{0}$, $\mathbf{y}_k - \mathbf{y}_i = \mathbf{0}$, and $\mathbf{y}_k - \mathbf{y}_j = \mathbf{0}$) need to be verified to say that the agents reached consensus. However, in the limit, only two of them are actually required. In general, for N agents, only $N-1$ relationships need to be studied and not all possible $N(N-1)$ non trivial relationships. This property is further exploited in the following sections to define the idea of *organization* and from there to redefine consensus as a stability problem.

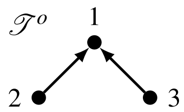
4.2.2. Hierarchical Organization

From a collective perspective, a network can be described through hierarchical relationships between the agents. The nature or structure of these relationships is defined arbitrarily by the analysis instance and are independent of the agents' dynamics or communication channels. Therefore, an organization is not a physical concept related to the network dynamics, but an arbitrary analysis tool.

Definition 4.2.1. Given a strictly directed graph, $\mathscr{T}^o$, derived from an unweighted undirected tree $\mathscr{T}$ over the set of agents $\mathscr{V}$, with $|\mathscr{V}| = N$, where $D(\mathscr{T}^o)$ denotes its incidence matrix as in Definition 2.1.9. Then $\mathbf{T} = D'(\mathscr{T}^o) \in \mathbb{R}^{(N-1)q\times Nq}$ is called an *organization matrix*.

Trees represent hierarchical organizations in a natural way when each agent is a node and the hierarchical relationship between the systems is an oriented edge. The use of a particular transformation instead of another, implies the choice of one particular way to study the network. For example, it can be defined that all systems follow only one reference system, that each agent follows only one other agent, or even that only one agent follows every other one.

The organization matrix is unique given a graph $\mathscr{T}^o$. However, note that different organization matrices could be constructed from strict digraphs defined from the same given undirected tree $\mathscr{T}$, by giving the edges different directions or labeling them in a different way. Considering Cayley's formula, between N agents, N^{N-2} spanning trees can be drawn. For each of these



$$\mathbf{T} = D'(\mathscr{T}^o) = \begin{bmatrix} \mathbf{I} & -\mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & -\mathbf{I} \end{bmatrix}$$

Figure 4.2.: Example of a consensus organization.

trees, there are 2^{N-1} ways to give orientations to the edges. Furthermore, there are $(N-1)!$ ways to label $N-1$ oriented edges. Therefore, the number of possible organization matrices that can be defined to study a network of N agents is $N^{N-2} \cdot 2^{N-1} \cdot (N-1)!$.

Example 4.1. Figure 4.2 gives an example of an organization matrix derived from a tree. The labels of the edges are assumed $e_1 = (2,1)$ and $e_2 = (3,1)$ and, therefore, the first row of matrix $\mathbf{T}$ corresponds to the edge $(2,1)$ and the second row to $(3,1)$. If another digraph is assumed with labels $e_1 = (3,1)$ and $e_2 = (2,1)$, the resulting organization matrix would be

$$\mathbf{T} = D'(\mathscr{T}^o) = \begin{bmatrix} \mathbf{I} & \mathbf{0} & -\mathbf{I} \\ \mathbf{I} & -\mathbf{I} & \mathbf{0} \end{bmatrix}.$$

With the original labeling, if the orientation of the edges is changed, thus defining different digraphs, the following additional organizations matrices can be defined.

$$\mathbf{T} = \begin{bmatrix} -\mathbf{I} & \mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & -\mathbf{I} \end{bmatrix}, \quad \mathbf{T} = \begin{bmatrix} \mathbf{I} & -\mathbf{I} & \mathbf{0} \\ -\mathbf{I} & \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{T} = \begin{bmatrix} -\mathbf{I} & \mathbf{I} & \mathbf{0} \\ -\mathbf{I} & \mathbf{0} & \mathbf{I} \end{bmatrix}.$$

By also changing the labeling, three more matrices can be defined. If other undirected trees over the same set of vertices are considered, several other matrices can be defined. As $N = 3$ only two additional undirected trees with $\mathscr{E} = \{(1,2),(2,3)\}$ and $\mathscr{E} = \{(1,3),(2,3)\}$ can be drawn. In total, for this simple case with three vertices there are $3^{3-2} \cdot 2^{3-1} \cdot (3-1)! = 24$ possible organization matrices. ■

By construction of $\mathbf{T} = D'(\mathscr{T}^o)$, an organization matrix is always full row rank and therefore it always has a unique pseudo-inverse matrix[1] $\mathbf{T}^+$ so that $\mathbf{T}\mathbf{T}^+ = \mathbf{I}$ and that can be computed as

$$\mathbf{T}^+ = \mathbf{T}'(\mathbf{T}\mathbf{T}')^{-1}.$$

Furthermore, the matrix $\mathbf{1} = \mathrm{col}\{\mathbf{I}_{q\times q}\}_{i=1}^N$ is a basis of the kernel of $\mathbf{T}$, *i.e.* $\mathbf{T1} = \mathbf{0}$ and the composed matrix $\begin{bmatrix} \mathbf{T}' & \mathbf{1} \end{bmatrix}$ is non singular. From here, the following can be written:

$$\begin{bmatrix} \mathbf{T}' & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{T}' & \mathbf{1} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{T} \\ \mathbf{1}' \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{T} \\ \mathbf{1}' \end{bmatrix} = \mathbf{I}$$

[1] More details on pseudo inverses can be found in the Appendix A.1.

By developing the inverse terms, one gets:

$$\begin{bmatrix} \mathbf{T}' & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{TT}' & \mathbf{0} \\ \mathbf{0} & N\mathbf{I} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{T} \\ \mathbf{1}' \end{bmatrix} = \mathbf{I}$$

Which finally leads to:

$$\mathbf{T}^{+}\mathbf{T} = \mathbf{I} - \frac{1}{N}\mathbf{11}' \tag{4.4}$$

For simplicity, we define

$$\mathbf{J} := \frac{1}{N}\mathbf{11}'.$$

Note that $\mathbf{JJ} = \mathbf{J}$, $\mathbf{TJ} = \mathbf{0}$ and $\mathbf{JT}^{+} = \mathbf{0}$.

Remark 4.1. Many of the possible organization matrices that can be defined have equivalent properties. In particular, given an organization $\mathbf{T}$ defined from an undirected unweighted tree $\mathscr{T}$, all $(N-1)!$ organizations $\hat{\mathbf{T}}$ resulting from different labeling of the edges are row permutations of $\mathbf{T}$. That is, they can be expressed as $\hat{\mathbf{T}} = \hat{\mathbf{M}}\mathbf{T}$, where $\hat{\mathbf{M}} \in \mathbb{R}^{(N-1)q\times(N-1)q}$ is a block permutation matrix and so $\hat{\mathbf{M}}\hat{\mathbf{M}}' = \hat{\mathbf{M}}'\hat{\mathbf{M}} = \mathbf{I}$. In this cases $\hat{\mathbf{T}}^{+} = \hat{\mathbf{T}}'(\hat{\mathbf{T}}\hat{\mathbf{T}}')^{-1} = \mathbf{T}'\hat{\mathbf{M}}'(\hat{\mathbf{M}}\mathbf{TT}'\hat{\mathbf{M}}')^{-1} = \mathbf{T}'\hat{\mathbf{M}}'(\hat{\mathbf{M}}')^{-1}(\mathbf{TT}')^{-1}(\hat{\mathbf{M}})^{-1} = \mathbf{T}^{+}\hat{\mathbf{M}}'$.

Furthermore, all 2^{N-1} organizations $\bar{\mathbf{T}}$ resulting from different orientations of the edges can be expressed as $\bar{\mathbf{T}} = \bar{\mathbf{M}}\mathbf{T}$, where $\bar{\mathbf{M}} \in \mathbb{R}^{(N-1)q\times(N-1)q}$ is a block diagonal matrix such that each of its $N-1$ diagonal $(q\times q)$-blocks is either $\mathbf{I}$ or $-\mathbf{I}$. Clearly, $\bar{\mathbf{M}} = \bar{\mathbf{M}}'$ and $\bar{\mathbf{M}}\bar{\mathbf{M}} = \mathbf{I}$. Therefore, $\bar{\mathbf{T}}^{+} = \mathbf{T}^{+}\bar{\mathbf{M}}'$.

4.2.3. Consensus Error

For any of these organizations a vector of consensus errors can be defined.

Definition 4.2.2. The *consensus error vector* of a network analyzed through organization $\mathbf{T} = D'(\mathscr{T}^{o})$ is defined as

$$\mathbf{e} = \mathbf{Ty} \tag{4.5}$$

The multiplication of each of the $N-1$ block rows of matrix $\mathbf{T}$ with vector $\mathbf{y}$ computes the difference between the outputs of the different agents. Therefore, if the norm of this error decreases to zero over time, the network will achieve consensus in steady state. Consensus can be then redefined by means of the organization idea as the convergence of the error targets $\mathbf{e}$:

Definition 4.2.3. A network of autonomous agents is said to reach or achieve *consensus* if the error $\mathbf{e}$ defined by an organization $\mathbf{T} = D'(\mathscr{T}^{o})$ asymptotically approaches the origin for any initial condition. That is, $\forall \mathbf{x}(0) \in \mathbb{R}^{n}$:

$$\text{Consensus} \iff \lim_{t\to+\infty} \|\mathbf{e}\| = 0$$

The analysis of consensus through the organization matrix and the convergence of the error vector is similar to what is done in [2, 37, 38] but considering the outputs space and not the particular dynamics of the states. From here, the definition of consensus suggested in Section 4.2.1 can be proved equivalent.

Lemma 4.1. $\lim_{t\to+\infty}\|\mathbf{e}\| = 0 \iff \lim_{t\to+\infty}\|\mathbf{y}_i - \mathbf{y}_j\| = 0, \forall i \in \mathscr{V} \wedge j \in \mathscr{V}.$

Proof. Define $\mathbf{T} = D'(\mathscr{T}^o)$ from a tree $\mathscr{T} = (\mathscr{V}, \mathscr{E})$. For any instant $t \geqslant 0$, it follows that,

$$\begin{aligned}
\|\mathbf{e}\| = 0 &\iff \|\mathbf{e}\|^2 = \|\mathbf{T}\mathbf{y}\|^2 = \sum_{(i,j)\in\mathscr{E}} \|\mathbf{y}_i - \mathbf{y}_j\|^2 = 0 \\
&\iff \forall (i,j) \in \mathscr{E} : \mathbf{y}_i - \mathbf{y}_j = \mathbf{0} \\
&\iff \forall (i,j) \in \mathscr{V} \times \mathscr{V} : \mathbf{y}_i - \mathbf{y}_j = \mathbf{0} \\
&\iff \forall (i,j) \in \mathscr{V} \times \mathscr{V} : \|\mathbf{y}_i - \mathbf{y}_j\| = 0.
\end{aligned}$$

Where the change of $\mathscr{E}$ by $\mathscr{V} \times \mathscr{V}$ holds because $\mathscr{T}$ is a tree and through transitivity and symmetry of the equality operation. □

Note that achieving consensus does not necessarily imply that the states or outputs of the agents approach the origin, only ensures that the target error vanishes. With this definition, the case where the outputs of the systems tend to infinity but are equal after some transient is considered a successful coordination. From these last observations, the following types of consensus may be defined.

Definition 4.2.4. For any initial condition, consensus is said to be

a) *Trivial* when

$$\lim_{t\to+\infty}\|\mathbf{e}(t)\| = 0 \iff \lim_{t\to+\infty}\|\mathbf{y}(t)\| = 0.$$

b) *Static* when it is not trivial and $\forall t \geqslant 0$, $\exists \mathbf{v} \in \mathbb{R}^q$:

$$\|\mathbf{e}(t)\| = 0 \implies \|\mathbf{y}(t) - \mathbf{1}\mathbf{v}\| = 0.$$

c) *Bounded dynamic* when it is not trivial and $\forall t \geqslant 0$, $\exists \mathbf{v} : \mathbb{R}_0^+ \longmapsto \mathbb{R}^q$ and $\exists b \in \mathbb{R}^+$:

$$\|\mathbf{e}(t)\| = 0 \implies \|\mathbf{y}(t) - \mathbf{1}\mathbf{v}(t)\| = 0 \text{ and } \|\mathbf{v}(t)\| \leqslant b.$$

d) *Unbounded dynamic* when it is not trivial and $\forall t \geqslant 0$, $\exists \mathbf{v} : \mathbb{R}_0^+ \longmapsto \mathbb{R}^q$:

$$\|\mathbf{e}(t)\| = 0 \implies \|\mathbf{y}(t) - \mathbf{1}\mathbf{v}(t)\| = 0 \text{ and } \lim_{t\to+\infty}\|\mathbf{v}(t)\| = \infty.$$

In the trivial case, the outputs of the systems are equal only because the systems in the network are asymptotically stable and not necessarily because of the control strategy. Simply defining a network with stable AAs, will result in trivial consensus for any initial condition when the input signals and perturbations are zero. Note that $\|\mathbf{y}\| \to 0$ always implies that $\|\mathbf{e}\| = \|\mathbf{Ty}\| \to 0$ and therefore the trivial case holds when additionally $\|\mathbf{e}\| \to 0 \implies \|\mathbf{y}\| \to 0$, as $t \to +\infty$.

The other cases are more interesting as then consensus is reached regardless of the kind of dynamics that the network presents. Static consensus refers to the case where the agents reach consensus to a constant point in the outputs space. In the dynamic cases, consensus is reached towards a non constant function of time. Note that the fundamental difference between bounded dynamic consensus and trivial consensus is that in the dynamic case, consensus may be reached long before the outputs approach to zero. Examples of these four definitions can be seen in Figure 4.3.

From these definitions, the dynamics of the error given by (4.6) become relevant for consensus analysis.

$$\dot{\mathbf{e}} = \mathbf{T}\dot{\mathbf{y}} = \mathbf{TC}\dot{\mathbf{x}} = \mathbf{TCAx} + \mathbf{TCBu} \tag{4.6}$$

Note that the consensus capacity of a network is strongly related to the properties of the feedback law chosen for the network but also to the dynamics of the states of the agents. We assume that $\operatorname{rank}\{\mathbf{CB}\} = Nq$ unless otherwise stated.

Observe that using the properties of an organization matrix, it can be written that $\mathbf{T}^+\mathbf{e} = \mathbf{T}^+\mathbf{Ty} = \mathbf{y} - \mathbf{Jy} = \mathbf{y} - \mathbf{JCx}$, and from here,

$$\mathbf{y} = \mathbf{T}^+\mathbf{e} + \mathbf{Jy} = \mathbf{T}^+\mathbf{e} + \mathbf{JCx} \tag{4.7}$$

In the general case, $\mathbf{y}$ cannot be written only as a function of $\mathbf{e}$ as matrix $\mathbf{T}$ is not square and, therefore, vector $\mathbf{e}$ is of lower dimension. Note that if $\mathbf{e}(t) = \mathbf{0}$, it holds that,

$$\mathbf{y}(t) = \mathbf{Jy}(t) = \frac{1}{N}\mathbf{1}\mathbf{1}'\mathbf{y}(t) \iff \mathbf{y}(t) = \mathbf{1v}(t),$$

with a (possibly) time dependent vector $\mathbf{v} : \mathbb{R}_0^+ \mapsto \mathbb{R}^q$. From here, if $\mathbf{e} = \mathbf{0}$, then it also holds that $\dot{\mathbf{e}} = \mathbf{T}\dot{\mathbf{y}} = \mathbf{T}(\mathbf{1}\dot{\mathbf{v}}) = (\mathbf{T1})\dot{\mathbf{v}} = \mathbf{0}$. Therefore, if a network reaches consensus, it stays in consensus regardless of the dynamic behavior of $\mathbf{v}$.

Example 4.2. To numerically evaluate the performance of a network regarding to how successfully it reaches consensus within a time interval $[0,t]$, the following indices can be defined considering the deviation of the outputs with respect to their mean value $\mathbf{e}_M := \mathbf{y} - \mathbf{1}\frac{1}{N}\sum_{i=1}^{M}[\mathbf{y}]_i = (\mathbf{I} - \mathbf{J})\mathbf{y}$:

$$ISD(t) = \int_0^t \sqrt{\mathbf{e}_M'\mathbf{e}_M}\, dt = \int_0^t \sqrt{\mathbf{y}'(\mathbf{I}-\mathbf{J})\mathbf{y}}\, dt,$$

$$IAD(t) = \int_0^t \sum_{i=1}^{N-1} |[\mathbf{e}_M]_i|\; dt.$$

Table 4.1.: Network simulation indicators for four different networks in Example 4.2.

Network	$ISD(3)$	$IAD(3)$
a)	0.859693	2.357284
b)	0.266529	0.730825
c)	0.242299	0.664386
d)	0.277635	0.761276

The abbreviation *ISD* stands for *Integral Square Deviation* and *IAD* for *Integral Absolute Deviation*. These indicators do not depend on the organization and they directly consider the evolution in time of the outputs of the systems. For given network and initial condition, a "good" consensus algorithm should imply a relatively small value for the defined indicators. Table 4.1 shows the simulation indicators evaluated at $t = 3$ for the four situations in Figure 4.3. ■

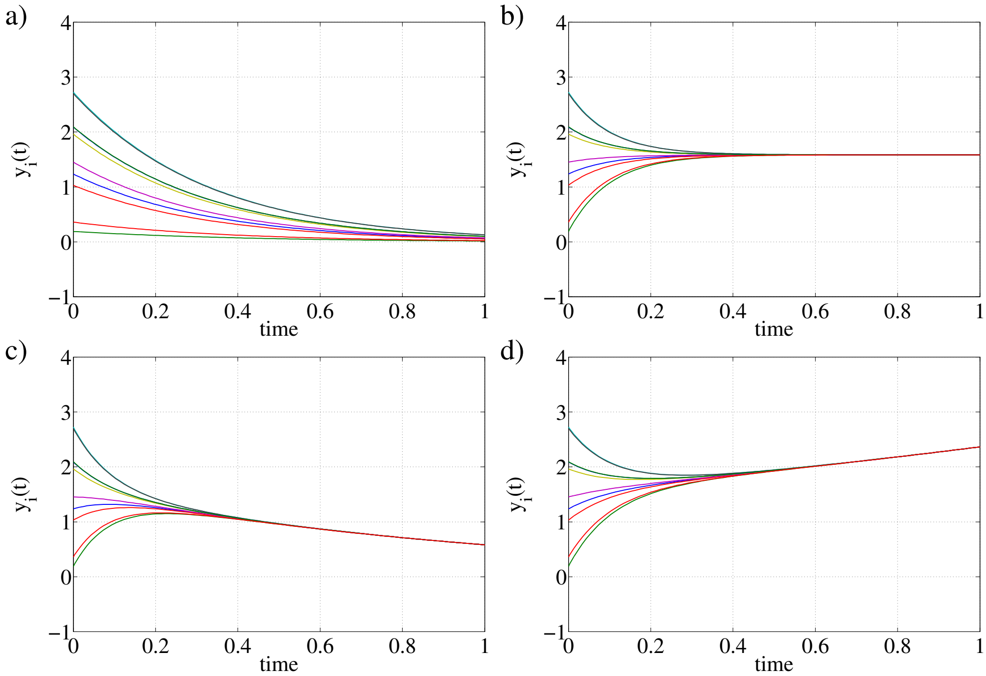


Figure 4.3.: Time trajectories examples for a) Trivial consensus, b) Static consensus, c) Bounded dynamic consensus, and d) Unbounded dynamic consensus.

5 Consensus Algorithms

Given a network and an organization, that is, a control plant and a control objective, a consensus algorithm is introduced as a feedback law for the network in order to achieve the consensus objective. We will concentrate on linear consensus algorithm defined as a proportional output feedback:

$$\mathbf{u} = \mathbf{L}\mathbf{y} \tag{5.1}$$

with $\mathbf{L} \in \mathbb{R}^{p \times Nq}$. In this work, only square consensus algorithms are considered with $p = Nq$. An external input $\mathbf{u}_{ext}$ can be considered to study the behavior of the network under other objectives considering $\mathbf{u} = \mathbf{L}\mathbf{y} + \mathbf{u}_{ext}$. However, unless contrary stated, for consensus analysis and without lost of generality this work considers that $\mathbf{u}_{ext} = \mathbf{0}$.

A consensus algorithm can be described by a block matrix in the shape of:

$$\mathbf{L} = \begin{bmatrix} -\Delta_1 & \mathbf{W}_{12} & \cdots & \mathbf{W}_{1N} \\ \mathbf{W}_{21} & -\Delta_2 & \cdots & \mathbf{W}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{W}_{N1} & \mathbf{W}_{N2} & \cdots & -\Delta_N \end{bmatrix}, \tag{5.2}$$

where the blocks $\mathbf{W}_{ij} = [\mathbf{L}]_{ij} \in \mathbb{R}^{q \times q}$, $i \neq j$, are gains that represents the weights with which the output of system $j \in \mathscr{V}$ is added to the input of agent $i \in \mathscr{V}$. The matrices $\Delta_i = [\mathbf{L}]_{ii} \in \mathbb{R}^{q \times q}$ are used to compute feedback signals from the output to the input of each system $i \in \mathscr{V}$. In general, a consensus algorithm described by the previous expression does not assume any further conditions over the blocks of the matrix. Equation (5.1) can be equivalently written with respect to the input signal $\mathbf{u}_i$ of each agent $i \in \mathscr{V}$:

$$\mathbf{u}_i = \sum_{j=1}^{N} [\mathbf{L}]_{ij} \mathbf{y}_j.$$

In the sequel, several special cases of interest are studied in terms of their ability to reach consensus.

5.1. Loopless Laplacian Algorithms

The most studied case is where the consensus algorithm is derived as the negative Laplacian matrix of an undirected weighted graph $\mathscr{G}_w$.

Definition 5.1.1. A *loopless negative Laplacian consensus algorithm* is a linear output feedback $\mathbf{u} = \mathbf{L}\mathbf{y}$ for network (4.3) where the feedback matrix is derived as $\mathbf{L} = -\hat{L}(\mathscr{G}_w) \in \mathbb{R}^{Nq \times Nq}$ with $\mathscr{G}_w$ an undirected weighted graph without selfloops over the vertices set $\mathscr{V}$.

A note on the interpretation of such an algorithm needs to be stated. While a graph $\mathscr{G}_w$ is commonly associated with physical properties of the communication links between the agents, here we simply regard it as a convenient description of the algorithm. In the same way, a strictly directed tree is a convenient description for an organization matrix. From the definition of algorithm as a linear feedback law, what is distributively implemented by the agents are the gains described by the block elements of matrix $\mathbf{L}$. Furthermore, the definition of the error $\mathbf{e}$ and of consensus as a stability problem, makes it natural to answer the question of consensus not through graph theoretical tools, but through algebraic properties of the involved matrices and general control theory. In this sense, a graph $\mathscr{G}_w$ becomes useful for the specification of different algorithms, but does not imply a particular specification of the underlaying communication processes.

Note that the feedback matrix is symmetric with the shape:

$$\mathbf{L} = -\hat{L}(\mathscr{G}_w) = \begin{bmatrix} -\Delta_1 & \mathbf{W}_{12} & \cdots & \mathbf{W}_{1N} \\ \mathbf{W}_{12} & -\Delta_2 & \cdots & \mathbf{W}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{W}_{1N} & \mathbf{W}_{2N} & \cdots & -\Delta_N \end{bmatrix},$$

where $\mathbf{W}_{ij} = \mathbf{W}'_{ij}$ and $\Delta_i = \sum_{j \neq i} \mathbf{W}_{ij}$ and therefore has the zero row sum property ($\mathbf{L}\mathbf{1} = \mathbf{0}$). Furthermore, it follows that $\mathbf{L}\mathbf{J} = -\hat{L}(\mathscr{G}_w)\mathbf{J} = -\frac{1}{N}\left(\hat{L}(\mathscr{G}_w)\mathbf{1}\right)\mathbf{1}' = \mathbf{0}$ and therefore, because of (4.7),

$$\mathbf{u} = \mathbf{L}\mathbf{y} = \mathbf{L}\left(\mathbf{T}^{+}\mathbf{e} + \mathbf{J}\mathbf{C}\mathbf{x}\right) = \mathbf{L}\mathbf{T}^{+}\mathbf{e}$$

For simplicity of notation, a matrix

$$\mathbf{H} = \mathbf{L}\mathbf{T}^{+} \in \mathbb{R}^{Nq \times (N-1)q} \tag{5.3}$$

can be defined. Note that $\mathbf{H}\mathbf{T} = \mathbf{L}\mathbf{T}^{+}\mathbf{T} = \mathbf{L}(\mathbf{I} - \mathbf{J}) = \mathbf{L}$ for Laplacian consensus algorithms as in Figure 5.1. With this, the dynamics of the error given by (4.6) for AAN can be presented as:

$$\dot{\mathbf{e}} = \mathbf{T}\mathbf{C}\mathbf{B}\mathbf{H}\mathbf{e} + \mathbf{T}\mathbf{C}\mathbf{A}\mathbf{x} \tag{5.4}$$

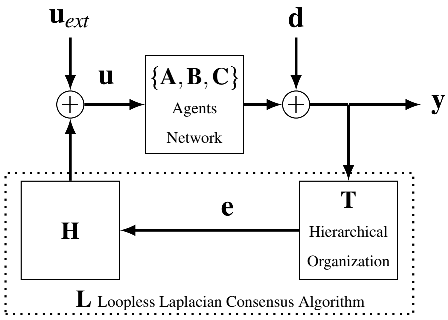


Figure 5.1.: Feedback configuration of an ANN with a Loopless Laplacian Algorithm including an external input $\mathbf{u}_{ext}$ and an output perturbation $\mathbf{d}$.

5.1.1. Integrators Network

In the case of networks composed only of integrators, as $\mathbf{A} = \mathbf{0}$, is immediate that the dynamics of the error depend only on the characteristics of the consensus algorithm in the following way:

$$\dot{\mathbf{e}} = \mathbf{TCBHe} \tag{5.5}$$

This simplifies the consensus problem greatly as it can be studied by simply analyzing the eigenvalues of matrix $\mathbf{G} := \mathbf{TCBH}$. Note that in the definitions of Section 4.1.2, it is assumed that the product $\mathbf{CB}$ is full rank.

Proposition 5.1. *In an IN with a loopless Laplacian algorithm, consensus is reached if and only if $\mathbf{G} := \mathbf{TCBH}$ is Hurwitz. That is, if and only if all its eigenvalues have a negative real part.*

Note that $\mathbf{CB} = \operatorname{diag}\{\mathbf{C}_i\mathbf{B}_i\}_{i\in\mathcal{V}}$ is block diagonal but not necessarily (element) diagonal. This makes it possible to expressly study agents with coupled input/output relationships.

Hurwitz Properties of TCBH

Equation (5.5) suggests that the dynamics of the error depend on the chosen organization. However this is not the case.

Lemma 5.2. *An IN with a loopless Laplacian algorithm reaches consensus if and only if the product $\mathbf{CBL} \in \mathbb{R}^{Nq\times Nq}$ has exactly $(N-1)q$ eigenvalues with negative real part and q zero eigenvalues.*

Proof. According to the Augmented Eigenvalues Proposition (see Proposition A.9 in the Appendix), as the organization matrix has more columns than rows,

$$\operatorname{eig}\{\mathbf{TCBH}\} \cup Z = \operatorname{eig}\left\{\mathbf{CBLT}^{+}\mathbf{T}\right\} = \operatorname{eig}\{\mathbf{CBL}\}.$$

Which does not depend on $\mathbf{T}$, and so $\mathbf{TCBH} \in \mathbb{R}^{(N-1)q\times(N-1)q}$ is Hurwitz only when the product $\mathbf{CBL}$ has $(N-1)q$ eigenvalues with negative real part and q zero eigenvalues. □

Note that $\dim\{\mathbf{TCBH}\} = (N-1)q$ and $\dim\{\mathbf{CBL}\} = Nq$, and then $|Z| = q$. It follows that:

Lemma 5.3. *If an IN with a loopless Laplacian algorithm reaches consensus, then the graph associated with the algorithm is connected.*

Proof. If $\mathbf{G} = \mathbf{TCBH}$ is Hurwitz, then $\operatorname{rank}\{\mathbf{G}\} = \dim\{\mathbf{G}\} = (N-1)q$. Hence, considering that $\mathbf{CBL}$ has in this case exactly $|Z| = q$ zero eigenvalues, $\operatorname{rank}\{\mathbf{CBL}\} = (N-1)q$. If it is assumed that $\operatorname{rank}\{\mathbf{CB}\} = Nq$ ($\mathbf{CB}$ is full rank), $\operatorname{rank}\{\mathbf{CBL}\} = \operatorname{rank}\{\mathbf{L}\} = (N-1)q$. When $\mathbf{L} = -\hat{L}(\mathscr{G}_w)$, from Lemma 2.2, this is only possible if $\mathscr{G}_w$ is connected. □

Observe that because of matrix $\mathbf{CB}$, the previous result is only a necessary condition and not sufficient. Unfortunately, in general, there is no relationship that links the eigenvalues of $\mathbf{CB}$ and $\mathbf{L}$ to the eigenvalues of $\mathbf{CBL}$ and therefore consensus cannot be studied as a property only of the chosen graph. Indeed, if the graph is connected so that the consensus algorithm has rank $(N-1)q$, we can only deduce that the matrix $\mathbf{G} = \mathbf{TCBH}$ is full rank, but nothing about the signs of the real parts of its eigenvalues. For example, if $\mathbf{CB} = -\mathbf{I}$ and a connected undirected graph $\mathscr{G}_w$ is used to define $\mathbf{L} = -\hat{L}(\mathscr{G}_w)$, consensus cannot be reached as all eigenvalues of $\mathbf{G}$ are located on the right complex half-plane.

Only in special cases Lemma 5.3 is valid in both directions. In particular, when $\mathbf{CB} = \mathbf{I}$, then $\operatorname{eig}\{\mathbf{TCBLT}^+\} \cup Z = \operatorname{eig}\{\mathbf{LT}^+\mathbf{T}\} = \operatorname{eig}\{\mathbf{L}\}$ which implies that $\mathbf{G} = \mathbf{TCBH}$ is Hurwitz as $\hat{L}(\mathscr{G}_w) \geqslant 0$ and $\operatorname{rank}\{\mathbf{L}\} = (N-1)q$ when $\mathscr{G}_w$ is connected.

Remark 5.1. Note that in the case that the consensus transformation were a square matrix or a matrix with more rows than columns, *i.e.* not derived from a tree but from a digraph with more than $N-1$ edges ($\mathbf{T} = D'(\mathscr{G}^o) \in \mathbb{R}^{r\times Nq}$), this condition could not be satisfied. As $\operatorname{eig}\{\mathbf{TCBH}\} = \operatorname{eig}\{\mathbf{CBHT}\} \cup Z$ whenever $r > Nq$. So $\mathbf{G} = \mathbf{TCBH}$ would have at least $r - Nq$ zero eigenvalues and cannot be Hurwitz. Thus, it is harder to study consensus considering more than $N-1$ relationships, as suggested by the definition in Section 4.2.1.

Another interesting result has to do with the kind of consensus that is reached.

Lemma 5.4. *If consensus is reached in an IN with a Loopless Laplacian algorithm, then it is static consensus.*

Proof. As $\mathbf{y} = \mathbf{T}^+\mathbf{e} + \mathbf{Jy}$, when $\mathbf{e} = \mathbf{0}$, then $\mathbf{y} = \mathbf{Jy} \Longleftrightarrow \mathbf{y} = \mathbf{1v}(t)$, where $\mathbf{v}(t) : \mathbb{R}_0^+ \longmapsto \mathbb{R}^q$. The dynamics of the outputs in a IN can be written as: $\dot{\mathbf{y}} = \mathbf{CBLy}$. Evaluating when consensus is reached, then $\dot{\mathbf{y}} = \mathbf{1}\dot{\mathbf{v}} = -\mathbf{CB}\left(\hat{L}(\mathscr{G}_w)\mathbf{1}\right)\mathbf{v}(t) = \mathbf{0}$, and therefore $\dot{\mathbf{v}} = \mathbf{0}$ and $\mathbf{v}(t) = \mathbf{v}$ is constant. □

Furthermore, $\mathbf{v} \in \mathbb{R}^q$ is not necessarily zero.

Stability of the consensus targets can be studied as an LMI feasibility problem.

Lemma 5.5. *Consensus in an IN with a loopless Laplacian algorithm is achieved if and only if a matrix* $\mathbf{P} = \mathbf{P}' > 0$ *exists such that*

$$\mathbf{PTCBH} + \mathbf{H}'\mathbf{B}'\mathbf{C}'\mathbf{T}'\mathbf{P} < 0. \tag{5.6}$$

Proof. The same as Theorem 2.10 with system matrix $\mathbf{G} = \mathbf{TCBH}$. □

Whenever the consensus algorithm is known, the previous expression can be used to check if the network reaches consensus by numerically solving the feasibility problem of LMI (5.6) with $\mathbf{P} = \mathbf{P}' > 0$. However, this inequality cannot be directly used to design different algorithms for a given network due to the product of variable matrices $\mathbf{P}$ and $\mathbf{H}$. Here, there are two possible ways to proceed. Either fix matrix $\mathbf{P}$ to a known value to design an arbitrary shaped loopless Laplacian algorithm, or restrict the algorithm to a have a tree structure. In the next sections, these two ways to design algorithms are presented by considering different performance criteria based on the previous matrix inequality.

Algorithm Rate of Convergence

From classical control theory, it is known that the convergence rate $\lambda > 0$ at which the agents reach consensus is given by the eigenvalues of the Hurwitz matrix $\mathbf{G} = \mathbf{TCBH}$. Namely, $\lambda := \min\{\text{real}\{\text{eig}\{-\mathbf{G}\}\}\} > 0$. As shown before with the help of Proposition A.9, these eigenvalues for an algorithm derived from a connected graph are the same as the non-zero eigenvalues of the product $\mathbf{CBL}$. In the usual consensus formulation for single integrator systems, $\mathbf{CB} = \mathbf{I}_{N\times N}$. Therefore, in this special case, the spectrum of the Laplacian matrix defines the consensus characteristics of the network. Particularly, the algebraic connectivity of the graph coincides with the convergence rate. Furthermore, because of Lemma 2.3, additional edges between the agents will speed up consensus.

Using remark 2.1, the following matrix inequality is always fulfilled for some $\mathbf{P} = \mathbf{P}' > 0$:

$$\mathbf{PG} + \mathbf{G}'\mathbf{P} + 2\lambda\mathbf{P} \leqslant 0 \tag{5.7}$$

Consider an undirected loopless graph $\mathscr{G}$ over the vertices $\mathscr{V}$ with a set of edges $\mathscr{E}$ and an unknown function of symmetric weights $w_q : \mathscr{E} \to \mathbb{R}^{q\times q}$. The Laplacian matrix of the weighted graph $\mathscr{G}_w = (\mathscr{G}, w_q)$ is then $\hat{L}(\mathscr{G}_w) = \mathbf{E}'\mathbf{W}\mathbf{E}$, where $\mathbf{E} = D'(\mathscr{G}^o) \in \mathbb{R}^{|\mathscr{E}|q\times Nq}$ and $\mathbf{W} = \text{diag}\{\mathbf{W}_{i_k j_k}\}_{k=1}^{|\mathscr{E}|} \in \mathbb{R}^{|\mathscr{E}|q\times|\mathscr{E}|q}$ is a block diagonal symmetric matrix of weights for each edge. Imposing $\mathbf{L} = -\hat{L}(\mathscr{G}_w)$, inequality (5.7) leads to:

$$-\mathbf{PTCBE}'\mathbf{WET}^+ - (\mathbf{T}^+)'\mathbf{E}'\mathbf{WEB}'\mathbf{C}'\mathbf{T}'\mathbf{P} + 2\lambda\mathbf{P} \leqslant 0 \tag{5.8}$$

The previous expression is not an LMI because of the multiplication of variables $\mathbf{P}$ and $\mathbf{W}$.

The linearization, in term of its variables, of an expression as (5.8) is usually done by pre- and post-multiplication of the condition by a full rank transformation $\mathbf{Q} = \mathbf{P}^{-1} > 0$ (See Proposition A.8 in the Appendix). An equivalent condition is thus obtained:

$$-\mathbf{TCBE}'\mathbf{WET}^+\mathbf{Q} - \mathbf{Q}(\mathbf{T}^+)'\mathbf{E}'\mathbf{WEB}'\mathbf{C}'\mathbf{T}' + 2\lambda\mathbf{Q} \leqslant 0$$

Defining an auxiliary variable $\mathbf{Z} = \mathbf{WET}^{+}\mathbf{Q}$ leads to an LMI on variables $\mathbf{Z}$ and $\mathbf{Q}$. However, the feasibility of this inequality can be used to design the weights of an algorithm, only if $\mathbf{W}$ is block diagonal and there is a bijective relationship between matrices $\mathbf{W}$ and $\mathbf{Z}$, *i.e.* only if $\det\{\mathbf{ET}^{+}\} \neq 0$ so that $\mathbf{W} = \mathbf{ZQ}^{-1}(\mathbf{ET}^{+})^{-1} \iff \mathbf{Z} = \mathbf{WET}^{+}\mathbf{Q}$. This is in general not possible, even forcing a special structures on $\mathbf{Q}$ and $\mathbf{Z}$, as the product $\mathbf{ET}^{+}$ is neither square nor full rank.

Note however that in the special case when the selected graph for the algorithm is the same as the one from where the organization is derived, *i.e.* when $\mathscr{G} = \mathscr{T}$, then we can choose $\mathbf{E} = \mathbf{T}$ and so $\mathbf{ET}^{+} = \mathbf{I}$. Therefore, this procedure can be used to design consensus algorithms described by the same tree as the organization and imposing matrices $\mathbf{Q}$ and $\mathbf{Z}$ to be block diagonal. This can be formalized in the following theorem.

Theorem 5.6. *Given an IN, a tree $\mathscr{T} = (\mathscr{V}, \{e_1, \cdots, e_{N-1}\})$, a corresponding organization matrix $\mathbf{T} = D'(\mathscr{T}^o) \in \mathbb{R}^{(N-1)q \times Nq}$, and a scalar $\lambda > 0$, a consensus algorithm described by $\mathbf{L} = -\hat{L}((\mathscr{T}, w_q)) = -\mathbf{T}'\mathbf{WT}$ such that $\min\{\text{real}\{\text{eig}\{-\mathbf{G}\}\}\} < \lambda$, can be designed if LMI (5.9) is feasible over the structured variable $\mathbf{Q} = \text{diag}\{\rho_i \mathbf{I}_{q\times q}\}_{i=1}^{N-1} > 0$, with $N-1$ scalars $\rho_i > 0$; and the block diagonal symmetric variable $\mathbf{Z} = \mathbf{Z}' = \text{diag}\{\mathbf{Z}_i\}_{i=1}^{N-1} \in \mathbb{R}^{(N-1)q\times(N-1)q}$. In that case, $\mathbf{W} = \mathbf{ZQ}^{-1}$ and each diagonal block $[\mathbf{W}]_{kk} = w_q(e_k) \in \mathbb{R}^{q\times q}$ represents the weight of the corresponding k-th edge of the tree.*

$$-\mathbf{TCBT}'\mathbf{Z} - \mathbf{ZTB}'\mathbf{C}'\mathbf{T}' + 2\lambda\mathbf{Q} < 0 \tag{5.9}$$

Proof. $\min\{\text{real}\{\text{eig}\{-\mathbf{G}\}\}\} < \lambda$ if and only if $\mathbf{PG} - \mathbf{G}'\mathbf{P} + 2\lambda\mathbf{P} < 0$ with $\mathbf{P} = \mathbf{P}' > 0$. Pre- and post- multiply this condition by a full rank symmetric congruence transformation $\mathbf{Q} = \mathbf{P}^{-1} > 0$ and replacing $\mathbf{HQ} = \mathbf{LT}^{+}\mathbf{Q} = -\mathbf{T}'\mathbf{WTT}^{+}\mathbf{Q} = -\mathbf{T}'\mathbf{Z}$, leads to LMI (5.9). The special structure of matrix $\mathbf{Q} = \text{diag}\{\rho_i \mathbf{I}_{q\times q}\}_{i=1}^{N-1} > 0$ is needed so that the product $\mathbf{W} = \mathbf{ZQ}^{-1}$ is block diagonal and symmetric. □

Unfortunately, arbitrary shaped consensus algorithms that guarantee certain known value of $\lambda > 0$ cannot be directly designed. In that case, other strategy is to impose

$$\frac{d}{dt}\|\mathbf{e}\| < -\varsigma\|\mathbf{e}\|, \tag{5.10}$$

where the scalar $\varsigma > 0$ represents the convergence rate of the consensus error error. As $\|\mathbf{e}\| = \sqrt{\mathbf{e}'\mathbf{e}}$, then (5.10) is equivalent to

$$\mathbf{e}'\dot{\mathbf{e}} + \dot{\mathbf{e}}'\mathbf{e} < -2\varsigma\mathbf{e}'\mathbf{e}.$$

From the dynamics in (5.5), the previous condition leads to the following matrix inequality:

$$\mathbf{TCBH} + \mathbf{H}'\mathbf{B}'\mathbf{C}'\mathbf{T}' + 2\varsigma\mathbf{I} < 0 \tag{5.11}$$

Note that this expression is equivalent to impose negativity of the derivate of a positive Lyapunov function $v = \mathbf{e}'\mathbf{P}\mathbf{e} + 2\varsigma \int_0^t \mathbf{e}'\mathbf{P}\mathbf{e}dt$, with $\mathbf{P} = \mathbf{I}$. When restricting $\mathbf{P}$ to the identity, equation (5.10) has a direct intuitive interpretation. However, this stability condition might be too restrictive for certain networks as a very particular Lyapunov function is imposed. That is, it might be the case that the matrix $\mathbf{TCBH} + \mathbf{H}'\mathbf{B}'\mathbf{C}'\mathbf{T}'$ is not negative definite, even when $\mathbf{TCBH}$ is Hurwitz. By defining a norm $\|\mathbf{e}\|_P = \sqrt{\mathbf{e}'\mathbf{P}\mathbf{e}}$, with a fixed matrix $\mathbf{P} = \mathbf{P}' > 0$, similar, but potentially less restrictive, stability conditions may be found.

Considering the Schur's Complement (See Proposition A.12 in Appendix), inequality (5.11) is equivalent to

$$\begin{bmatrix} \mathbf{TCBH} + \mathbf{H}'\mathbf{B}'\mathbf{C}'\mathbf{T}' & \mathbf{I} \\ \star & -\varepsilon\mathbf{I} \end{bmatrix} < 0, \tag{5.12}$$

with $\varepsilon = 1/(2\varsigma)$. From here, it is immediate that the consensus converge rate of the norm related to an organization $\mathbf{T}$ of an IN with a consensus algorithm described by $\mathbf{L} = \mathbf{HT}$, can be calculated as $\varsigma = 1/(2\varepsilon_{min})$ where $\varepsilon_{min} > 0$ is obtained from a convex minimization problem $\varepsilon_{min} = \inf\{\varepsilon > 0 | \text{LMI } (5.12)\}$.

This result can be used to compare the performance of different algorithms subject to the same organization. However, using different organizations will, in general, lead to different values of ς for the same consensus algorithm. Therefore the following results become relevant.

Lemma 5.7. *All the organizations derived from a tree $\mathscr{T}$, independently of the orientations and labels given to the edges, can be used to obtain the same value $\varsigma > 0$ for a given algorithm $\mathbf{L} = -\hat{L}(\mathscr{G}_w)$.*

Proof. Given an organization $\mathbf{T}_1 = D'(\mathscr{T}^o)$, then every other organization derived from the same tree can be expressed as $\mathbf{T}_2 = \mathbf{MT}_1$, where $\mathbf{M}'\mathbf{M} = \mathbf{MM}' = \mathbf{I}$ ($\mathbf{M}$ is either a permutation matrix or a matrix with positive and negative ones in the diagonal, see Remark 4.1), and therefore $\mathbf{T}_2^+ = \mathbf{T}_1^+\mathbf{M}'$. Then,

$$\begin{aligned} & \mathbf{T}_1\mathbf{CBLT}_1^+ + (\mathbf{T}_1^+)'\mathbf{L}'\mathbf{B}'\mathbf{C}'\mathbf{T}_1' + 2\varsigma\mathbf{I} < 0 \\ \Longleftrightarrow\ & \mathbf{M}\left(\mathbf{T}_1\mathbf{CBLT}_1^+ + (\mathbf{T}_1^+)'\mathbf{L}'\mathbf{B}'\mathbf{C}'\mathbf{T}_1' + 2\varsigma\mathbf{I}\right)\mathbf{M}' < 0 \\ \Longleftrightarrow\ & \mathbf{T}_2\mathbf{CBLT}_2^+ + (\mathbf{T}_2^+)'\mathbf{L}'\mathbf{B}'\mathbf{C}'\mathbf{T}_2' + 2\varsigma\mathbf{I} < 0. \end{aligned}$$

$\square$

Note that, for any organization and any consensus algorithm, if it is possible to obtain the value of $\varsigma = \sup\{\varsigma > 0 | \text{LMI } (5.11)\}$ then it holds that $\mathbf{TCBH} + \mathbf{H}'\mathbf{B}'\mathbf{C}'\mathbf{T}' + 2\varsigma\mathbf{I} \leqslant 0$. This can be rewritten as $\mathbf{I}(\varsigma\mathbf{I} + \mathbf{TCBH}) + (\varsigma\mathbf{I} + \mathbf{TCBH})'\mathbf{I} \leqslant 0$. This implies that the real parts of the eigenvalues of $\mathbf{TCBH} + \varsigma\mathbf{I}$ are non positive. From here, $\lambda = \min\{\text{real}\{\text{eig}\{-\mathbf{TCBH}\}\}\} \geqslant \varsigma$. Therefore, designing an algorithm based on LMI (5.11) will guarantee that the convergence rate of the error λ is not smaller than the design value ς.

To design algorithms that fulfill certain convergence rate condition described by an arbitrary weighted connected graph $\mathscr{G}_w = (\mathscr{G}, w)$ the following theorem can be stated:

Theorem 5.8. *Given an IN, an organization* $\mathbf{T} = D'(\mathscr{T}^o)$*, a consensus convergence rate value* $\varsigma_d > 0$*, and a connected graph* $\mathscr{G} = (\mathscr{V}, \mathscr{E} = \{e_1, \cdots, e_{|\mathscr{E}|}\})$*, a consensus algorithm described by* $\mathbf{L} = -\mathbf{E}'\mathbf{W}\mathbf{E}$*, with* $\mathbf{E} = D'(\mathscr{G}^o) \in \mathbb{R}^{|\mathscr{E}|q \times Nq}$*, such that*

$$\frac{d}{dt}\|\mathbf{e}\| < -\varsigma_d\|\mathbf{e}\|,$$

can be designed if LMI (5.13) is feasible over the block diagonal symmetric variable $\mathbf{W} = \operatorname{diag}\{\mathbf{W}_{i_k j_k}\}_{k=1}^{|\mathscr{E}|} \in \mathbb{R}^{|\mathscr{E}|q \times |\mathscr{E}|q}$.

$$-\mathbf{TCBE}'\mathbf{WET}^+ - (\mathbf{T}^+)'\mathbf{E}'\mathbf{WEB}'\mathbf{C}'\mathbf{T}' + 2\varsigma_d\mathbf{I} < 0 \tag{5.13}$$

Proof. Evaluating $\mathbf{L} = -\mathbf{E}'\mathbf{W}\mathbf{E}$ in inequality (5.11) leads directly to the result. □

Note that this design procedure involves only solving a feasibility problem and not a minimization. Furthermore, the imposed condition might be very restrictive due to the particular Lyapunov function considered.

Algorithm Sensitivity to External Signals

A network cannot only be thought of as an isolated system with no interactions with external signals. Perturbations like external inputs to each agent or output bias need to be considered when analyzing a particular algorithm. This can be done by considering the H_∞-norm of the system.

A general engineering interpretation of H_∞-norm associates it with the largest input-output gain of a system through all frequencies. That is, the maximum factor by which the magnitude of an (uncontrolled) input vector $\mathbf{v}$ is amplified by the network. In fact, if the error vector $\mathbf{e}$ approaches the origin,

$$\frac{\left(\int_0^{+\infty} \mathbf{e}(t)'\mathbf{e}(t)dt\right)^{1/2}}{\left(\int_0^{+\infty} \mathbf{v}(t)'\mathbf{v}(t)dt\right)^{1/2}} \leqslant \|H_{ve}(s)\|_\infty < +\infty.$$

Where $\mathbf{v}$ is some external signal to be defined and $H_{ve}(s)$ the transfer function matrix from this input to the consensus error vector. From here $\gamma_v = \|H_{ve}(s)\|_\infty$ can be interpreted as a sensitivity measure of the algorithm against the effect of $\mathbf{v}$. This value can be calculated with the help of the Bounded Real Lemma (BRL, see Lemma A.13 in Appendix). Here, this will be done in two cases.

First consider the external signal $\mathbf{u}_{ext} : \mathbb{R}_0^+ \mapsto \mathbb{R}^{Nq}$ as in Figure 4.1. In that case, $\mathbf{u} = \mathbf{Ly} + \mathbf{u}_{ext}$ and so the dynamics of the error of the IN are modified to obtain:

$$\dot{\mathbf{e}} = \mathbf{TCBHe} + \mathbf{TCBu}_{ext} \tag{5.14}$$

Considering $\mathbf{e}$ as the state and output variable and $\mathbf{u}_{ext}$ as the perturbation signal, directly applying the BRL over the system (5.14) leads to the following matrix inequality:

$$\begin{bmatrix} \mathbf{PTCBH}+\mathbf{H}'\mathbf{B}'\mathbf{C}'\mathbf{T}'\mathbf{P}+\mathbf{I} & \mathbf{PTCB} \\ \star & -\gamma^2\mathbf{I} \end{bmatrix} < 0 \tag{5.15}$$

Therefore, given a weighted graph $\mathscr{G}_w$ and the associated consensus algorithm $\mathbf{L} = -\hat{L}(\mathscr{G}_w)$; and given an organization $\mathbf{T} = D'(\mathscr{T}^o)$, the sensitivity of the algorithm against additive external inputs can be measured by a scalar $\gamma_{ext} > 0$ that can be calculated by a convex minimization problem $\gamma_{ext}^2 = \inf\{\mu = \gamma^2 \in \mathbb{R}^+ | \text{LMI (5.15)} \wedge \mathbf{P} = \mathbf{P}' > 0\}$.

In general, in terms of analysis, this way to obtain the H_∞-norm of the system is not numerically the most efficient one[1]. Nevertheless, the introduction of LMI (5.15) gives the possibility of designing algorithms that guarantee certain H_∞ performance in a simple way.

Indeed, considering a connected graph $\mathscr{G}$, an organization $\mathbf{T} = D'(\mathscr{T}^o)$, a given positive definite matrix $\mathbf{P} \in \mathbb{R}^{(N-1)q\times(N-1)q}$ and a scalar $\gamma > 0$, it is immediately clear that a consensus algorithm described by $\mathbf{L} = -\mathbf{E}'\mathbf{W}\mathbf{E}$, with $\mathbf{E} = D'(\mathscr{G}^o) \in \mathbb{R}^{|\mathscr{E}|q\times Nq}$ such that $\|H_{ext}(s)\|_\infty < \gamma$, can be designed if LMI (5.15) is feasible over the symmetric block diagonal variable $\mathbf{W} = \operatorname{diag}\{\mathbf{W}_{i_k j_k}\}_{k=1}^{|\mathscr{E}|} \in \mathbb{R}^{|\mathscr{E}|q\times|\mathscr{E}|q}$ with $\mathbf{H} = -\mathbf{E}'\mathbf{W}\mathbf{E}\mathbf{T}^+$.

This result gives the possibility of designing consensus algorithms with any given structure but imposing particular Lyapunov functions that might be too restrictive. However, if consensus algorithms with a tree structure only are searched, the following theorem is helpful.

Theorem 5.9. *Given an IN, a tree $\mathscr{T} = (\mathscr{V}, \{e_1, \cdots, e_{N-1}\})$, a corresponding organization matrix $\mathbf{T} = D'(\mathscr{T}^o) \in \mathbb{R}^{(N-1)q\times Nq}$, and a scalar $\gamma > 0$, a consensus algorithm described by $\mathbf{L} = -\hat{L}((\mathscr{T}, w_q)) = -\mathbf{T}'\mathbf{W}\mathbf{T}$ such that $\|H_{ext}(s)\|_\infty < \gamma$, can be designed if LMI (5.16) is feasible over the structured variable $\mathbf{Q} = \operatorname{diag}\{\rho_i \mathbf{I}_{q\times q}\}_{i=1}^{N-1} > 0$, with $N-1$ scalars $\rho_i > 0$; and the block diagonal symmetric variable $\mathbf{Z} = \mathbf{Z}' = \operatorname{diag}\{\mathbf{Z}_i\}_{i=1}^{N-1} \in \mathbb{R}^{(N-1)q\times(N-1)q}$. In that case, $\mathbf{W} = \mathbf{Z}\mathbf{Q}^{-1}$ and each diagonal block $[\mathbf{W}]_{kk} = w_q(e_k) \in \mathbb{R}^{q\times q}$ represents the weight of the corresponding k-th edge of the tree.*

$$\begin{bmatrix} -\mathbf{TCBT}'\mathbf{Z} - \mathbf{Z}'\mathbf{TB}'\mathbf{C}'\mathbf{T}' & \mathbf{TCB} & \mathbf{Q} \\ \star & -\gamma^2\mathbf{I} & \mathbf{0} \\ \star & \star & -\mathbf{I} \end{bmatrix} < 0 \tag{5.16}$$

Proof. Pre- and post- multiply LMI (5.15) by a full rank symmetric congruence transformation $\operatorname{diag}\{\mathbf{Q}, \mathbf{I}\} \in \mathbb{R}^{(2N-1)q\times(2N-1)q}$ with $\mathbf{Q} = \mathbf{P}^{-1} > 0$. Applying Schur complement over the term $\mathbf{QQ}$ and replacing $\mathbf{HQ} = \mathbf{LT}^+\mathbf{Q} = -\mathbf{T}'\mathbf{WTT}^+\mathbf{Q} = -\mathbf{T}'\mathbf{Z}$ leads to LMI (5.16). The special structure of matrix $\mathbf{Q} = \operatorname{diag}\{\rho_i \mathbf{I}_{q\times q}\}_{i=1}^{N-1} > 0$ is needed so that the product $\mathbf{W} = \mathbf{Z}\mathbf{Q}^{-1}$ is symmetric. □

[1] In fact, the Matlab command `norm` applied over a state space model constructed through function `ss`, usually gives faster answers for big systems. This is related to the additional decisions variables in $\mathbf{P}$ that also need to be determined, slowing the overall computation and compromising its precision.

Consider now that the IN is perturbed by a signal $\mathbf{d}: \mathbb{R}_0^+ \mapsto \mathbb{R}^{Nq}$ so that $\mathbf{y} = \mathbf{Cx} + \mathbf{d}$. Given the integrator dynamics and a Laplacian algorithm, then the dynamics of the error become:

$$\dot{\mathbf{e}} = \mathbf{TCBHe} + \mathbf{Tw} \tag{5.17}$$

where $\mathbf{w} = \frac{d}{dt}\mathbf{d}$. Note that the error dynamics do not depend on the actual value of the perturbation $\mathbf{d}$, but on its change rate over time $\mathbf{w}$. That is, constant output perturbations do not have an influence in reaching consensus. However, as in this case $\dot{\mathbf{x}} = (\mathbf{A} + \mathbf{BLC})\mathbf{x} + \mathbf{Bd}$, output perturbations do affect the value of the states of the network, and therefore, the value at which consensus will be reached.

Nevertheless, in some cases it might be meaningful to analyze the effect of time varying perturbations on the consensus error. This can also be done using a H_∞ argument. Applying the BRL to (5.17), the following inequality is obtained.

$$\begin{bmatrix} \mathbf{PTCBH} + \mathbf{H'B'C'T'P} & \mathbf{PT} \\ \star & -\gamma^2\mathbf{I} \end{bmatrix} < 0 \tag{5.18}$$

This expression can be again used as a comparison tool to calculate the H_∞-norm $\gamma_w > 0$ of the transfer function matrix $H_{we}(s)$ from vector $\mathbf{w}$ to the consensus error $\mathbf{e}$ by solving the convex optimization problem: $\gamma_w^2 = \inf\left\{\mu = \gamma^2 \in \mathbb{R}^+ | \text{LMI (5.18)} \wedge \mathbf{P} = \mathbf{P}' > 0\right\}$.

For design of algorithms, analog as in the previous case, considering a connected graph $\mathscr{G}$, an organization $\mathbf{T} = D'(\mathscr{T}^o)$, a given positive definite matrix $\mathbf{P} \in \mathbb{R}^{(N-1)q\times(N-1)q}$ and a scalar $\gamma > 0$, is clear that a consensus algorithm described by $\mathbf{L} = -\mathbf{E'WE}$, with $\mathbf{E} = D'(\mathscr{G}^o) \in \mathbb{R}^{|\mathscr{E}|q\times Nq}$ such that $\|H_{we}(s)\|_\infty < \gamma$, can be designed if LMI (5.18) is feasible over variable $\mathbf{W} = \operatorname{diag}\{\mathbf{W}_{i_k j_k}\}_{k=1}^{|\mathscr{E}|} \in \mathbb{R}^{|\mathscr{E}|q\times|\mathscr{E}|q}$ with $\mathbf{H} = -\mathbf{E'WET}^+$. Furthermore, in the case of algorithms described by trees, the following procedure is analog to Theorem 5.9:

Theorem 5.10. *Given an IN, a tree $\mathscr{T} = (\mathscr{V}, \{e_1, \cdots, e_{N-1}\})$, a corresponding organization matrix $\mathbf{T} = D'(\mathscr{T}^o) \in \mathbb{R}^{(N-1)q\times Nq}$, and a scalar $\gamma > 0$, a consensus algorithm described by $\mathbf{L} = -\hat{L}((\mathscr{T}, w_q)) = -\mathbf{T'WT}$ such that $\|H_{we}(s)\|_\infty < \gamma$, can be designed if LMI (5.19) is feasible over the structured variable $\mathbf{Q} = \operatorname{diag}\{\rho_i \mathbf{I}_{q\times q}\}_{i=1}^{N-1} > 0$, with $N-1$ scalars $\rho_i > 0$; and the block diagonal symmetric variable $\mathbf{Z} = \mathbf{Z}' = \operatorname{diag}\{\mathbf{Z}_i\}_{i=1}^{N-1} \in \mathbb{R}^{(N-1)q\times(N-1)q}$. In that case, $\mathbf{W} = \mathbf{ZQ}^{-1}$ and each diagonal block $[\mathbf{W}]_{kk} = w_q(e_k) \in \mathbb{R}^{q\times q}$ represents the weight of the corresponding k-th edge of the tree.*

$$\begin{bmatrix} -\mathbf{TCBT'Z} - \mathbf{Z'TB'C'T'} & \mathbf{T} & \mathbf{Q} \\ \star & -\gamma^2\mathbf{I} & \mathbf{0} \\ \star & \star & -\mathbf{I} \end{bmatrix} < 0 \tag{5.19}$$

Proof. Analogous to Theorem 5.9. □

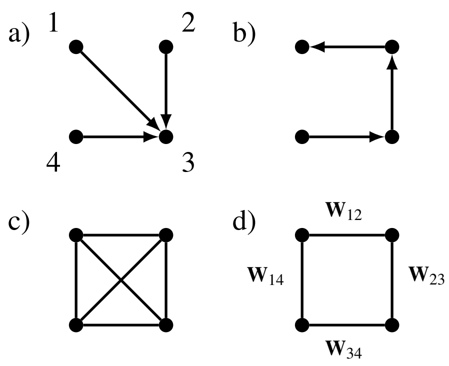


Figure 5.2.: Organizations and Consensus Algorithms for Example 5.1.

Example 5.1. Consider $N = 4$ integrator systems with, $\forall i \in \{1,2,3,4\}$,

$$\mathbf{C}_i\mathbf{B}_i = \begin{bmatrix} 1.0 & 0.5 \\ 0.2 & 1.0 \end{bmatrix}.$$

That is, coupled integrator systems over two dimensions. We will study the network through organizations described by the directed trees of Figure 5.2 a) and b); and the corresponding matrices:

$$\mathbf{T}_a = \begin{bmatrix} -\mathbf{I} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & -\mathbf{I} \end{bmatrix}, \; \mathbf{T}_b = \begin{bmatrix} \mathbf{I} & -\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & -\mathbf{I} \end{bmatrix}.$$

The fully connected algorithm derived from the graph in Figure 5.2 c) with the corresponding matrix:

$$\mathbf{L}_1 = \begin{bmatrix} -3\mathbf{I} & \mathbf{I} & \mathbf{I} & \mathbf{I} \\ \mathbf{I} & -3\mathbf{I} & \mathbf{I} & \mathbf{I} \\ \mathbf{I} & \mathbf{I} & -3\mathbf{I} & \mathbf{I} \\ \mathbf{I} & \mathbf{I} & \mathbf{I} & -3\mathbf{I} \end{bmatrix},$$

reaches consensus faster than any other algorithm derived from an unweighted graph due to the larger number of edges. Using Theorem 5.8 with a structure given by the graph in Figure 5.2 d), organization $\mathbf{T}_a$ and a fixed value $\varsigma_d = 4.00$, the following matrix can be obtained:

$$\mathbf{L}_2 = \begin{bmatrix} -\mathbf{W}_{12} - \mathbf{W}_{14} & \mathbf{W}_{12} & \mathbf{0} & \mathbf{W}_{14} \\ \mathbf{W}_{12} & -\mathbf{W}_{12} - \mathbf{W}_{23} & \mathbf{W}_{23} & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_{23} & -\mathbf{W}_{23} - \mathbf{W}_{34} & \mathbf{W}_{34} \\ \mathbf{W}_{14} & \mathbf{0} & \mathbf{W}_{34} & -\mathbf{W}_{14} - \mathbf{W}_{34} \end{bmatrix},$$

Table 5.1.: Consensus algorithms performance in Example 5.1.

				$\mathbf{T}_a$			$\mathbf{T}_b$		
	λ	$ISD(8)$	$IAD(8)$	ς	γ_{ext}	γ_w	ς	γ_{ext}	γ_w
$\mathbf{L}_1$	2.735089	8.567	19.861	2.600000	0.505594	0.756215	2.600000	0.467108	0.698652
$\mathbf{L}_2$	8.292125	2.536	5.836	7.168781	0.158732	0.227251	6.888741	0.122408	0.175098

with

$$\mathbf{W}_{12} = \mathbf{W}_{14} = \begin{bmatrix} 6.8675 & 0.0000 \\ 0.0000 & 6.1513 \end{bmatrix} \text{ and } \mathbf{W}_{23} = \mathbf{W}_{34} = \begin{bmatrix} 6.0663 & 0.0000 \\ 0.0000 & 5.4823 \end{bmatrix}.$$

The performance indicators γ_{ext}, γ_w and ς, are computed for both organizations in Table 5.1. From the table is clear that the designed weighted consensus algorithm is faster than the fully connected one. Furthermore, it is also less sensitive to external signals.

This can be seen in Figure 5.3 where the response in time of both outputs of the agents are drawn separately for the same initial conditions. At $t = 2$ the outputs of the agents are directly perturbed by a random vector $\mathbf{d}$ that stays constant until the end of the simulation time. Between $t = 4$ and $t = 6$ a random external input $\mathbf{u}_{ext}$ is added to the systems. From the simulation, it is clear that consensus is reached in less time by the algorithm defined by $\mathbf{L}_2$. In presence of the outputs perturbation $\mathbf{d}$ the consensus value is changed, but the the algorithms still ensure consensus when this signal is constant. The external inputs modify the consensus value constantly making the outputs of the systems to have different values. However, the difference between these values in the case of the fully connected algorithm is also larger. This is coherent with the simulation indicators *ISD* and *IAD* in Table 5.1 which show that the designed algorithm presents a lower accumulated deviation. ■

Algorithm Robustness Against Parametric Uncertainties

Consider now that the parameters of the network are known only up to certain precision. In particular we consider that $\mathbf{B} = \mathbf{B}_0 + \Delta_B(t)$, where $\mathbf{B}_0 \in \mathbb{R}^{n \times Nq}$ is a precisely known matrix and $\Delta_B(t) : \mathbb{R}^+ \mapsto \mathbb{R}^{n \times Nq}$ an unknown, (possibly) function of time, matrix such that $\forall t \geqslant 0$, $\Delta_B'(t)\Delta_B(t) \leqslant \varepsilon^2 \mathbf{I}$, with $\varepsilon > 0$. Similar analysis as in the sequel can be done considering uncertainties in other matrices that describe the network or the algorithm.

For an uncertain network described in this way, directly from Lemma 5.5, a sufficient condition for stability of the error is the existence of matrix $\mathbf{Q} = \mathbf{Q}' > 0$ such that:

$$\mathbf{TCB}_0\mathbf{HQ} + \mathbf{QH}'\mathbf{B}_0'\mathbf{C}'\mathbf{T}' + \mathbf{TC}\Delta_B(t)\mathbf{HQ} + \mathbf{QH}'\Delta_B'(t)\mathbf{C}'\mathbf{T}' < 0 \tag{5.20}$$

This condition cannot be numerically verified as it includes matrix $\Delta_B(t)$ which is unknown. Using the crossed products property (see Proposition A.15 in the Appendix), to impose nega-

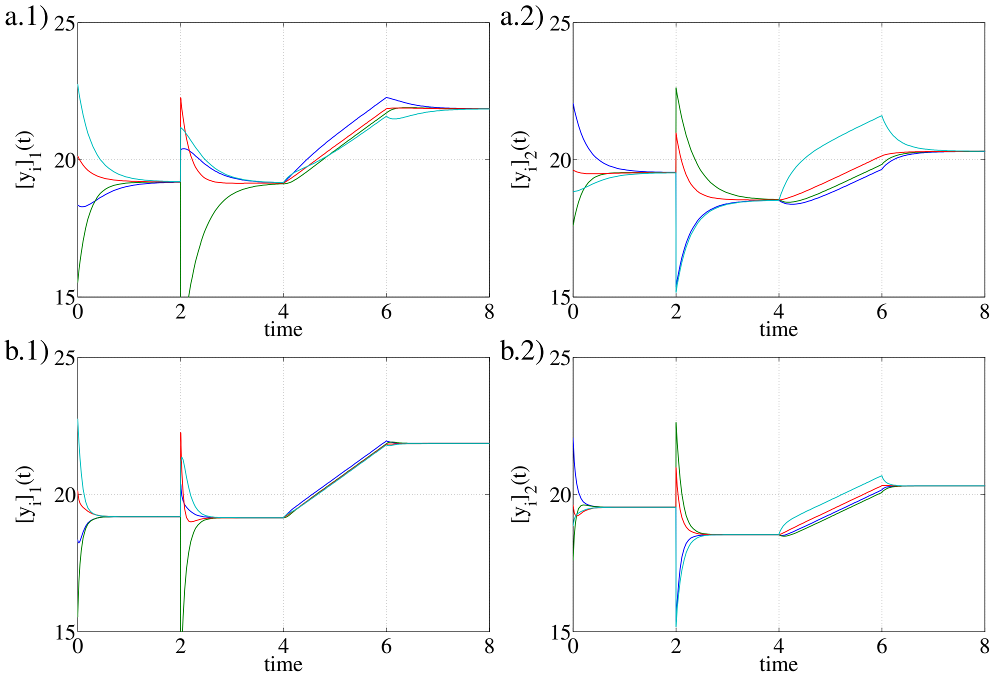


Figure 5.3.: First and second output evolution for integrators agents in Example 5.1 with a) Algorithm $\mathbf{L}_1$ and b) Algorithm $\mathbf{L}_2$.

tivity of an upper bound of the matrix in (5.20) is a sufficient condition for stability, *i.e.*

$$\begin{aligned}\mathbf{G}_0\mathbf{Q}+\mathbf{Q}\mathbf{G}_0'+\mathbf{T}\mathbf{C}\Delta_B(t)\mathbf{H}\mathbf{Q}+\mathbf{Q}\mathbf{H}'\Delta_B'(t)\mathbf{C}'\mathbf{T}' &\leqslant \mathbf{G}_0\mathbf{Q}+\mathbf{Q}\mathbf{G}_0'+\alpha\mathbf{T}\mathbf{C}\mathbf{C}'\mathbf{T}'+\frac{1}{\alpha}\mathbf{Q}\mathbf{H}'\Delta_B'(t)\Delta_B(t)\mathbf{H}\mathbf{Q}\\ &\leqslant \mathbf{G}_0\mathbf{Q}+\mathbf{Q}\mathbf{G}_0'+\alpha\mathbf{T}\mathbf{C}\mathbf{C}'\mathbf{T}'+\frac{\varepsilon^2}{\alpha}\mathbf{Q}\mathbf{H}'\mathbf{H}\mathbf{Q}\\ &\overset{!}{<} 0\end{aligned}$$

with $\mathbf{G}_0 = \mathbf{T}\mathbf{C}\mathbf{B}_0\mathbf{H}$ and scalar $\alpha > 0$. Applying Schur's complement, the following sufficient condition for stability is obtained:

$$\begin{bmatrix}\mathbf{T}\mathbf{C}\mathbf{B}_0\mathbf{H}\mathbf{Q}+\mathbf{Q}\mathbf{H}'\mathbf{B}_0'\mathbf{C}'\mathbf{T}'+\alpha\mathbf{T}\mathbf{C}\mathbf{C}'\mathbf{T}' & \mathbf{Q}\mathbf{H}'\\ \star & -\frac{\alpha}{\varepsilon^2}\mathbf{I}\end{bmatrix}<0 \tag{5.21}$$

This condition depends only on the quadratic bound of the uncertainty, not on the uncertain matrix itself, what makes it possible to evaluate it numerically. Note that a small ε^2 makes it more likely to find a suitable $\alpha > 0$ such that the $(1,1)$-block term of inequality (5.21) is negative definite. That is, it is numerically less demanding to prove stability of a system with "small" uncertainties than other with "large" uncertainties. A stability condition for the system under these circumstances can be formalized by the following theorem.

Theorem 5.11. *A given IN with parametric uncertainties described as* $\mathbf{B} = \mathbf{B}_0 + \Delta_B(t)$*, such that* $\mathbf{B}_0 \in \mathbb{R}^{n\times Nq}$ *is a precisely known matrix and* $\Delta_B(t) : \mathbb{R}^+ \mapsto \mathbb{R}^{n\times Nq}$ *an unknown, (possibly) function of time, matrix where* $\forall t \geqslant 0$*,* $\Delta_B'(t)\Delta_B(t) \leqslant \varepsilon^2\mathbf{I}$ *and* $\varepsilon > 0$*; and with a Laplacian algorithm* $\mathbf{L} = -\hat{L}(\mathscr{G}_w)$*, reaches consensus if for a given organization matrix* $\mathbf{T} = D(\mathscr{T}^o)$*, there exists* $\mathbf{Q} = \mathbf{Q}' > 0$ *of proper dimensions and a scalar* $\alpha > 0$ *such that LMI (5.21) holds.*

Note that there is no structure assumption about the uncertain matrix $\Delta_B(t)$. That means that Theorem 5.11 proves stability in the overly restrictive case where structural zero entries of matrix $\mathbf{B}$ (entries not in the main block diagonal) are possibly uncertain. Furthermore, with the previous formulation it is considered that the uncertainties of every system in the network are bounded by the same value. This is of course not necessarily the case in all networks. To keep this section brief, no further developments to solve these problems are explicitly considered. However, it is possible to study them by considering the uncertainties to be in the form of $\Delta_B(t) = \sum_{i=1}^{N}\mathbf{s}_i\Delta_i(t)\mathbf{s}_i'$ where $\forall i \in \mathscr{V}$, $\Delta_i'(t)\Delta_i(t) \leqslant \varepsilon_i^2\mathbf{I}$ and $\mathbf{s}_i$ are column vectors of matrices where each block is zero except for the i-th block which is the identity matrix. That results in a block diagonal matrix $\Delta_B(t)$ where each block has its own quadratic bound.

Example 5.2. Consider the same network, organization and algorithms as in Example 5.1 but with uncertainty in the input matrix. That is, $\mathbf{B} = \operatorname{diag}\{\mathbf{B}_i\}_{i\in\mathscr{V}} + \Delta_B(t)$. With the value $\varepsilon = 0.6$

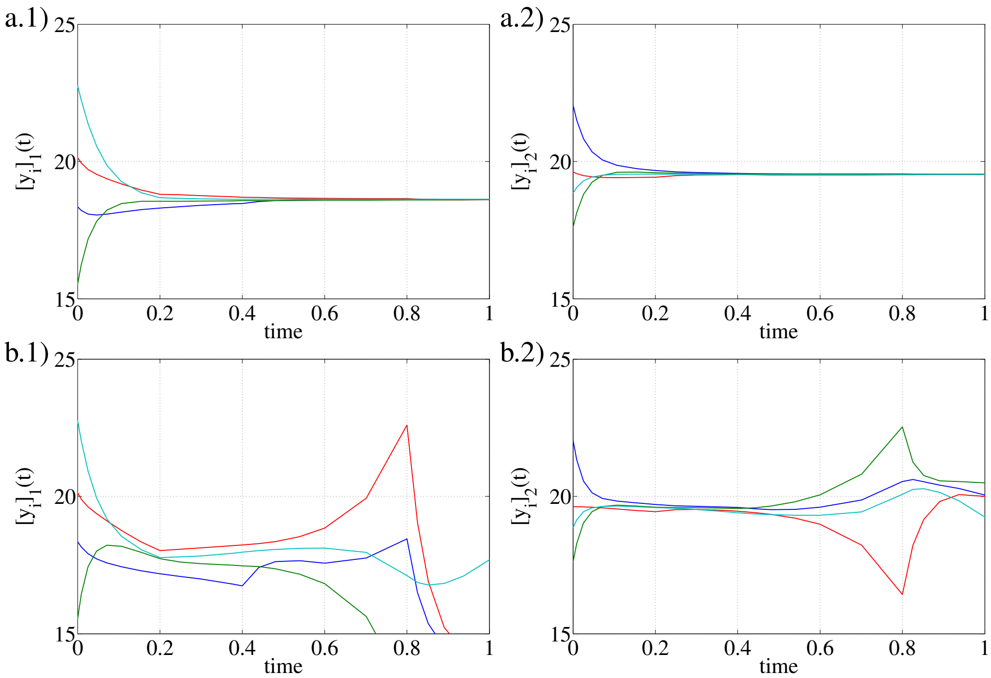


Figure 5.4.: First and second output evolution for integrators agents in Example 5.2 with Algorithm $\mathbf{L}_2$ under uncertainties with a) $\varepsilon = 0.6$ and b) $\varepsilon = 1.2$.

for the designed algorithm $\mathbf{L}_2$, Theorem 5.11 with organization $\mathbf{T}_a$ leads to $\alpha = 0.5457$ and

$$\mathbf{Q} = \begin{bmatrix} 0.1095 & -0.0283 & 0.0457 & -0.0195 & 0.0457 & -0.0195 \\ -0.0283 & 0.1000 & -0.0192 & 0.0426 & -0.0192 & 0.0426 \\ 0.0457 & -0.0192 & 0.0885 & -0.0218 & 0.0134 & -0.0138 \\ -0.0195 & 0.0426 & -0.0218 & 0.0802 & -0.0138 & 0.0135 \\ 0.0457 & -0.0192 & 0.0134 & -0.0138 & 0.0885 & -0.0218 \\ -0.0195 & 0.0426 & -0.0138 & 0.0135 & -0.0218 & 0.0802 \end{bmatrix} > 0$$

which proves that the network reaches consensus for this level of uncertainty. However, if $\varepsilon = 1.2$ then it is not possible to solve the corresponding feasibility problem. As Theorem 5.11 involves only a sufficient condition for stability there is no formal guarantee that the network does not reach consensus under this condition.

To verify this, the network is simulated with a randomly generated time varying uncertainty over the diagonal elements of $\mathbf{B}$ with $\varepsilon = 0.6$ in Figure 5.4 a). The network is then again simulated under identical conditions but with the uncertain matrix multiplied by two at every instant. The results are shown in Figure 5.4 b). It is clear that in the first case the network reaches consensus but not when the uncertainty is larger. ■

5.1.2. Autonomous Agents Network

Consider the more general case where $\mathbf{A} \neq \mathbf{0}$. Here consensus does not only depend on the properties of the algorithm, but on the relationship between the states of the systems and the consensus errors as defined in equation (5.4):

$$\dot{\mathbf{e}} = \mathbf{TCBLT}^{+}\mathbf{e} + \mathbf{TCAx}.$$

However, the inverse relationship between the consensus error and the states of the systems is not explicitly considered. That is, the effect of $\mathbf{x}$ over $\mathbf{e}$ is not adequately expressed. Therefore, the condition to guarantee consensus that can be derived from this expression (namely that $\mathbf{G} = \mathbf{TCBLT}^{+}$ must be Hurwitz and $\mathbf{A} = \mathbf{0}$), is overly restrictive.

Pseudoinverse of the Output Matrix C

Under the assumption that all matrices $\mathbf{C}_i \in \mathbb{R}^{q \times n_i}$, with $q < n_i$, of an AAN are full row-rank, then the right pseudoinverse of $\mathbf{C}$ can be calculated as

$$\mathbf{C}^{+} = \mathbf{C}'(\mathbf{CC}')^{-1} \in \mathbb{R}^{n \times Nq},$$

so that $\mathbf{CC}^{+} = \mathbf{I}$, where $n > Nq$. Let matrix $\mathbf{C}^{\perp} \in \mathbb{R}^{n \times (n-Nq)}$ be an orthonormal basis for the null space of $\mathbf{C}$, that is, $\mathbf{CC}^{\perp} = \mathbf{0}$ and $(\mathbf{C}^{\perp})'\mathbf{C}^{\perp} = \mathbf{I}$.

Similar as for $\mathbf{T}$ in section 4.2.2, the following can be written as the composed matrix $\begin{bmatrix} \mathbf{C}' & \mathbf{C}^{\perp} \end{bmatrix}$ is non singular:

$$\begin{aligned}
&\begin{bmatrix} \mathbf{C}' & \mathbf{C}^{\perp} \end{bmatrix} \begin{bmatrix} \mathbf{C}' & \mathbf{C}^{\perp} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{C} \\ (\mathbf{C}^{\perp})' \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{C} \\ (\mathbf{C}^{\perp})' \end{bmatrix} = \mathbf{I} \\
\Longleftrightarrow\ &\begin{bmatrix} \mathbf{C}' & \mathbf{C}^{\perp} \end{bmatrix} \begin{bmatrix} \mathbf{CC}' & \mathbf{CC}^{\perp} \\ (\mathbf{C}^{\perp})'\mathbf{C}' & (\mathbf{C}^{\perp})'\mathbf{C}^{\perp} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{C} \\ (\mathbf{C}^{\perp})' \end{bmatrix} = \mathbf{I} \\
\Longleftrightarrow\ &\mathbf{C}'(\mathbf{CC}')^{-1}\mathbf{C} + \mathbf{C}^{\perp}((\mathbf{C}^{\perp})'\mathbf{C}^{\perp})^{-1}(\mathbf{C}^{\perp})' = \mathbf{I}.
\end{aligned}$$

Replacing the definition of the pseudo inverse and the null space basis,

$$\mathbf{C}^{+}\mathbf{C} = \mathbf{I} - \mathbf{C}^{\perp}(\mathbf{C}^{\perp})'.$$

From here, an inverse relationship for $\mathbf{x}$ in terms of $\mathbf{y}$ can be developed:

$$\begin{aligned}
&\mathbf{y} = \mathbf{Cx} \\
\Longleftrightarrow\ &\mathbf{C}^{+}\mathbf{y} = \mathbf{C}^{+}\mathbf{Cx} = \mathbf{x} - \mathbf{C}^{\perp}(\mathbf{C}^{\perp})'\mathbf{x} \\
\Longleftrightarrow\ &\mathbf{x} = \mathbf{C}^{+}\mathbf{y} + \mathbf{C}^{\perp}(\mathbf{C}^{\perp})'\mathbf{x}.
\end{aligned}$$

Then, using equation (4.7), $\mathbf{y} = \mathbf{T}^{+}\mathbf{e} + \mathbf{JCx}$, we finally obtain that:

$$\mathbf{x} = \mathbf{C}^{+}\mathbf{T}^{+}\mathbf{e} + \left(\mathbf{C}^{+}\mathbf{JC} + \mathbf{C}^{\perp}(\mathbf{C}^{\perp})'\right)\mathbf{x}. \tag{5.22}$$

When $\mathbf{C} \in \mathbb{R}^{Nq \times Nq}$ is square and non singular, the previous relationships are simplified as $\mathbf{C}^{+} = \mathbf{C}^{-1}$ and $\mathbf{C}^{\perp}$ cannot be defined[2]. In that case $\mathbf{x} = \mathbf{C}^{-1}\mathbf{T}^{+}\mathbf{e} + \mathbf{C}^{-1}\mathbf{JCx}$.

States Influence

Replacing equation (5.22) in the dynamics of the error (5.4), the following is obtained:

$$\dot{\mathbf{e}} = \mathbf{TC}\left(\mathbf{BL} + \mathbf{AC}^{+}\right)\mathbf{T}^{+}\mathbf{e} + \mathbf{TCA}\left(\mathbf{C}^{+}\mathbf{JC} + \mathbf{C}^{\perp}(\mathbf{C}^{\perp})'\right)\mathbf{x} \tag{5.23}$$

Note that consensus is not only dependent on the chosen algorithm but also on internal characteristics of the network, namely, on the matrices $\mathbf{A}$ and $\mathbf{C}$. Because of this, from expression (5.23) we can state a less restrictive result on consensus.

Lemma 5.12. *In an AAN, if* $\mathbf{R}_A := \mathbf{TCA}\left(\mathbf{C}^{+}\mathbf{JC} + \mathbf{C}^{\perp}(\mathbf{C}^{\perp})'\right)$ *is identically zero, and if* $\mathbf{L} = -\hat{L}(\mathscr{G}_w)$ *is selected so that* $\mathbf{G}_A := \mathbf{TC}\left(\mathbf{BL} + \mathbf{AC}^{+}\right)\mathbf{T}^{+}$ *is Hurwitz, then the network reaches consensus. Furthermore, if* $\mathbf{CA} \neq \mathbf{0}$*, then consensus is not static.*

Proof. Directly from equation (5.23) is easy to verify that if $\mathbf{R}_A = \mathbf{0}$, then the error convergences to the origin if $\mathbf{G}_A$ is Hurwitz. To prove that it is non statical consensus, consider that if $\mathbf{e} = \mathbf{0}$, then $\mathbf{y} = \mathbf{1v}(t)$. It is then sufficient to show that $\mathbf{v}(t)$ is indeed a function of time. As $\dot{\mathbf{y}} = \mathbf{CAx} + \mathbf{CBLy}$, evaluating when the error vanishes, $\mathbf{1}\dot{\mathbf{v}}(t) = \mathbf{CAx} + \mathbf{CBL1v}(t) = \mathbf{CAx}$ thus $\dot{\mathbf{v}} = \frac{1}{N}\mathbf{1}'\mathbf{CAx}$. Then $\mathbf{v} : \mathbb{R}_0^{+} \mapsto \mathbb{R}^{q}$ is a function of time which is not constant at all times unless $\mathbf{1}'\mathbf{CA} = \mathbf{0} \iff \mathbf{CA} = \mathbf{0}$ (because of the block diagonal structure of $\mathbf{CA}$). □

Note that when $\mathbf{R}_A = \mathbf{0}$, similar properties as for integrator networks hold. For example, if $\mathbf{G}_A$ is Hurwitz, then it is regardless of the organization used to define it: $Z \cup \operatorname{eig}\{\mathbf{G}_A\} = Z \cup \operatorname{eig}\left\{\mathbf{TC}\left(\mathbf{BL} + \mathbf{AC}^{+}\right)\mathbf{T}^{+}\right\} = \operatorname{eig}\left\{\mathbf{CBL} + \mathbf{CAC}^{+}(\mathbf{I} - \mathbf{J})\right\}$ and so the non zero eigenvalues of $\mathbf{G}_A$ are the non zero eigenvalues of $\mathbf{CBL} + \mathbf{CAC}^{+}(\mathbf{I} - \mathbf{J})$. This also shows that consensus also depends on the dynamics of the agents through matrices $\mathbf{A}$ and $\mathbf{C}$, and not only on the chosen algorithm.

Unfortunately, it is difficult to characterize the eigenvalues of the sum of matrices as a function of the eigenvalues of the individual matrices, so it is not easy to find graph theoretical properties on $\mathscr{G}_w$ that ensure that consensus can be achieved. Nevertheless, Lyapunov methods remain valid to study consensus. Therefore, the convergence rate of the error can still be characterized either through the eigenvalues of $\mathbf{G}_A$ or a scalar $\varsigma > 0$ such that (5.10) holds. Also, sensibility against external signals or robustness against parametric uncertainties can be studied. From here, analogous to those of INs, design procedures of consensus algorithms that consider certain performance criteria can be developed.

[2]The null space of $\mathbf{C}$ in that case is $\operatorname{null}\{\mathbf{C}\} = \{\mathbf{0}\}$ and its basis, $\mathbf{C}^{\perp} = \mathbf{0}$, is not orthonormal.

Augmented Error Dynamics

To address the more challenging case when $\mathbf{R}_A \neq \mathbf{0}$, the dynamics of the error can be expressed in terms of an augmented states space. Define the following additional signals:

$$\mathbf{y}_\perp = (\mathbf{C}^\perp)'\mathbf{x},$$
$$\mathbf{v} = \frac{1}{N}\mathbf{1}'\mathbf{C}\mathbf{x}.$$

Signal $\mathbf{y}_\perp \in \mathbb{R}^{n-Nq}$ is a complementary output that maps all the information of the states that is not mapped into the output vector $\mathbf{y} \in \mathbb{R}^{Nq}$, while signal $\mathbf{v} \in \mathbb{R}^q$ corresponds to the mean value of the outputs. The composed matrix transformation

$$\mathbf{R} := \begin{bmatrix} \mathbf{TC} \\ (\mathbf{C}^\perp)' \\ \frac{1}{N}\mathbf{1}'\mathbf{C} \end{bmatrix} \in \mathbb{R}^{n\times n}$$

is invertible with

$$\mathbf{R}^{-1} = \begin{bmatrix} \mathbf{C}^+\mathbf{T}^+ & \mathbf{C}^\perp & \mathbf{C}^+\mathbf{1} \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

This can be easily checked by performing the multiplications $\mathbf{R}\mathbf{R}^{-1} = \mathbf{R}^{-1}\mathbf{R} = \mathbf{I}$. Therefore we have that

$$\begin{bmatrix} \mathbf{e} \\ \mathbf{y}_\perp \\ \mathbf{v} \end{bmatrix} = \mathbf{R}\mathbf{x} \iff \mathbf{x} = \mathbf{R}^{-1}\begin{bmatrix} \mathbf{e} \\ \mathbf{y}_\perp \\ \mathbf{v} \end{bmatrix}.$$

Because the dynamics of the states in closed loop are given by

$$\dot{\mathbf{x}} = (\mathbf{A} + \mathbf{BLC})\,\mathbf{x}, \tag{5.24}$$

the dynamics of the error can be expressed by the augmented system:

$$\begin{bmatrix} \dot{\mathbf{e}} \\ \dot{\mathbf{y}}_\perp \\ \dot{\mathbf{v}} \end{bmatrix} = \mathbf{R}\,(\mathbf{A} + \mathbf{BLC})\,\mathbf{R}^{-1}\begin{bmatrix} \mathbf{e} \\ \mathbf{y}_\perp \\ \mathbf{v} \end{bmatrix}. \tag{5.25}$$

Equation (5.25) is simply a coordinates transformation of the states space of (5.24) in terms of the error and the additional signals $\mathbf{v}$ and $\mathbf{y}_\perp$.

Considering that $\mathbf{L1} = \mathbf{0}$, developing the matrix in (5.25) leads to

$$\mathbf{G}_x := \mathbf{R}\,(\mathbf{A} + \mathbf{BLC})\,\mathbf{R}^{-1} = \begin{bmatrix} \mathbf{TC}(\mathbf{BL} + \mathbf{AC}^+)\mathbf{T}^+ & \mathbf{TCAC}^\perp & \mathbf{TCAC}^+\mathbf{1} \\ (\mathbf{C}^\perp)'(\mathbf{BL} + \mathbf{AC}^+)\mathbf{T}^+ & (\mathbf{C}^\perp)'\mathbf{AC}^\perp & (\mathbf{C}^\perp)'\mathbf{AC}^+\mathbf{1} \\ \frac{1}{N}\mathbf{1}'\mathbf{C}(\mathbf{BL} + \mathbf{AC}^+)\mathbf{T}^+ & \frac{1}{N}\mathbf{1}'\mathbf{CAC}^\perp & \frac{1}{N}\mathbf{1}'\mathbf{CAC}^+\mathbf{1} \end{bmatrix}. \tag{5.26}$$

From this matrix is easy to obtain sufficient conditions for consensus. For example, if $\mathbf{G}_x \in \mathbb{R}^{n\times n}$ is Hurwitz, then consensus is trivially reached. That would impose that not only the error $\mathbf{e}$ approaches the origin, but also signals $\mathbf{y}_\perp$ and $\mathbf{v}$ (and therefore also $\mathbf{x}$). This is of course restrictive when we are only interested on the consensus error. The desired case is that of the dynamics of the error totally decoupled from the rest of the variables. That is, when the block matrices in positions $(1,2)$ and $(1,3)$ of $\mathbf{G}_x$ are identically zero. In that case, consensus is reached if matrix $\mathbf{G}_A = \mathbf{TC}(\mathbf{BL}+\mathbf{AC}^+)\mathbf{T}^+$ is Hurwitz.

Is not difficult to show that the condition that the dynamics of the error are decoupled from the rest of the variables, *i.e.* $\begin{bmatrix}\mathbf{TCAC}^\perp & \mathbf{TCAC}^+\mathbf{1}\end{bmatrix} = \mathbf{0}$, is equivalent to the condition stated in Lemma 5.12, *i.e.* $\mathbf{R}_A = \mathbf{0}$. Indeed, if we assume that the dynamics of the error are decoupled we can show that the condition of the lemma holds:

$$\begin{aligned}\begin{bmatrix}\mathbf{TCAC}^\perp & \mathbf{TCAC}^+\mathbf{1}\end{bmatrix} = \mathbf{0} \Rightarrow \begin{bmatrix}\mathbf{TCAC}^\perp & \mathbf{TCAC}^+\mathbf{1}\end{bmatrix}\begin{bmatrix}(\mathbf{C}^\perp)' \\ \frac{1}{N}\mathbf{1}'\mathbf{C}\end{bmatrix} = \mathbf{0} \\ \Rightarrow \mathbf{R}_A = \mathbf{TCA}\left(\mathbf{C}^+\mathbf{JC} + \mathbf{C}^\perp(\mathbf{C}^\perp)'\right) = \mathbf{0}.\end{aligned}$$

On the other side, if we assume that the condition in Lemma 5.12 holds, we obtain that the dynamics are decoupled:

$$\begin{aligned}\mathbf{R}_A = \mathbf{TCA}\left(\mathbf{C}^+\mathbf{JC} + \mathbf{C}^\perp(\mathbf{C}^\perp)'\right) = \mathbf{0} \Rightarrow \mathbf{TCA}\left(\mathbf{C}^+\mathbf{JC} + \mathbf{C}^\perp(\mathbf{C}^\perp)'\right)\begin{bmatrix}\mathbf{C}^\perp & \mathbf{C}^+\mathbf{1}\end{bmatrix} = \mathbf{0} \\ \Rightarrow \begin{bmatrix}\mathbf{TCAC}^\perp & \mathbf{TCAC}^+\mathbf{1}\end{bmatrix} = \mathbf{0}.\end{aligned}$$

Between these two extreme cases, several partially decoupled cases can be defined. For example, if $\mathbf{TCAC}^+\mathbf{1} = \mathbf{0}$ and $(\mathbf{C}^\perp)'\mathbf{AC}^+\mathbf{1} = \mathbf{0}$, then the dynamics of the error are decoupled from the mean value of the outputs $\mathbf{v}$ but not from the complementary output $\mathbf{y}_\perp$. In that case, if the matrix

$$\begin{bmatrix}\mathbf{I}_{n-q} & \mathbf{0}\end{bmatrix}\mathbf{G}_x\begin{bmatrix}\mathbf{I}_{n-q} \\ \mathbf{0}\end{bmatrix} = \begin{bmatrix}\mathbf{TC}(\mathbf{BL}+\mathbf{AC}^+)\mathbf{T}^+ & \mathbf{TCAC}^\perp \\ (\mathbf{C}^\perp)'(\mathbf{BL}+\mathbf{AC}^+)\mathbf{T}^+ & (\mathbf{C}^\perp)'\mathbf{AC}^\perp\end{bmatrix},$$

is Hurwitz, then the network reaches consensus and the complementary output $\mathbf{y}_\perp$ approaches the origin. Note that in this case, it is sufficient for decoupling that $\mathbf{AC}^+\mathbf{1} = \mathbf{0}$. This condition is, however, in general not necessary:

$$\begin{aligned}\begin{bmatrix}\mathbf{TCAC}^+\mathbf{1} \\ (\mathbf{C}^\perp)'\mathbf{AC}^+\mathbf{1}\end{bmatrix} = \mathbf{0} \Rightarrow \begin{bmatrix}\mathbf{C}^+\mathbf{T}^+ & \mathbf{C}^\perp\end{bmatrix}\begin{bmatrix}\mathbf{TC} \\ (\mathbf{C}^\perp)'\end{bmatrix}\mathbf{AC}^+\mathbf{1} = \mathbf{0} \\ \Rightarrow \left(\mathbf{C}^+\mathbf{T}^+\mathbf{TC} + \mathbf{C}^\perp(\mathbf{C}^\perp)'\right)\mathbf{AC}^+\mathbf{1} = \mathbf{0} \\ \Rightarrow \left(\mathbf{I} - \mathbf{C}^+\mathbf{JC}\right)\mathbf{AC}^+\mathbf{1} = \mathbf{0} \\ \Rightarrow \mathbf{AC}^+\mathbf{1} = \mathbf{C}^+\mathbf{JCAC}^+\mathbf{1} \neq \mathbf{0}.\end{aligned}$$

A similar exercise can be done by decoupling the error from the complementary outputs but not from the mean value. In this case we need to introduce a permutation transformation in order to select the rows and columns associated to vector $\mathbf{v}$. Then, if $\mathbf{TCAC}^{\perp} = \mathbf{0}$, $(\mathbf{C}^{\perp})'\mathbf{AC}^{+}\mathbf{1} = \mathbf{0}$, and

$$\begin{bmatrix}\mathbf{I}_{Nq} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{I}_{(N-1)q} & \mathbf{0}\\ \mathbf{0} & \mathbf{U}\end{bmatrix}\mathbf{G}_x\begin{bmatrix}\mathbf{I}_{(N-1)q} & \mathbf{0}\\ \mathbf{0} & \mathbf{U}^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{I}_{Nq}\\ \mathbf{0}\end{bmatrix} = \begin{bmatrix}\mathbf{TC}(\mathbf{BL}+\mathbf{AC}^{+})\mathbf{T}^{+} & \mathbf{TCAC}^{+}\mathbf{1}\\ \frac{1}{N}\mathbf{1}'\mathbf{C}(\mathbf{BL}+\mathbf{AC}^{+})\mathbf{T}^{+} & \frac{1}{N}\mathbf{1}'\mathbf{CAC}^{+}\mathbf{1}\end{bmatrix}$$

is Hurwitz, with

$$\mathbf{U} := \begin{bmatrix}\mathbf{0} & \mathbf{I}_q\\ \mathbf{I}_{n-Nq} & \mathbf{0}\end{bmatrix} \in \mathbb{R}^{(n-(N-1)q)\times(n-(N-1)q)},$$

consensus is reached and the mean value of the outputs approaches the origin. We can generalize this procedure to verify if the consensus error and $r \in \{0,1,2,\dots,n-(N-1)q\}$ additional arbitrary signals approach the origin with a given matrix $\mathbf{L}$ as follows.

Theorem 5.13. *In an AAN with a consensus algorithm described by* $\mathbf{L} = -\hat{L}(\mathscr{G}_w)$*, let* $r \in \{0,1,2,\dots,n-(N-1)q\}$ *be the minimal value for which it exists an invertible matrix* $\mathbf{U} \in \mathbb{R}^{(n-(N-1)q)\times(n-(N-1)q)}$ *such that*

$$\begin{bmatrix}\mathbf{I}_{r+(N-1)q} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{I}_{(N-1)q} & \mathbf{0}\\ \mathbf{0} & \mathbf{U}\end{bmatrix}\mathbf{G}_x\begin{bmatrix}\mathbf{I}_{(N-1)q} & \mathbf{0}\\ \mathbf{0} & \mathbf{U}^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{0}\\ \mathbf{I}_{n-r-(N-1)q}\end{bmatrix} = \mathbf{0}. \tag{5.27}$$

In that case, if

$$\mathbf{G}_r := \begin{bmatrix}\mathbf{I}_{r+(N-1)q} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{I}_{(N-1)q} & \mathbf{0}\\ \mathbf{0} & \mathbf{U}\end{bmatrix}\mathbf{G}_x\begin{bmatrix}\mathbf{I}_{(N-1)q} & \mathbf{0}\\ \mathbf{0} & \mathbf{U}^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{I}_{r+(N-1)q}\\ \mathbf{0}\end{bmatrix}$$

is Hurwitz, then the network reaches consensus and, additionally, r elements of $\mathbf{U}\,[\mathbf{y}'_{\perp}\ \ \mathbf{v}']'$ *approach the origin.*

Proof. For some invertible matrix $\mathbf{U} \in \mathbb{R}^{(n-(N-1)q)\times(n-(N-1)q)}$, the vector $\tilde{\mathbf{e}} = [\mathbf{I}_r\ \ \mathbf{0}]\,\mathbf{U}\,[\mathbf{y}'_{\perp}\ \ \mathbf{v}']' \in \mathbb{R}^r$ corresponds to r linear combinations of the elements of $[\mathbf{y}'_{\perp}\ \ \mathbf{v}']'$. If (5.27) holds, the dynamics of the $\mathbf{e}$ and $\tilde{\mathbf{e}}$ are decoupled from the elements of $\begin{bmatrix}\mathbf{0} & \mathbf{I}_{n-(N-1)q-r}\end{bmatrix}\mathbf{U}\,[\mathbf{y}'_{\perp}\ \ \mathbf{v}']' \in \mathbb{R}^{n-(N-1)q-r}$ and given by $\begin{bmatrix}\dot{\mathbf{e}}' & \dot{\tilde{\mathbf{e}}}'\end{bmatrix}' = \mathbf{G}_r\,[\mathbf{e}'\ \ \tilde{\mathbf{e}}']'$, which is clearly stable if $\mathbf{G}_r$ is Hurwitz. □

For a given network and algorithm, Theorem 5.13 gives a way to check if consensus can be reached when the dynamics of the error depend on the internal dynamics of the agents. Note that if $r = 0$, then the error dynamics are decoupled from other signals and we obtain an equivalent result as that of Lemma 5.12. In the case where $r = n-(N-1)q$, only trivial consensus can be reached as then $\mathbf{G}_r = \mathbf{G}_x$.

The following convergence rate indicators can be defined,

$$\lambda_r := \min\{\text{real}\{\text{eig}\{-\mathbf{G}_r\}\}\}$$
$$\lambda_x := \min\{\text{real}\{\text{eig}\{-\mathbf{G}_x\}\}\}.$$

Clearly consensus is reached if $\lambda_r > 0$. Furthermore, if $\lambda_x > 0$, the system (5.24) is asymptotically stable. However, if $\lambda_r \gg \lambda_x > 0$, consensus is reached faster than the convergence rate of the states of the system, leading to a case of bounded dynamical consensus, and not merely trivial consensus. On the contrary, if $0 < \lambda_r < \lambda_x$, the states of the network reach the origin faster, what characterizes trivial consensus.

On the rest of this thesis, we will concentrate on the case where the dynamics of the error are decoupled from any other signal, *i.e.*, when $\mathbf{R}_A = \mathbf{0}$, as in other cases, extensions of the following discussions can be easily stated.

Example 5.3. Consider $N = 10$ SISO agents (*i.e.* with $q = p_i = 1$) and the following state matrices:

$$\bar{\mathbf{A}}_a = \begin{bmatrix} -0.1 & 0.0 \\ 0.0 & -0.1 \end{bmatrix}, \bar{\mathbf{A}}_b = \begin{bmatrix} 0.0 & 0.1 \\ 0.1 & 0.0 \end{bmatrix}, \bar{\mathbf{A}}_c = \begin{bmatrix} -0.2 & 0.1 \\ 0.1 & -0.1 \end{bmatrix}, \bar{\mathbf{A}}_d = \begin{bmatrix} -0.8 & 0.1 \\ 0.1 & -0.1 \end{bmatrix},$$
$$\bar{\mathbf{A}}_e = \begin{bmatrix} -0.1 & 0.1 \\ 0.1 & 0.0 \end{bmatrix}, \bar{\mathbf{A}}_f = \begin{bmatrix} -0.1 & 0.1 \\ 0.1 & 0.1 \end{bmatrix}, \bar{\mathbf{A}}_g = \begin{bmatrix} 0.1 & 0.0 \\ 0.0 & -0.1 \end{bmatrix}, \bar{\mathbf{A}}_h = \begin{bmatrix} 0.1 & -0.1 \\ 0.1 & 0.0 \end{bmatrix}.$$

From these matrices, eight different networks are defined, each of them with identical agents with the respective state matrix and considering $\forall i \in \{1,2,\ldots,N\}$,

$$\mathbf{B}_i = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}, \text{ and } \mathbf{C}_i = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}.$$

Note that the eigenvalues of the aggregated matrices $\mathbf{A}_k = \text{diag}\{\bar{\mathbf{A}}_k\}_{i=1}^N, k \in \{a,b,c,d,e,f,g,h\}$, of each network are the two eigenvalues of $\bar{\mathbf{A}}_k$ with multiplicity N each.

For consensus analysis, a star organization centered on the first agent shown in Figure 5.5 a) with its respective matrix $\mathbf{T} = D'(\mathscr{T}^o) = \text{row}\{\mathbf{1}, -\mathbf{I}\}$ will be used. The networks will be studied under a the fully connected algorithm shown in Figure 5.5 b) with its respective matrix $\mathbf{L}_g = -\hat{L}(\mathscr{G}) = N(\mathbf{J} - \mathbf{I}) \in \mathbb{R}^{N\times N}$. Note that for this algorithm and this organization the consensus convergence rate as defined in Section 5.1.1 becomes $\lambda = 10$. Additionally, consider the weighted tree in Figure 5.5 c) to define a second consensus algorithm $\mathbf{L}_t = -\hat{L}(\mathscr{T}_w) = -\mathbf{T}'\mathbf{W}\mathbf{T}$, where $\mathbf{W} = \text{diag}\{\mathbf{W}_{1,i}\}_{i=2}^N$. The weights of this algorithm are chosen as $\mathbf{W}_{1,i} = 10$, $\forall i \in \{2,3,\ldots,N\}$, to also obtain $\lambda = 10$.

In Table 5.2 the characteristics of the previous networks in closed loop considering both algorithms are shown. The first two networks are such that $\mathbf{R}_A = \mathbf{0}$. As additionally $\mathbf{G}_A$ is Hurwitz, consensus can be reached even if the states present unstable dynamics. Networks c)

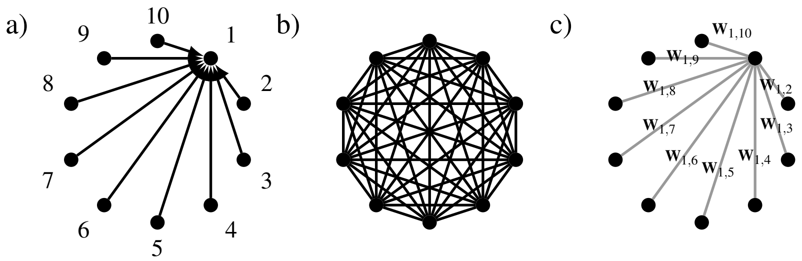


Figure 5.5.: a) Organization Tree $\mathscr{T}^o$, b) Unweighted Graph $\mathscr{G}$ and, c) Weighted Tree $\mathscr{T}_w$ for Example 5.3.

Table 5.2.: Networks characteristics in Example 5.3.

Network	$\text{eig}\left\{\bar{\mathbf{A}}^k\right\}$		$\mathbf{L}_g = -\hat{L}(\mathscr{G})$				$\mathbf{L}_t = -\hat{L}(\mathscr{T}_w)$			
			λ_r	λ_x	$ISD(50)$	$IAD(50)$	λ_r	λ_x	$ISD(50)$	$IAD(50)$
a)	-0.1000	-0.1000	10.1000	0.1000	1.2856	3.4385	10.1000	0.1000	1.2127	3.1467
b)	-0.1000	0.1000	9.9000	-0.1000	1.3106	3.5080	9.9000	-0.1000	1.2371	3.2097
c)	-0.2618	-0.0382	0.2497	0.0382	1.5027	4.0204	0.2497	0.0382	1.4314	3.7370
d)	-0.8140	-0.0860	0.5375	0.0860	1.9256	5.1841	0.5375	0.0860	1.8598	4.9243
e)	-0.1618	0.0618	0.1497	-0.0618	1.6678	4.4590	0.1497	-0.0618	1.5958	4.1710
f)	-0.1414	0.1414	0.0990	-0.1414	2.4285	6.4855	0.0990	-0.1414	2.3570	6.1958
g)	-0.1000	0.1000	-0.0010	-0.1000	6.9045	18.3051	-0.0010	-0.1000	6.8297	17.8930
h)	$0.0500 \pm 0.0866j$		-0.0500	-0.0500	37.8208	100.3292	-0.0500	-0.0500	37.7478	99.6120

and d) are both stable in open and closed loop for both algorithms, therefore they can also reach consensus in the long term. Nevertheless, by inspection of matrix $\mathbf{G}_x$ (which is not shown for space limitations), one can verify that in these cases the dynamics of the error $\mathbf{e}$ and the complementary outputs $\mathbf{y}_\perp$, are both decoupled of the mean value $\mathbf{v}$. By inspecting the values of λ_x and λ_r one conclude that consensus is reached before the states approximate the origin.

In the case of networks e) and f), the closed loop systems are unstable with only one eigenvalue at the right-hand side of the complex plane. However, as for networks c) and d), the dynamics of the error and the complementary outputs are decoupled of the mean value. Because $\lambda_r > 0$, this allows to reach consensus even though the overall dynamics of the closed loop networks are unstable. A different situation is observed in the last two networks. These are unstable under both algorithms. Network g) can be decoupled as in the previous cases, but the resulting $\mathbf{G}_r$ matrix is not Hurwitz. Network h) cannot be decoupled, *i.e.* $r = n - (N-1)q$ or, equivalently, $\mathbf{G}_r = \mathbf{G}_x$, which is also not Hurwitz. This leads to conclude that in these cases, consensus cannot be reached.

To verify the previous statements, we will simulate the closed loop response for identical initial conditions. The evolution of the networks is simulated until $t = 50$. In Figure 5.6, for algorithm $\mathbf{L}_g$ and until $t = 2$, the outputs of the networks are shown for networks a) to f), and the error for the last two cases. Algorithm $\mathbf{L}_t$ presents similar graphical results, and therefore the images are not shown.

Note that networks a) and b) reach dynamic consensus very fast as in these cases $\mathbf{R}_A = \mathbf{0}$ and the respective matrices $\mathbf{G}_A$ are Hurwitz for both algorithms. For the stable networks c) and d), consensus is reached before the outputs approaches the origin. A similar situation is observed for networks e) and f), which reach consensus even though the overall dynamics of the closed loop system are unstable. This is not the case for networks g) and h) where consensus is first approximated but, due to the unbounded behavior of $\mathbf{x}$, the algorithm is not able to maintain this approximation too long. Observe further that the ISD and IAD indicators, also shown in Table 5.2 for the described simulations, are coherent with the previous statements. ■

Identical Agents

The study of consensus can be simplified for the special case where in an AAN *all* agents have the same dynamical behavior. Note that if a block diagonal matrix is such that all diagonal blocks are identical, *i.e.* when $\mathbf{X} = \operatorname{diag}\{\mathbf{Y}\}_{i\in\mathscr{V}} \in \mathbb{R}^{Nq\times Nq}$ with $\mathbf{Y} \in \mathbb{R}^{q\times q}$, then it holds that

$$\mathbf{X1} = \operatorname{diag}\{\mathbf{Y}\}_{i\in\mathscr{V}}\mathbf{1} = \operatorname{col}\{\mathbf{Y}\}_{i\in\mathscr{V}} = \mathbf{1Y} \text{ and}$$
$$\mathbf{1}'\mathbf{X} = \mathbf{1}'\operatorname{diag}\{\mathbf{Y}\}_{i\in\mathscr{V}} = \operatorname{row}\{\mathbf{Y}\}_{i\in\mathscr{V}} = \mathbf{Y1}'.$$

Therefore,

$$\mathbf{XJ} = \mathbf{X}\frac{1}{N}\mathbf{11}' = \frac{1}{N}\mathbf{1Y1}' = \frac{1}{N}\mathbf{11}'\mathbf{X} = \mathbf{JX}.$$

Particularly, consider an AAN where, for every $i \in \mathscr{V}$, $\mathbf{A}_i = \mathbf{A}_0 \in \mathbb{R}^{n_0\times n_0}$ and $\mathbf{C}_i = \mathbf{C}_0 \in \mathbb{R}^{q\times n_0}$ with known matrices $\mathbf{A}_0$ and $\mathbf{C}_0$. Because of the block diagonal structure of matrices $\mathbf{A} = \operatorname{diag}\{\mathbf{A}_i\}_{i\in\mathscr{V}}$ and $\mathbf{C} = \operatorname{diag}\{\mathbf{C}_i\}_{i\in\mathscr{V}}$, the product $\mathbf{CAC}^+ = \operatorname{diag}\left\{\mathbf{C}_i\mathbf{A}_i\mathbf{C}_i^+\right\}_{i\in\mathscr{V}}$ is also block diagonal. From here, the residual matrix in (5.23) can be written as:

$$\begin{aligned}
\mathbf{R}_A &= \mathbf{TCA}\left(\mathbf{C}^+\mathbf{JC} + \mathbf{C}^\perp(\mathbf{C}^\perp)'\right) \\
&= \mathbf{TCAC}^+\mathbf{JC} + \mathbf{TCAC}^\perp(\mathbf{C}^\perp)' \\
&= (\mathbf{TJ})\mathbf{CAC}^+\mathbf{C} + \mathbf{TCAC}^\perp(\mathbf{C}^\perp)' \\
&= \mathbf{TCAC}^\perp(\mathbf{C}^\perp)'
\end{aligned}$$

Therefore, the dynamics of the error are simplified to:

$$\dot{\mathbf{e}} = \mathbf{TC}\left(\mathbf{BL} + \mathbf{AC}^+\right)\mathbf{T}^+\mathbf{e} + \mathbf{TCAC}^\perp(\mathbf{C}^\perp)'\mathbf{x}$$

Consensus can then be reached independently of the dynamics of $\mathbf{x}$ only if the product $\mathbf{CAC}^\perp = \mathbf{0}$ or, equivalently because of the block diagonal structure of the matrices, $\mathbf{C}_0\mathbf{A}_0\mathbf{C}_0^\perp = \mathbf{0} \in \mathbb{R}^{q\times q}$.

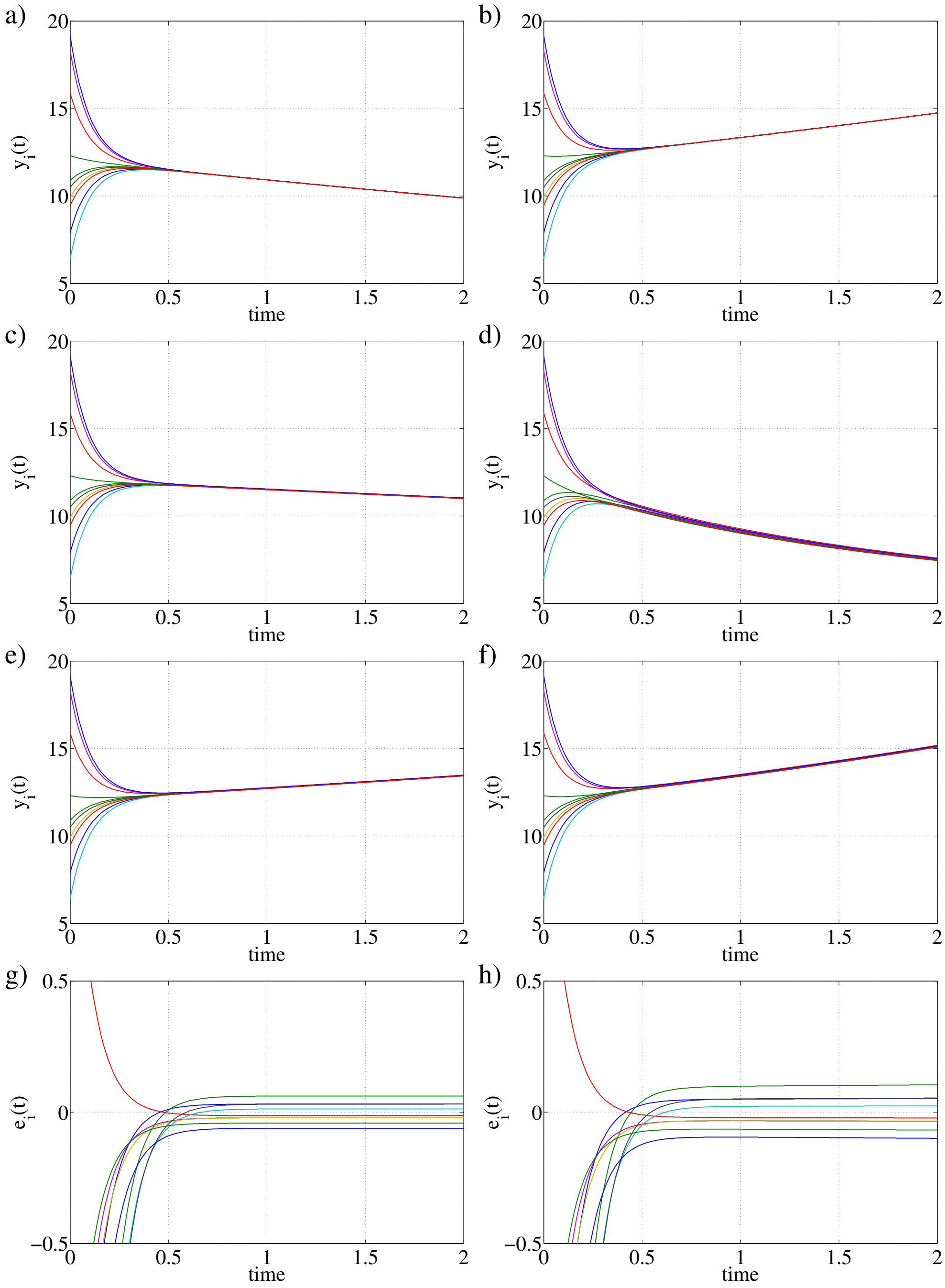


Figure 5.6.: Dynamic evolution for the eight networks analyzed in Example 5.3 with algorithm $\mathbf{L}_g$. a) to f): Outputs $\mathbf{y}$ of the systems, for the respective networks. g) to h): Error $\mathbf{e}$ between the outputs of systems, for the respective networks.

Observe that the requirement that all system and output matrices are identical can be relaxed to only impose that $\forall i \in \mathscr{V}$ the product $\mathbf{C}_i\mathbf{A}_i\mathbf{C}_i^+ \in \mathbb{R}^{q\times q}$ is the same.

A particular case is when all agents are identical and $n_0 = q$. *i.e.* the number of outputs are the same as the number of states of the agents, what implies that $\mathbf{C}_0 \in \mathbb{R}^{q\times q}$ is square. If it is also non singular, the dynamics of the error become:

$$\begin{aligned}
\dot{\mathbf{e}} &= \mathbf{TC}\left(\mathbf{BL}+\mathbf{AC}^+\right)\mathbf{T}^+\mathbf{e}+\mathbf{TCAC}^{-1}\mathbf{JCx}\\
&= \mathbf{TC}\left(\mathbf{BL}+\mathbf{AC}^+\right)\mathbf{T}^+\mathbf{e}+(\mathbf{TJ})\mathbf{CAC}^{-1}\mathbf{Cx}\\
&= \mathbf{TC}\left(\mathbf{BL}+\mathbf{AC}^+\right)\mathbf{T}^+\mathbf{e}
\end{aligned}$$

That is, consensus can be reached independently of $\mathbf{x}$ if $\mathbf{L}$ is chosen correctly.

Sensitivity Against External Signals

An AAN can be under the influence of external signals in the form of control inputs $\mathbf{u}_{ext}: \mathbb{R}_0^+ \mapsto \mathbb{R}^{Nq}$ or perturbations $\mathbf{d}: \mathbb{R}_0^+ \mapsto \mathbb{R}^{Nq}$ as seen in Figure 4.1. Similar as what was done for INs, the sensitivity against this kind of signals can be studied through the H_∞-norms of the respective transfer function matrices.

Considering $\mathbf{y} = \mathbf{Cx}+\mathbf{d} \implies \mathbf{x} = \mathbf{C}^+\mathbf{y}-\mathbf{C}^+\mathbf{d}+\mathbf{C}^\perp(\mathbf{C}^\perp)'\mathbf{x}$ and $\mathbf{u} = \mathbf{Ly}+\mathbf{u}_{ext}$, with $\mathbf{LJ} = \mathbf{0}$, the definition of the error $\mathbf{e} = \mathbf{Ty} \implies \mathbf{y} = \mathbf{T}^+\mathbf{e}+\mathbf{Jy} = \mathbf{T}^+\mathbf{e}+\mathbf{JCx}+\mathbf{Jd}$ leads to the dynamical expression:

$$\begin{aligned}
\dot{\mathbf{e}} &= \mathbf{TC}\dot{\mathbf{x}}+\mathbf{T}\frac{d}{dt}\mathbf{d}\\
&= \mathbf{TCBu}+\mathbf{TCAx}+\mathbf{Tw}\\
&= \mathbf{TCB}\left(\mathbf{L}\left(\mathbf{T}^+\mathbf{e}+\mathbf{JCx}+\mathbf{Jd}\right)+\mathbf{u}_{ext}\right)+\ldots\\
&\qquad \ldots+\mathbf{TCA}\left(\mathbf{C}^+\left(\mathbf{T}^+\mathbf{e}+\mathbf{JCx}+\mathbf{Jd}\right)-\mathbf{C}^+\mathbf{d}+\mathbf{C}^\perp(\mathbf{C}^\perp)'\mathbf{x}\right)+\mathbf{Tw}\\
&= \mathbf{TC}\left(\mathbf{BL}+\mathbf{AC}^+\right)\mathbf{T}^+\mathbf{e}+\mathbf{TCA}\left(\mathbf{C}^+\mathbf{JC}+\mathbf{C}^\perp(\mathbf{C}^\perp)'\right)\mathbf{x}+\mathbf{TCBu}_{ext}+\mathbf{TCAC}^+\left(\mathbf{J}-\mathbf{I}\right)\mathbf{d}+\mathbf{Tw}\\
&= \mathbf{G}_A\mathbf{e}+\mathbf{R}_A\mathbf{x}+\mathbf{TCBu}_{ext}+\mathbf{TCAC}^+\left(\mathbf{J}-\mathbf{I}\right)\mathbf{d}+\mathbf{Tw},
\end{aligned}$$

where $\mathbf{w} = \frac{d}{dt}\mathbf{d}$. Note that, contrary to the case of IN, the dynamics of the error depend on both the value of the perturbation $\mathbf{d}$ and its variation in time $\mathbf{w}$, because $\mathbf{A} \neq \mathbf{0}$.

In the case that $\mathbf{R}_A = \mathbf{0}$, the sensitivity of the consensus error against the external inputs can be studied by defining three transfer function matrices $H_{ext}(s)$, $H_{de}(s)$ and $H_{we}(s)$ between, respectively, $\mathbf{u}_{ext}$, $\mathbf{d}$ and $\mathbf{w}$, and the consensus error $\mathbf{e}$, when the rest of the external signals are zero.

Using the BRL, LMIs (5.28), (5.29) and (5.30), can be stated to respectively study $\|H_{ext}(s)\|_\infty$, $\|H_{de}(s)\|_\infty$ and $\|H_{we}(s)\|_\infty$. As in the previous sections, these inequalities can be used for analysis or design of consensus algorithms that fulfill certain performance criteria. Particularly, if tree shaped algorithms are searched, *i.e.* imposing $\mathbf{L} = -\mathbf{T}'\mathbf{WT}$ with $\mathbf{W}$ block diagonal, the inequalities are sufficient and necessary conditions for design of such algorithms in analogous way as for INs. To maintain this section brief, this will not be explicitly addressed.

$$\begin{bmatrix} \mathbf{PG}_A + \mathbf{G}_A'\mathbf{P} + \mathbf{I} & \mathbf{PTCB} \\ \star & -\gamma_{ext}^2\mathbf{I} \end{bmatrix} < 0 \iff \begin{bmatrix} \mathbf{G}_A\mathbf{Q} + \mathbf{QG}_A' & \mathbf{TCB} & \mathbf{Q} \\ \star & -\gamma_{ext}^2\mathbf{I} & \mathbf{0} \\ \star & \star & -\mathbf{I} \end{bmatrix} < 0 \tag{5.28}$$

$$\begin{aligned} &\begin{bmatrix} \mathbf{PG}_A + \mathbf{G}_A'\mathbf{P} + \mathbf{I} & \mathbf{PTCAC}^{+}(\mathbf{J}-\mathbf{I}) \\ \star & -\gamma_{de}^2\mathbf{I} \end{bmatrix} < 0 \iff \\ &\quad \begin{bmatrix} \mathbf{G}_A\mathbf{Q} + \mathbf{QG}_A' & \mathbf{TCAC}^{+}(\mathbf{J}-\mathbf{I}) & \mathbf{Q} \\ \star & -\gamma_{de}^2\mathbf{I} & \mathbf{0} \\ \star & \star & -\mathbf{I} \end{bmatrix} < 0 \end{aligned} \tag{5.29}$$

$$\begin{bmatrix} \mathbf{PG}_A + \mathbf{G}_A'\mathbf{P} + \mathbf{I} & \mathbf{PT} \\ \star & -\gamma_{we}^2\mathbf{I} \end{bmatrix} < 0 \iff \begin{bmatrix} \mathbf{G}_A\mathbf{Q} + \mathbf{QG}_A' & \mathbf{T} & \mathbf{Q} \\ \star & -\gamma_{we}^2\mathbf{I} & \mathbf{0} \\ \star & \star & -\mathbf{I} \end{bmatrix} < 0 \tag{5.30}$$

With $\mathbf{P} = \mathbf{P}' > 0$ and $\mathbf{Q} = \mathbf{P}^{-1}$ of proper dimensions.

Example 5.4. Consider $N = 15$ identical agents with dynamics described by the following matrices:

$$\mathbf{A}_0 = \begin{bmatrix} -0.5 & 0.1 \\ 0.0 & 0.2 \end{bmatrix}, \mathbf{B}_0 = \begin{bmatrix} 1.5 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}, \text{ and } \mathbf{C}_0 = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}.$$

The agents are submitted to the action of external signals in the form of perturbations $\mathbf{d}_i$. As the agents are identical and the output matrix $\mathbf{C}_0$ is invertible, the residual matrix is identically zero, $\mathbf{R}_A = \mathbf{TCAC}^{-1}\mathbf{JC} = \mathbf{0}$, and the dynamics of the error are described simply by $\dot{\mathbf{e}} = \mathbf{TC}\left(\mathbf{BL} + \mathbf{AC}^{-1}\right)\mathbf{T}^{+}\mathbf{e}$ when the external signals are zero.

The network will be analyzed by an organization derived from $\mathscr{T}^o$ in Figure 5.7 a): $\mathbf{T} = D'(\mathscr{T}^o) = \text{row}\left\{\mathbf{I}_{(N-1)q}, -\mathbf{1}_{(N-1)q\times q}\right\}$. As the network is submitted to the action of external perturbations, LMI (5.29) will be used to design consensus algorithms with the same shape as the organization and diagonal weights. *i.e.* fixing an upper bound $\gamma > 0$ for the H_∞-norm of the system and forcing $\mathbf{L} = -\mathbf{T}'\mathbf{WT}$, with $\mathbf{W}$ diagonal. By making the variable $\mathbf{Q}$ to be diagonal and defining the diagonal auxiliary variable $\mathbf{Z} = \mathbf{WQ} \iff \mathbf{W} = \mathbf{ZQ}^{-1}$, solving the feasibility problem of LMI (5.29) leads to an algorithm in negative Laplacian form based on the shape of the unweighted tree $\mathscr{T}$ in Figure 5.7 b) with a weights function $w_q : \mathscr{E} \mapsto \mathbb{R}^{q\times q}$ such that the H_∞-norm bound holds. Two algorithms, respectively $\mathbf{L}_1 = \hat{L}(\mathscr{T}_{w_1})$ and $\mathbf{L}_2 = \hat{L}(\mathscr{T}_{w_2})$ are designed with this methodology for values $\gamma_1 = 0.10$ and $\gamma_2 = 0.05$.

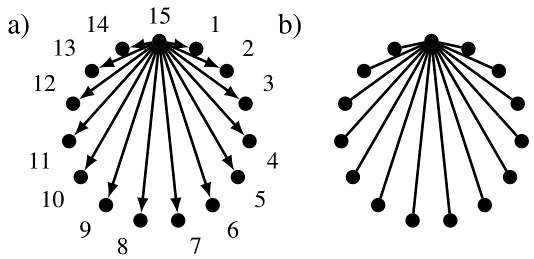


Figure 5.7.: a) Organization Tree $\mathscr{T}^o$, b) Unweighted Tree $\mathscr{T}$ for Example 5.4.

Table 5.3.: Algorithms design and simulation parameters in Example 5.4.

Algorithm	γ_{de}	λ_A	$\Vert H_{de}(s)\Vert_\infty$	$\Vert H_{we}(s)\Vert_\infty$	$\Vert H_{ext}(s)\Vert_\infty$	$ISD(2)$	$IAD(2)$
$\mathbf{L}_1$	0.10	4.570154	0.058780	0.218830	0.218845	3.550483	14.452537
$\mathbf{L}_2$	0.05	7.801018	0.030561	0.128191	0.128192	2.084995	8.243606

Both algorithms are tested through a simulation with identical initial conditions during two seconds. At $t = 1.0$ a randomly generated signal $\mathbf{d}$ is added as a perturbation to the outputs of the network. At $t = 1.5$ the perturbation is modified to a more critical condition (a perturbation with a greater Euclidean norm). The results are to be seen in Table 5.3 where the ISD and IAD indicators are shown along with the design parameter γ_{de}, the module $\lambda_A = \lambda_r$ of the greatest negative eigenvalue of matrix $\mathbf{G}_A$, and the H$_\infty$-norms of the transfer functions from, respectively, $\mathbf{d}$, $\mathbf{w}$, and $\mathbf{u}_{ext}$ to the error $\mathbf{e}$. Additionally, the time responses of the networks are depicted in Figure 5.8. Note that, as $q = 2$, the agents have two outputs with different dynamics that can be clearly distinguished in the figures.

From the table and the figure, it is clear that the second algorithm has a better performance. When the perturbation is not present, it achieves consensus faster and more accurately. Also

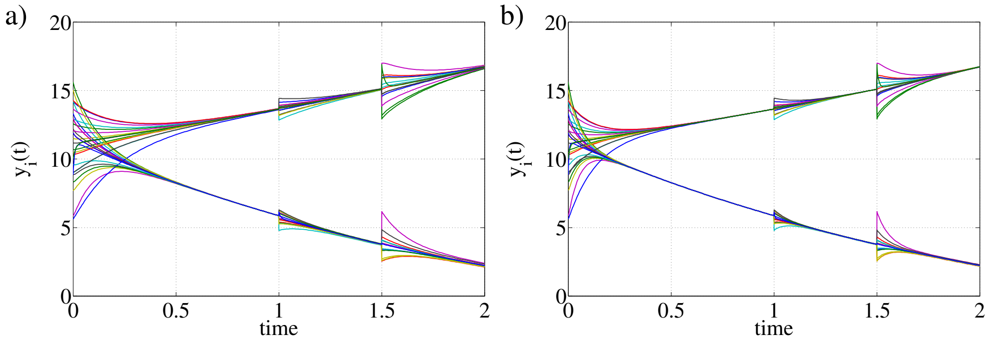


Figure 5.8.: Outputs evolution for the network analyzed in Example 5.4 with algorithm a) $\mathbf{L}_1$ and b) $\mathbf{L}_2$.

after instants when the perturbations are added, the network response is closer to the optimal. This is clearly explained by the convergence rate indicator λ_A which is greater in the second case. This also implies that the three norms defined in the table are smaller for a better algorithm. ■

5.2. Other Algorithms

Up to this point, only loopless Laplacian algorithms have been analyzed. This kind of algorithm is by far the most common one in the consensus field. However, the definition of algorithm as a linear feedback gives space to study other control strategies to deal with specific characteristics of the agents. In this section different kinds of alternative algorithms are proposed and their main characteristics are described.

5.2.1. Self-looped Algorithms

In this section selfloops in the graph that describes the algorithm are considered. The motivation for this is derived from the previous discussion about the role of the dynamics of the states of the systems in the consensus error. It is desired to have alternatives to minimize the effect of the residual signal $\mathbf{r} = \mathbf{R}_A\mathbf{x}$ over the dynamics of the error $\mathbf{e}$. Therefore, here local feedback for each agent in an AAN is proposed to directly modify the dynamics of the states.

Definition 5.2.1. A *selflooped Laplacian consensus algorithm* is a linear output feedback $\mathbf{u} = \mathbf{L}\mathbf{y}$ for system (4.3) where the feedback matrix is derived as $\mathbf{L} = -L(\mathscr{G}_w) \in \mathbb{R}^{Nq\times Nq}$ with $\mathscr{G}_w$ an undirected weighted graph over the vertices set $\mathscr{V}$ that considers selfloops in some or all of its vertices.

The feedback matrix of this kind of algorithms can be expressed as:

$$\mathbf{L} = -L(\mathscr{G}_w) = \begin{bmatrix} -\Delta_1 & \mathbf{W}_{12} & \cdots & \mathbf{W}_{1N} \\ \mathbf{W}_{12} & -\Delta_2 & \cdots & \mathbf{W}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{W}_{1N} & \mathbf{W}_{2N} & \cdots & -\Delta_N \end{bmatrix},$$

where $\mathbf{W}_{ij} = \mathbf{W}'_{ij} = w((i,j))$ and $\Delta_i = \sum_{j=1}^{N} \mathbf{W}_{ij}$. Note that the block diagonal elements add the weights of all edges including the self loops. Therefore, this kind of algorithms can be decomposed as $\mathbf{L} = \mathbf{L}_c + \mathbf{L}_l$, where $\mathbf{L}_c = -\hat{L}(\mathscr{G}_w)$ and $\mathbf{L}_l = -\mathrm{diag}\{\mathbf{W}_{ii}\}_{i=1}^{N}$, so that only $\mathbf{L}_c$ has the zero sum row property and not $\mathbf{L}$. With this, and considering that $\mathbf{y} = \mathbf{T}^{+}\mathbf{e} + \mathbf{J}\mathbf{C}\mathbf{x}$, the feedback signal $\mathbf{u}$ can be written as

$$\mathbf{u} = \mathbf{L}\mathbf{y} = (\mathbf{L}_c + \mathbf{L}_l)\left(\mathbf{T}^{+}\mathbf{e} + \mathbf{J}\mathbf{C}\mathbf{x}\right) = \mathbf{L}\mathbf{T}^{+}\mathbf{e} + \mathbf{L}_l\mathbf{J}\mathbf{C}\mathbf{x}.$$

Autonomous Agents Network

As seen in section 5.1.2 an AAN does not necessarily reach consensus with a loopless algorithm. When considering that $\mathbf{x} = \mathbf{C}^+\mathbf{T}^+\mathbf{e} + (\mathbf{C}^+\mathbf{JC} + \mathbf{C}^\perp(\mathbf{C}^\perp)')\mathbf{x}$, then the dynamics of the error can be written as:

$$\dot{\mathbf{e}} = \mathbf{TC}\left(\mathbf{BL} + \mathbf{AC}^+\right)\mathbf{T}^+\mathbf{e} + \mathbf{TC}\left(\left(\mathbf{BL}_l + \mathbf{AC}^+\right)\mathbf{JC} + \mathbf{AC}^\perp(\mathbf{C}^\perp)'\right)\mathbf{x}. \tag{5.31}$$

From here, the loops of an algorithm can be used to limit the effect of the states on the dynamics of the consensus error. Indeed, if $\mathbf{L}_l = \mathbf{L}_l' = \operatorname{diag}\{\mathbf{W}_{ii}\}_{i=1}^N$ is chosen so that the residual matrix $\mathbf{R}_l := \mathbf{TC}\left(\left(\mathbf{BL}_l + \mathbf{AC}^+\right)\mathbf{JC} + \mathbf{AC}^\perp(\mathbf{C}^\perp)'\right)$ vanishes, then the dynamics of the error depend only on $\mathbf{G}_l := \mathbf{TC}\left(\mathbf{BL}_c + \mathbf{BL}_l + \mathbf{AC}^+\right)\mathbf{T}^+$.

However, because the term $\mathbf{AC}^\perp(\mathbf{C}^\perp)'$ is not necessarily zero, it is in general not possible to force $\mathbf{R}_l = \mathbf{0}$. At most, the weights on the selfloops can be chosen so that the norm of the residual matrix is minimal, what does not guarantee that consensus can be reached. Considering that the dynamics of the states are such that,

$$\mathbf{x}(t) = e^{(\mathbf{A}+\mathbf{BLC})t}\mathbf{x}_0 = e^{\mathbf{A}_{cl}t}\mathbf{x}_0,$$

then choosing $\mathbf{L}_l$ such that the product $\mathbf{r}_l = \mathbf{R}_l e^{\mathbf{A}_{cl}t} \to \mathbf{0}$ as $t \to +\infty$ will guarantee consensus in the long term if $\mathbf{G}_l$ is Hurwitz. This problem is difficult to solve (numerically and algebraically) as it is highly non-linear because both matrices, $\mathbf{R}_l$ and $e^{\mathbf{A}_{cl}t}$, depend explicitly on the unknown variable $\mathbf{L}_l$. The problem becomes more difficult when $\mathbf{L}_c$ is also a variable.

Because of this, a more heuristic approach can be followed to choose a candidate feedback matrix. In particular, if the dynamics of the agents combined with the respective selfloops can be approximated by the behavior of an integrator system, *i.e.* if it is possible to find $\mathbf{W}_{ii}$ such that $\mathbf{A}_i + \mathbf{B}_i\mathbf{W}_{ii}\mathbf{C}_i \approx \mathbf{0}$, then the network can be approximated by an IN. Slightly more general, if each agent is Hurwitz but slow, dynamical consensus can also be approximated in a good way. From,

$$\begin{aligned}\mathbf{A} + \mathbf{BXC} = -\lambda_l\mathbf{I} &\implies -\mathbf{BXC} = \mathbf{A} + \lambda_l\mathbf{I}\\ &\implies \mathbf{X} = -\mathbf{B}^+\left(\mathbf{A} + \lambda_l\mathbf{I}\right)\mathbf{C}^+,\end{aligned}$$

with $0 \leqslant \lambda_l \ll \min\{\operatorname{abs}\left\{\operatorname{real}\left\{\operatorname{eig}\left\{\mathbf{TCBL}_c\mathbf{T}^+\right\}\right\}\right\}\}$, a possible candidate for the selfloops feedback is to define

$$\mathbf{L}_l := -\mathbf{B}^+\left(\mathbf{A} + \lambda_l\mathbf{I}\right)\mathbf{C}^+. \tag{5.32}$$

Because of the block diagonal structure of the matrices, $\mathbf{L}_l$ is also block diagonal and its blocks can be computed locally by each agent making $[\mathbf{L}_l]_{ii} = -\mathbf{B}_i^+\left(\mathbf{A}_i + \lambda_l\mathbf{I}\right)\mathbf{C}_i^+$. Note that in general, as $\mathbf{C}^+\mathbf{C} \neq \mathbf{I}$ and $\mathbf{BB}^+ \neq \mathbf{I}$, $\mathbf{L}_l = -\mathbf{B}^+\left(\mathbf{A} + \lambda_l\mathbf{I}\right)\mathbf{C}^+$ does not imply that $\mathbf{A} + \mathbf{BL}_l\mathbf{C} = -\lambda_l\mathbf{I}$ and the closed loop network only approximates the desired behavior. Indeed, replacing (5.32) in (5.31), the dynamics of the error become:

$$\dot{\mathbf{e}} = \mathbf{TC}\left(\mathbf{BL}_c + \mathbf{M}_l\right)\mathbf{T}^+\mathbf{e} + \mathbf{TC}\left(\mathbf{M}_l\mathbf{JC} + \mathbf{AC}^\perp(\mathbf{C}^\perp)'\right)\mathbf{x},$$

where $\mathbf{M}_l = \left(\mathbf{I} - \mathbf{B}\mathbf{B}^+\right)\mathbf{A}\mathbf{C}^+ - \lambda_l \mathbf{B}\mathbf{B}^+\mathbf{C}^+$. The dynamics of the states are

$$\dot{\mathbf{x}} = \left(\mathbf{B}\mathbf{L}_c\mathbf{C} + \mathbf{M}_l\mathbf{C}\right)\mathbf{x}.$$

Nevertheless, observe that in the special case where $\mathbf{B}^+ = \mathbf{B}^{-1}$ and $\mathbf{C}^+ = \mathbf{C}^{-1}$, then $\mathbf{x} = \mathbf{C}^{-1}\mathbf{T}^+\mathbf{e} + \mathbf{C}^{-1}\mathbf{J}\mathbf{C}\mathbf{x}$ and, considering (5.32), the dynamics of the error and the states become:

$$\begin{aligned}
\dot{\mathbf{e}} &= \mathbf{T}\mathbf{C}\left(\mathbf{B}\left(\mathbf{L}_c + \mathbf{L}_l\right) + \mathbf{A}\mathbf{C}^{-1}\right)\mathbf{T}^+\mathbf{e} + \mathbf{T}\mathbf{C}\left(\mathbf{B}\mathbf{L}_l + \mathbf{A}\mathbf{C}^{-1}\right)\mathbf{J}\mathbf{C}\mathbf{x}\\
&= \mathbf{T}\mathbf{C}\left(\mathbf{B}\mathbf{L}_c - \mathbf{A}\mathbf{C}^{-1} - \lambda_l\mathbf{C}^{-1} + \mathbf{A}\mathbf{C}^{-1}\right)\mathbf{T}^+\mathbf{e} + \mathbf{T}\mathbf{C}\left(-\lambda_l\mathbf{C}^{-1}\right)\mathbf{J}\mathbf{C}\mathbf{x}\\
&= \left(\mathbf{T}\mathbf{C}\mathbf{B}\mathbf{L}_c\mathbf{T}^+ - \lambda_l\mathbf{I}\right)\mathbf{e} \approx \mathbf{T}\mathbf{C}\mathbf{B}\mathbf{L}_c\mathbf{T}^+\mathbf{e}
\end{aligned}$$

$$\dot{\mathbf{x}} = \left(\mathbf{B}\mathbf{L}_c\mathbf{C} - \lambda_l\mathbf{I}\right)\mathbf{x} \approx \mathbf{B}\mathbf{L}_c\mathbf{C}\mathbf{x}$$

From here, consensus can be reached by properly designing $\mathbf{L}_c = -\hat{L}(\mathscr{G}_w)$ to fulfill certain performance specification as in Section 5.1.1.

Complementary States Feedback

The previous discussion reveals that through the inclusion of self looped algorithms, at least in some cases, the behavior of the individual agents can be approximated by integrator dynamics. However, as the consensus algorithm is an output feedback, the states of the agents that are not reflected in the output cannot be controlled, making the residual matrix non zero in most of the cases.

If *all* the states of each agent are measured and are available for feedback, then the following complementary signal can be defined:

$$\mathbf{y}_\perp = (\mathbf{C}^\perp)'\mathbf{x}.$$

As $\mathbf{C}\mathbf{C}^\perp = \mathbf{0}$, this signal includes the information of the states that are not mapped into the output $\mathbf{y}$. With this, an alternative feedback law including this complementary information can be defined as:

$$\begin{aligned}
\mathbf{u} &:= \mathbf{L}\mathbf{y} + \mathbf{L}_\perp\mathbf{y}_\perp\\
&= \mathbf{L}\left(\mathbf{T}^+\mathbf{e} + \mathbf{J}\mathbf{C}\mathbf{x}\right) + \mathbf{L}_\perp(\mathbf{C}^\perp)'\mathbf{x}\\
&= \mathbf{L}\mathbf{T}^+\mathbf{e} + \left(\mathbf{L}\mathbf{J}\mathbf{C} + \mathbf{L}_\perp(\mathbf{C}^\perp)'\right)\mathbf{x}.
\end{aligned}$$

Where $\mathbf{L} = -L(\mathscr{G}_w) = \mathbf{L}_c + \mathbf{L}_l$, with $\mathscr{G}_w$ a graph with self loops at every node, so that $\mathbf{L}_c = -\hat{L}(\mathscr{G}_w)$ and $\mathbf{L}\mathbf{J} = \mathbf{L}_l\mathbf{J}$; and $\mathbf{L}_\perp \in \mathbb{R}^{n-Nq\times n-Nq}$. Replacing this expression in the dynamics of the error, we obtain:

$$\begin{aligned}
\dot{\mathbf{e}} &= \mathbf{T}\mathbf{C}\mathbf{B}\mathbf{u} + \mathbf{T}\mathbf{C}\mathbf{A}\left(\mathbf{C}^+\mathbf{T}^+\mathbf{e} + \left(\mathbf{C}^+\mathbf{J}\mathbf{C} + \mathbf{C}^\perp(\mathbf{C}^\perp)'\right)\mathbf{x}\right)\\
&= \mathbf{T}\mathbf{C}\left(\mathbf{B}\mathbf{L}_c + \mathbf{B}\mathbf{L}_l + \mathbf{A}\mathbf{C}^+\right)\mathbf{T}^+\mathbf{e} + \mathbf{T}\mathbf{C}\left(\left(\mathbf{B}\mathbf{L}_l + \mathbf{A}\mathbf{C}^+\right)\mathbf{J}\mathbf{C} + \left(\mathbf{B}\mathbf{L}_\perp + \mathbf{A}\mathbf{C}^\perp\right)(\mathbf{C}^\perp)'\right)\mathbf{x}\\
&= \mathbf{G}_l\mathbf{e} + \mathbf{R}_\perp\mathbf{x}.
\end{aligned}$$

Note that the introduction of the feedback matrix $\mathbf{L}_\perp$ does not affect the matrix $\mathbf{G}_l$. It only appears in the residual matrix. Clearly, if $\mathbf{R}_\perp$ is zero, then consensus can be reached independently of the dynamics of the states.

In terms of the complementary output $\mathbf{y}_\perp = (\mathbf{C}^\perp)'\mathbf{x}$ and the mean value of the outputs $\mathbf{v} = \frac{1}{N}\mathbf{1}'\mathbf{C}\mathbf{x}$, from the previous dynamic expression one can also write

$$\dot{\mathbf{e}} = \mathbf{G}_l\mathbf{e} + \mathbf{TC}\left(\mathbf{BL}_l + \mathbf{AC}^+\right)\mathbf{1v} + \mathbf{TC}\left(\mathbf{BL}_\perp + \mathbf{AC}^\perp\right)\mathbf{y}_\perp.$$

From here is clear that the selfloops in an algorithm can be used to minimize the effect of $\mathbf{v}$ over the dynamics of the error, while the complementary feedback to minimize the effect of $\mathbf{y}_\perp$.

Algebraically, to find matrices $\mathbf{L}_l$ and $\mathbf{L}_\perp$ that make the residual matrix to vanish might be a hard task. However, numerically, it can be achieved easily by bounding the Euclidean norm of $\mathbf{R}_\perp$ by a known tolerance $\varepsilon > 0$. That is,

$$\mathbf{R}_\perp(\mathbf{R}_\perp)' < \varepsilon^2\mathbf{I}.$$

Applying Schur's complement, the previous expression is equivalent to the following LMI:

$$\left[\begin{array}{c|c} -\varepsilon^2\mathbf{I} & \mathbf{TC}\left(\left(\mathbf{BL}_l + \mathbf{AC}^+\right)\mathbf{JC} + \left(\mathbf{BL}_\perp + \mathbf{AC}^\perp\right)(\mathbf{C}^\perp)'\right) \\ \hline \star & -\mathbf{I} \end{array}\right] < 0 \tag{5.33}$$

Fixing the tolerance value, the feasibility problem of LMI (5.33) can be solved to find the desired variables $\mathbf{L}_l$ and $\mathbf{L}_\perp$ so that $\|\mathbf{R}_\perp\|$ is arbitrarily small. Note that imposing structural restrictions on the variables, like for example imposing that $\mathbf{L}_\perp$ is derived as the Laplacian of a graph, makes it harder to solve the corresponding feasibility problem.

The self loops of an algorithm and the complementary states feedback are used to minimize the effect of the residual matrix over the error dynamics. In more simple words, the effect achieved is to approximate a decoupled system where consensus depends only on the error and not any other internal variables. Consensus however still relies on the edges between nodes of the graph describing the algorithms.

It could be argued that simply considering a graph with only self loops to define the consensus algorithm, *i.e.* without interconnections between the agents neither as output feedback nor as complementary states feedback, would lead to consensus as matrix $\mathbf{G}_l$ can still be forced to be Hurwitz. This is indeed true when $\mathbf{x}$ approaches the origin so that $\mathbf{r} = \mathbf{R}_l\mathbf{x}$ vanishes and allows the network to reach trivial consensus. However, this kind of control strategy is very weak with respect to perturbations and external signals. If no interconnections are considered then an arbitrary local modification of the input of any agent will irremediable lead to a difference between the outputs. The same would happen if agents are under the influence of other kind of external perturbations. For this reason, the discussed strategies can only be used to complement the interconnections in a AAN and not to replace them.

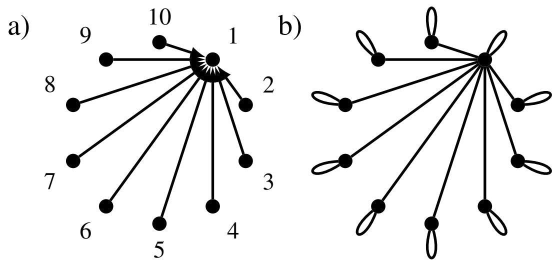


Figure 5.9.: a) Organization tree $\mathscr{T}^o$, b) Looped graph $\mathscr{T}$ for Example 5.5.

Example 5.5. Consider network h) in Example 5.3, that is, $N = 10$ agents with identical dynamics given by:

$$\mathbf{A}_i = \begin{bmatrix} 0.1 & -0.1 \\ 0.1 & 0.0 \end{bmatrix}, \mathbf{B}_i = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}, \mathbf{C}_i = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}.$$

As shown in the quoted example, with a loopless Laplacian algorithm it is difficult for this network to reach consensus because of the unstable dynamics of the agents. This can be solved by considering looped algorithms and complementary state feedback.

We consider the organization matrix $\mathbf{T} = D'(\mathscr{T}^o)$ derived from the oriented tree $\mathscr{T}^o$ in Figure 5.9 a). The undirected looped graph $\mathscr{T} = (\mathscr{V}, \mathscr{E})$ in Figure 5.9 b) can be used as a base for a looped consensus algorithm $\mathbf{L} = -L(\mathscr{T}_w)$ defining the weighted undirected graph $\mathscr{T}_w = (\mathscr{V}, \mathscr{E}, w)$, where $\forall i \in \mathscr{V}$, $w((i,i)) = \mathbf{B}_i^+ \mathbf{A}_i \mathbf{C}_i^+ = 0.05$ and $\forall j \in \mathscr{V} \backslash \{1\}$, $w((1,j)) = 10$.

However, with this pure loop control law, the network still cannot reach consensus because of the unbounded residual signal $\mathbf{r} = \mathbf{R}_l \mathbf{x}$. This can be seen in Figure 5.10 a) where the response of the network under identical initial conditions as in Example 5.3 is shown. This response does not differ substantially from the loop-less case in Example 5.3, and only approximates consensus before the error diverges.

Solving the feasibility problem of LMI (5.33) with $\varepsilon = 10^{-5}$ and $\mathbf{L}_l = \mathbf{B}^+ \mathbf{A} \mathbf{C}^+$, a complementary state feedback $\mathbf{L}_\perp$ can be calculated. Unfortunately, this matrix cannot be forced to have a particular structure. Imposing $\mathbf{L}_\perp$ to be diagonal, symmetric, or with rows that add up to zero, makes the LMI unfeasible in all three cases. However, the following matrix can still be obtained as a solution of the LMI feasibility problem without structural restrictions:

$$\mathbf{L}_\perp = \begin{bmatrix}
-0.0375 & -0.0375 & -0.0375 & -0.0375 & -0.0375 & -0.0375 & 0.0265 & 0.0265 & 0.1015 & -0.0485 \\
-0.0375 & -0.0375 & 0.0375 & 0.0375 & 0.0375 & 0.0375 & 0.0265 & 0.0265 & 0.1015 & -0.0485 \\
-0.0530 & -0.0530 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0375 & -0.0375 & 0.0375 & -0.1125 \\
0.0000 & 0.0000 & 0.0530 & 0.0530 & -0.0530 & -0.0530 & 0.0000 & 0.0000 & 0.0750 & -0.0750 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0530 & 0.0530 & 0.0220 & -0.1280 \\
-0.0750 & 0.0750 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0750 & -0.0750 \\
0.0000 & 0.0000 & -0.0750 & 0.0750 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0750 & -0.0750 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0750 & 0.0750 & 0.0000 & 0.0000 & 0.0750 & -0.0750 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0750 & 0.0750 & 0.0750 & -0.0750 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000
\end{bmatrix}$$

Even though the matrix does not have a Laplacian structure, the algorithm that it describes can still be distributively implemented when the corresponding signals are communicated

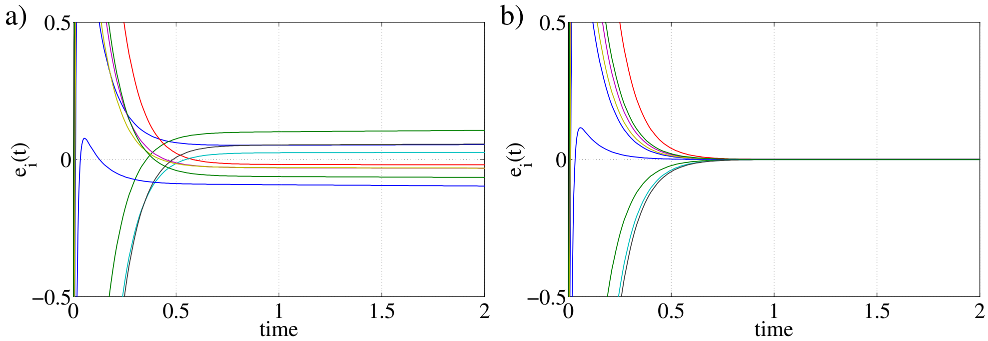


Figure 5.10.: Error evolution for the network analyzed in Example 5.5 with a) $\mathbf{L} = -L(\mathscr{T}_w)$ and b) with complementary state feedback matrix $\mathbf{L}_\perp$.

between the agents. The simulation results of the network including this control law are to be seen in Figure 5.10 b). Clearly, the network reaches consensus successfully, although the complementary feedback matrix cannot be directly associated to a graph as studied up to here. This example shows that consensus cannot be always achieved if we restrict the algorithms to be in the form of the Laplacian of an unweighted graph and, if possible, other strategies need to be developed. ■

Connected Agents Network

In Section 5.1.1 it is noticed that an algorithm can induce consensus in an IN only if rank $\{\mathbf{L}\} = (N-1)q$ (Lemma 5.3). This comes from the assumption that the product $\mathbf{CB}$ is full rank which implies that $\mathbf{G} = \mathbf{TCBLT}^+$ is Hurwitz only if the rank condition on $\mathbf{L}$ is fulfilled. Similarly, in an AAN, non trivial consensus can also only be induced by $(N-1)q$-rank algorithms. However, in some cases the assumption that $\mathbf{CB}$ is full rank may be dropped, making it possible to use full rank looped Laplacian algorithms to reach consensus.

Definition 5.2.2. A *connected agent* (CA) is an agent $i \in \mathscr{V}$ that has hardware interconnections with other agents and individual dynamics described by:

$$\begin{aligned}\dot{\mathbf{x}}_i &= \mathbf{A}_i\mathbf{x}_i + \mathbf{B}_i\mathbf{u}_i \\ \mathbf{y}_i &= \mathbf{C}_{ii}\mathbf{x}_i + \sum_{j\neq i}(\mathbf{C}_{ij}\mathbf{x}_j)\end{aligned} \tag{5.34}$$

With $\mathbf{A}_i \in \mathbb{R}^{q\times q}$, $\mathbf{B}_i \in \mathbb{R}^{q\times q}$, and $\mathbf{C}_i \in \mathbb{R}^{q\times q}$. The matrices $\mathbf{C}_{ij} \in \mathbb{R}^{q\times n_j}$ describe the hardware interconnections between agent $i \in \mathscr{V}$ and agent $j \in \mathscr{V}$.

Note that it is assumed that the number of inputs is the same as the number of outputs and of states. A Connected Agents Network is then defined as:

Definition 5.2.3. A *connected agents network* (CAN) is the aggregation of N connected agents in a set $\mathscr{V}$ where the hardware interconnections between the agents are described by an undirected, weighted and connected graph $\mathscr{C}_w$ and a full rank diagonal matrix $\mathbf{M} \in \mathbb{R}^{Nq \times Nq}$. The dynamics of such a network are described by:

$$\begin{aligned} \dot{\mathbf{x}} =& \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} =& \mathbf{C}\mathbf{x} \end{aligned} \tag{5.35}$$

with,

$$\begin{aligned} &\mathbf{A} = \operatorname{diag}\{\mathbf{A}_i\}_{i=1}^{N}, \mathbf{B} = \operatorname{diag}\{\mathbf{B}_i\}_{i=1}^{N}, \mathbf{C} = \mathbf{M}\hat{L}(\mathscr{C}_w) \in \mathbb{R}^{Nq \times Nq}, \\ &\mathbf{x} = \operatorname{col}\{\mathbf{x}_i\}_{i=1}^{N}, \mathbf{u} = \operatorname{col}\{\mathbf{u}_i\}_{i=1}^{N}, \mathbf{y} = \operatorname{col}\{\mathbf{y}_i\}_{i=1}^{N} \end{aligned}$$

We assume that $\mathbf{B}$ is full rank and therefore its inverse exists. Note that $\mathbf{A} \in \mathbb{R}^{Nq \times Nq}$, $\mathbf{B} \in \mathbb{R}^{Nq \times Nq}$, $\mathbf{C} \in \mathbb{R}^{Nq \times Nq}$ and, $\mathbf{CJ} = \mathbf{M}\hat{L}(\mathscr{C}_w)\mathbf{1}\mathbf{1}'\frac{1}{N} = \mathbf{0}$.

For this kind of networks a self looped algorithm derived from an undirected weighted graph $\mathscr{S}_w$ with *only* self loops can be used to reach consensus. That is, an algorithm that avoids the need to communicate signals between the agents.

To demonstrate the previous statement we need to quote the properties of the pseudo inverse of $\mathbf{C}$ with respect to its singular values decomposition (SVD, see Appendix A.1). Note that as $\mathscr{C}_w$ is connected, then $\operatorname{rank}\{\mathbf{C}\} = \min\{\operatorname{rank}\{\mathbf{M}\}, \operatorname{rank}\{\hat{L}(\mathscr{C}_w)\}\} = (N-1)q$. Additionally $\mathbf{C}'\mathbf{C}\mathbf{1} = \hat{L}(\mathscr{C}_w)\mathbf{M}\mathbf{M}\hat{L}(\mathscr{C}_w)\mathbf{1} = \mathbf{0}$ and $\mathbf{1}'\mathbf{1} = N\mathbf{I}$. So the columns of $\mathbf{V}_z = \frac{1}{\sqrt{N}}\mathbf{1}$ are the orthonormal eigenvectors associated with the q zero eigenvalues of $\mathbf{C}'\mathbf{C}$. From Corollary A.7, with $\mathbf{V} = \begin{bmatrix}\mathbf{V}_+ & \mathbf{V}_z\end{bmatrix}$ for some matrix $\mathbf{V}_+$ consistent of orthonormal eigenvectors of $\mathbf{C}'\mathbf{C}$, it is easy to show that

$$\begin{aligned} \mathbf{C}^+\mathbf{C} &= \mathbf{V}\begin{bmatrix}\mathbf{I}_{(N-1)q\times(N-1)q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{q\times q}\end{bmatrix}\mathbf{V}' \\ &= \begin{bmatrix}\mathbf{V}_+ & \mathbf{V}_z\end{bmatrix}\left(\mathbf{I}_{Nq\times Nq} - \begin{bmatrix}\mathbf{0}_{(N-1)q\times q} \\ \mathbf{I}_{q\times q}\end{bmatrix}\begin{bmatrix}\mathbf{0}_{q\times(N-1)q} & \mathbf{I}_{q\times q}\end{bmatrix}\right)\begin{bmatrix}\mathbf{V}'_+ \\ \mathbf{V}'_z\end{bmatrix} \\ &= \mathbf{V}\mathbf{V}' - \begin{bmatrix}\mathbf{V}_+ & \mathbf{V}_z\end{bmatrix}\begin{bmatrix}\mathbf{0}_{(N-1)q\times q} \\ \mathbf{I}_{q\times q}\end{bmatrix}\begin{bmatrix}\mathbf{0}_{q\times(N-1)q} & \mathbf{I}_{q\times q}\end{bmatrix}\begin{bmatrix}\mathbf{V}'_+ \\ \mathbf{V}'_z\end{bmatrix} \\ &= \mathbf{I} - \mathbf{V}_z\mathbf{V}'_z = \mathbf{I} - \frac{1}{\sqrt{N}}\mathbf{1}\frac{1}{\sqrt{N}}\mathbf{1}' \\ &= \mathbf{I} - \mathbf{J} \end{aligned}$$

From here an inverse relationship between the states and the outputs of the systems can be stated:

$$\mathbf{y} = \mathbf{C}\mathbf{x} \implies \mathbf{C}^+\mathbf{y} = \mathbf{C}^+\mathbf{C}\mathbf{x} \implies \mathbf{x} = \mathbf{C}^+\mathbf{y} + \mathbf{J}\mathbf{x}.$$

This, together with the error equation, $\mathbf{e} = \mathbf{T}\mathbf{y} \implies \mathbf{y} = \mathbf{T}^{+}\mathbf{e} + \mathbf{J}\mathbf{C}\mathbf{x}$, leads to:

$$\begin{aligned}\dot{\mathbf{e}} &= \mathbf{TCBu} + \mathbf{TCAx} \\ &= \mathbf{TC}\left(\mathbf{BL} + \mathbf{AC}^{+}\right)\mathbf{T}^{+}\mathbf{e} + \mathbf{TC}\left(\mathbf{BLJC} + \mathbf{A}\left(\mathbf{C}^{+}\mathbf{JC} + \mathbf{J}\right)\right)\mathbf{x}\end{aligned}$$

When the feedback matrix is chosen so that $\mathbf{L} = \mathbf{L}_l = -\alpha\mathbf{B}^{-1} = -L(\mathscr{S}_w)$, with $\alpha > 0$ a scalar, then $\mathbf{TCBLJC} = -\alpha\mathbf{TCJC} = \mathbf{0}$ and the expression for the dynamics of the error is reduced to

$$\dot{\mathbf{e}} = \mathbf{TC}\left(-\alpha\mathbf{I} + \mathbf{AC}^{+}\right)\mathbf{T}^{+}\mathbf{e} + \mathbf{TCA}\left(\mathbf{C}^{+}\mathbf{JC} + \mathbf{J}\right)\mathbf{x}. \tag{5.36}$$

From (5.36) is clear that non trivial consensus can be reached if the residual signal $\mathbf{r} = \mathbf{TCA}\left(\mathbf{C}^{+}\mathbf{JC} + \mathbf{J}\right)\mathbf{x}$ vanishes with time. In the special case where, for given matrices $\mathbf{A}_0 \in \mathbb{R}^{q\times q}$ and $\mathbf{M}_0 \in \mathbb{R}^{q\times q}$, $\mathbf{A} = \operatorname{diag}\{\mathbf{A}_0\}_{i\in\mathscr{V}}$ and $\mathbf{M} = \operatorname{diag}\{\mathbf{M}_0\}_{i\in\mathscr{V}}$, *i.e.* is when the agents are all identical except possible for the matrices $\mathbf{B}_i$, then $\mathbf{AJ} = \frac{1}{N}\mathbf{1}\mathbf{A}_0\mathbf{1}' = \mathbf{JA}$ and $\mathbf{JM} = \frac{1}{N}\mathbf{1}\mathbf{M}_0\mathbf{1}' = \mathbf{MJ}$ making $\mathbf{TCA}\left(\mathbf{C}^{+}\mathbf{JC} + \mathbf{J}\right) = \mathbf{TCAC}^{+}\mathbf{MJ}\hat{L}(\mathscr{C}_w) + \mathbf{TCJA} = \mathbf{0}$.

Contrary to the case of an AAN, due to the hardware interconnections between the agents, a controller based on a graph with only self loops is still robust against external signals. To verify this, let us consider an external control input $\mathbf{u}_{ext} : \mathbb{R}_0^{+} \mapsto \mathbb{R}^{Nq}$ so that $\mathbf{u} = \mathbf{Ly} + \mathbf{u}_{ext} = -\alpha\mathbf{B}^{-1}\mathbf{y} + \mathbf{u}_{ext}$. In that case, the dynamics of the error become

$$\dot{\mathbf{e}} = \mathbf{TC}\left(-\alpha\mathbf{I} + \mathbf{AC}^{+}\right)\mathbf{T}^{+}\mathbf{e} + \mathbf{TCA}\left(\mathbf{C}^{+}\mathbf{JC} + \mathbf{J}\right)\mathbf{x} + \mathbf{TCBu}_{ext}.$$

Considering that agents are identical so that $\mathbf{A} = \operatorname{diag}\{\mathbf{A}_0\}_{i\in\mathscr{V}}$ and $\mathbf{M} = \operatorname{diag}\{\mathbf{M}_0\}_{i\in\mathscr{V}}$, then the previous expression is simplified to:

$$\dot{\mathbf{e}} = \mathbf{TC}\left(-\alpha\mathbf{I} + \mathbf{AC}^{+}\right)\mathbf{T}^{+}\mathbf{e} + \mathbf{TCBu}_{ext}. \tag{5.37}$$

Expression (5.37) can be used to quantify the influence of the external input over the consensus error by considering the H_∞-norm of the transfer function matrix $H_{ext}(s)$ between the external signal and the error as in the previous sections. Note that, regardless of whether matrix $\mathbf{G}_\alpha = \mathbf{TC}\left(-\alpha\mathbf{I} + \mathbf{AC}^{+}\right)\mathbf{T}^{+}$ is Hurwitz, if the external signal is such that $\lim_{t\to+\infty}\mathbf{TCBu}_{ext} \neq \mathbf{0}$, consensus cannot be reached as then $\lim_{t\to+\infty}\mathbf{e} \neq \mathbf{0}$. However, even in these cases, the parameter α characterizes the behavior of the network.

A similar analysis can be made by considering an external perturbation in the form of $\mathbf{y} = \mathbf{Cx} + \mathbf{d}$. In this case both, the perturbation signal $\mathbf{d}$ and its change over time $\mathbf{w} = \frac{d}{dt}\mathbf{d}$ needs to be taken into account when developing an expression for the dynamics of the error.

Example 5.6. Consider a CAN with $N = 7$ agents with identical dynamics and $q = 1$. The hardware interconnections are described by the undirected weighted loopless graph $\mathscr{C}_w = (\mathscr{C}, w)$, where $\mathscr{C} = (\mathscr{V}, \mathscr{E})$ is the graph given in Figure 5.11 a) and $\forall(i,j) \in \mathscr{E}$, $w((i,j)) = 1.0$. Furthermore, the dynamics of the network are fully described by the matrices:

$$\mathbf{A} = \operatorname{diag}\{0.1\}_{i\in\mathscr{V}}, \mathbf{B} = \operatorname{diag}\{1.0\}_{i\in\mathscr{V}} \text{ and, } \mathbf{M} = \operatorname{diag}\{2.0\}_{i\in\mathscr{V}}.$$

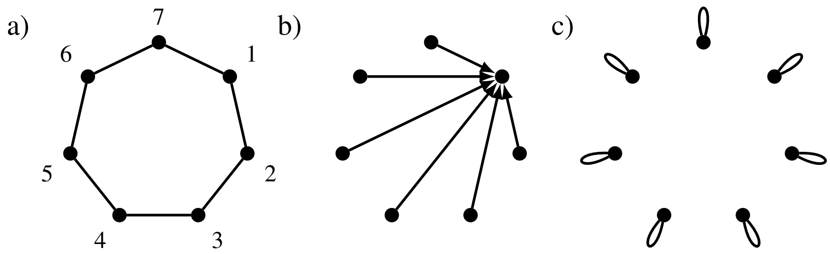


Figure 5.11.: a) Hardware interconnections graph $\mathscr{C}$, b) Organization tree $\mathscr{T}^o$, and c) pure loops graph $\mathscr{S}$ for Example 5.6.

The organization to be used is derived from the directed tree in Figure 5.11 b). We also consider the algorithm $\mathbf{L} = -L(\mathscr{S}_v) = -\alpha\mathbf{I}$, where $\mathscr{S}_v = (\mathscr{S}, v)$. $\mathscr{S} = (\mathscr{V}, \mathscr{E}_s)$ is the looped undirected graph in Figure 5.11 c) and $\forall(i,i) \in \mathscr{E}_s$, $v((i,i)) = \alpha > 0$.

The H_∞-norm of the transfer function between the external signal and the error can be studied as a function of the design parameter α as shown in Figure 5.12. The value of $\|H_{ext}(s)\|_\infty$ peaks for the values of α where the matrix $\mathbf{G}_\alpha = \mathbf{TC}\left(-\alpha\mathbf{I} + \mathbf{AC}^+\right)\mathbf{T}^+$ is close to be singular increasing the corresponding DC gain of the system. Nevertheless, besides from these points, there is a clear inverse relationship between α and the H_∞-norm.

The two highlighted points in the graph correspond to the algorithms defined by $\alpha = 0.5$ and $\alpha = 5.0$. Note that for the last value, $\|H_{ext}(s)\|_\infty < 1$, and therefore the network damps the action of the external signal instead of amplifying it as with the first value. A time simulation of the behavior of the network in both cases is shown in Figure 5.13. The agents start with identical initial conditions and (only) the first agent is submitted to the action of an external signal $\mathbf{u}_{ext,1}$, which is changed every four seconds identically in both cases.

As expected, it is clear from the graphs that the network cannot reach consensus under these conditions. Independently of the chosen algorithm, as the transfer function between the external input and the error cannot be forced to be identically zero, the error vector always reaches certain value in steady state. However, there are clear differences between the performance of both cases. First, due to the smaller gain when $\alpha = 5.0$, the differences between the outputs are in this case around ten times smaller than when $\alpha = 0.5$. Secondly, as the eigenvalues of matrix $\mathbf{G}_\alpha$ are further to the left in the second case, the network reacts faster to the changes reaching steady state sooner and minimizing the transient error between the signals. In simpler words, by increasing the parameter α the network becomes "heavier", making it more resilient to external signals. ■

5.2.2. Non Graphically Restricted Algorithms

With the introduction of the complementary states feedback, it has been shown that a consensus algorithm can be selected in such a way that there is no immediate graph representation of it. The idea of selfloops somehow also generalizes consensus algorithms to the use of more

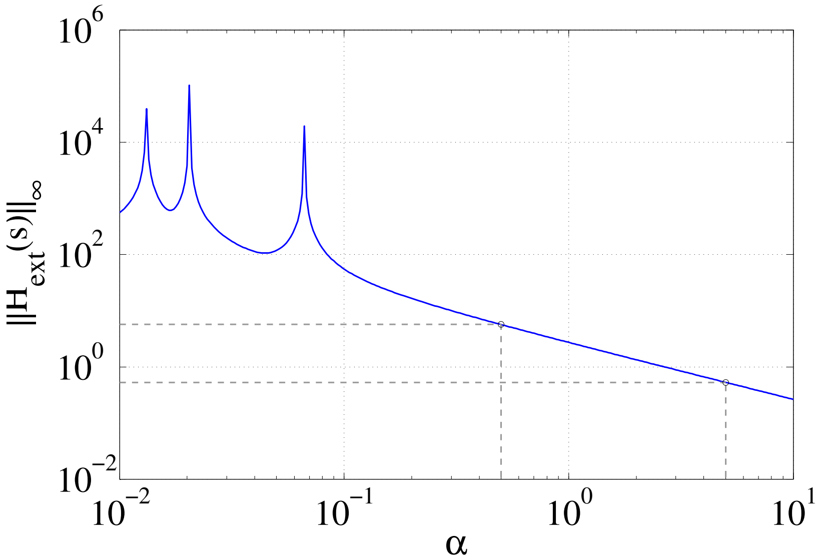


Figure 5.12.: H_∞-norm of the transfer function matrix $H_{ext}(s)$ as a function of parameter α in Example 5.6.

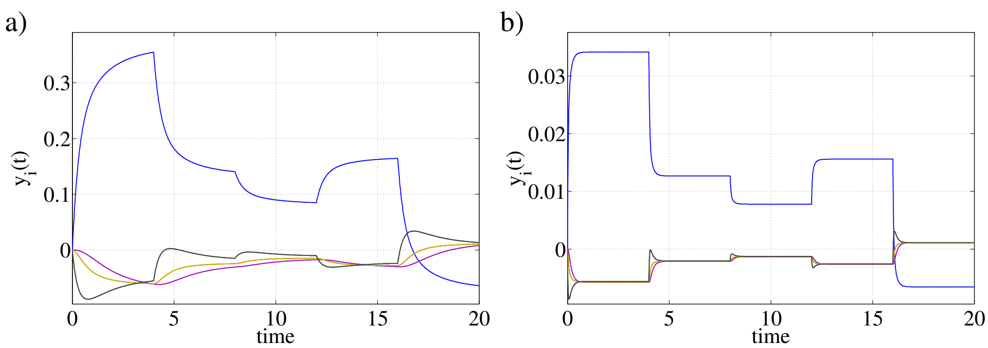


Figure 5.13.: Outputs evolution for the network analyzed in Example 5.6 with $\mathbf{L} = -L(\mathscr{S}_v) = -\alpha\mathbf{I}$ for a) $\alpha = 0.5$ and b) $\alpha = 5.0$.

complex matrices, namely, without the row zero sum property. Indeed, a consensus algorithm could be specified merely by the gains in a matrix $\mathbf{L}$ not associated with any other mathematical entity like a graph. Furthermore, in the formulation of this thesis, the analysis of the behavior of the network under a given algorithm is not done through graph related tools but only in terms of classical control theory. Therefore, the use of graph theory to describe an algorithm is justified only because it might simplify its representation for human analysis.

As an effort to include this more complex cases without loosing the convenient graph representation of an algorithm, a common generalization is through Laplacian matrices of *directed graphs* or *digraphs*. See for example [22, 33, 34]. Following this idea, one can define two different kind of algorithms.

Definition 5.2.4.1. An *incoming consensus algorithm* is a linear output feedback $\mathbf{u} = \mathbf{L}\mathbf{y}$ for system (4.3) where the feedback matrix $\mathbf{L} \in \mathbb{R}^{Nq \times Nq}$ is such that $\mathbf{L}\mathbf{1} = \mathbf{0}$, *i.e.*, it has the zero row sum property.

Definition 5.2.4.2. An *outgoing consensus algorithm* is a linear output feedback $\mathbf{u} = \mathbf{L}\mathbf{y}$ for system (4.3) where the feedback matrix $\mathbf{L} \in \mathbb{R}^{Nq \times Nq}$ is such that $\mathbf{1}'\mathbf{L} = \mathbf{0}$, *i.e.*, it has the zero column sum property.

Note that there is no restriction on the signs or structure of the elements of incoming or outgoing algorithms as in the case of algorithms derived from undirected graphs. Furthermore, observe that any arbitrary feedback matrix $\mathbf{L} \in \mathbb{R}^{Nq \times Nq}$ can be decomposed as the sum of a block diagonal matrix matrix $\mathbf{L}_d = \operatorname{diag}\{\mathbf{L}_{ii}\}_{i \in \mathscr{V}}$, with $\mathbf{L}_{ii} \in \mathbb{R}^{q \times q}$, an incoming consensus algorithm $\mathbf{L}_{in}$, and an outgoing consensus algorithm $\mathbf{L}_{out}$. That is, $\mathbf{L} = \mathbf{L}_d + \mathbf{L}_{in} + \mathbf{L}_{out}$. In any case, for an AAN, the dynamics of the error defined through an organization matrix $\mathbf{T} = D'(\mathscr{T}^o)$ can be expressed as:

$$\dot{\mathbf{e}} = \mathbf{TC}\left(\mathbf{BL} + \mathbf{AC}^{+}\right)\mathbf{T}^{+}\mathbf{e} + \mathbf{TC}\left(\mathbf{BLJC} + \mathbf{A}\left(\mathbf{C}^{+}\mathbf{JC} + \mathbf{C}^{\perp}(\mathbf{C}^{\perp})'\right)\right)\mathbf{x}.$$

As for incoming algorithms the product $\mathbf{LJ}$ vanishes, it seems natural to prefer them over outgoing algorithms. Particularly when $\mathbf{R}_A = \mathbf{TCA}\left(\mathbf{C}^{+}\mathbf{JC} + \mathbf{C}^{\perp}(\mathbf{C}^{\perp})'\right) = \mathbf{0}$. In [22, pp. 26] an intuitive justification for this choice is also given in terms that the incoming algorithm captures more directly how the dynamics of an agent are influenced by others while the out-degree version captures better how one agent influences others. However, when matrix $\mathbf{R}_A$ does not vanish, outgoing algorithms are no worse than any other choice in the sense that the dynamics of the error still depend on the coupled states of the agents.

Similar to the case of selfloops in undirected algorithms, in general, an outgoing algorithm cannot be used to cancel the effect of the residual matrix $\mathbf{R}_A$ because of the terms including the null space base $\mathbf{C}^{\perp}$ for the output matrix $\mathbf{C}$. A particular case is when $Nq = n$ and the output matrix is invertible, *i.e.* $\mathbf{C}^{+} = \mathbf{C}^{-1}$. In that case, the dynamics of the error are simplified to:

$$\dot{\mathbf{e}} = \mathbf{TC}\left(\mathbf{BL} + \mathbf{AC}^{-1}\right)\mathbf{T}^{+}\mathbf{e} + \mathbf{TC}\left(\mathbf{BL} + \mathbf{AC}^{-1}\right)\mathbf{JCx}$$

Therefore, if the algorithm is chosen in such a way that the residual matrix vanishes and the system matrix $\mathbf{G}_A = \mathbf{TC}\left(\mathbf{BL}+\mathbf{AC}^{-1}\right)\mathbf{T}^+$ is Hurwitz, then consensus can be reached in spite of the characteristics of the agents.

To numerically determinate a suitable matrix $\mathbf{L}$, an LMI can be stated to quadratically bound the norm of $\mathbf{R}_A$. That is, to enforce that $\mathbf{R}_A\mathbf{R}_A' < \varepsilon^2\mathbf{I}$, with $\varepsilon > 0$ a design scalar, which is by Shur's complement equivalent to

$$\begin{bmatrix} -\varepsilon^2\mathbf{I} & \mathbf{TC}\left(\mathbf{BL}+\mathbf{AC}^{-1}\right)\mathbf{JC} \\ \star & -\mathbf{I} \end{bmatrix} < 0 \tag{5.38}$$

To fulfill this condition does however not ensure that the system matrix is Hurwitz. To do that through an unknown Lyapunov matrix $\mathbf{P} = \mathbf{P}' > 0$ leads to non linear conditions. These cannot be linearized by considering the inverse of the Lyapunov matrix because in (5.38), matrix $\mathbf{L}$ is multiplied to the left and to the right by singular matrices. Furthermore, forcing the feedback matrix to have the shape $\mathbf{L} = \mathbf{XT}$ so that $\mathbf{LT}^+\mathbf{P}^{-1} = \mathbf{XP}^{-1} := \mathbf{Z}$ is also not a valid condition as then the product $\mathbf{LJ} = \mathbf{XTJ} = \mathbf{0}$ vanishes in (5.38). Therefore, the only remaining option is to fix the Lyapunov matrix to a known constant. In particular $\mathbf{P} = \mathbf{I}$ leads to the same situation as in Section 5.1.1 for the calculation of the convergence rate of an undirected algorithm for an IN. From condition (5.10), the following LMI is obtained:

$$\mathbf{TC}\left(\mathbf{BL}+\mathbf{AC}^+\right)\mathbf{T}^+ + (\mathbf{T}^+)'\left(\mathbf{BL}+\mathbf{AC}^+\right)'\mathbf{C}'\mathbf{T}' + 2\varsigma\mathbf{I} < 0, \tag{5.39}$$

with a design scalar $\varsigma > 0$ that represents the desired convergence rate of the consensus error norm. If matrix $\mathbf{L}$ needs to be in the shape of an outgoing algorithm, structural restrictions also need to be imposed. This can be easily achieved by imposing that $\mathbf{1}'\mathbf{L} = \mathbf{0}$. If these three minimal conditions are simultaneously feasible, then an outgoing algorithm that ensures consensus can be numerically determined. Additional structural conditions can also be imposed to shape the algorithm according to other criteria. For instance, to impose unidirectional exchange of signals.

Example 5.7. Consider network with $N = 9$ agents such that $\forall i \in \mathscr{V}$, $\mathbf{B}_i = \mathbf{C}_i = 1$; and for $i \in \{1,2,3\}$, $\mathbf{A}_i = 0.10$, for $i \in \{4,5,6\}$, $\mathbf{A}_i = -0.10$, and for $i \in \{7,8,9\}$, $\mathbf{A}_i = 0.05$. To define the error vector, we consider an organization matrix $\mathbf{T} = \text{row}\{\mathbf{1}, -\mathbf{I}\}$.

Defining $\varepsilon = 0.1$ and $\varsigma = 1.0$, by solving the feasibility problem of LMIs (5.38) and (5.39), with additional structural restrictions $\mathbf{1}'\mathbf{L} = \mathbf{0}$ and, $\forall i \in \mathscr{V}$ and $k \in \{2,3,\ldots,N\}$, $\mathbf{L}(i,i+k) = 0$, we can obtain an outgoing algorithm characterized by:

$$\mathbf{L}_{out} = \begin{bmatrix} -1.2810 & 1.1977 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ -9.1777 & -0.2435 & 9.3378 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 1.6012 & -8.1401 & -1.4411 & 7.8967 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 1.4012 & 1.1977 & -7.8967 & -1.2411 & 6.6556 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 1.4012 & 1.1977 & 0.0000 & -6.6556 & -1.2411 & 5.4145 & 0.0000 & 0.0000 & 0.0000 \\ 1.4012 & 1.1977 & 0.0000 & 0.0000 & -5.4145 & -1.2411 & 4.1733 & 0.0000 & 0.0000 \\ 1.5512 & 1.1977 & 0.0000 & 0.0000 & 0.0000 & -4.1733 & -1.3911 & 2.7822 & 0.0000 \\ 1.5512 & 1.1977 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -2.7822 & -1.3911 & 1.3911 \\ 1.5512 & 1.1977 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -1.3911 & -1.3911 \end{bmatrix}$$

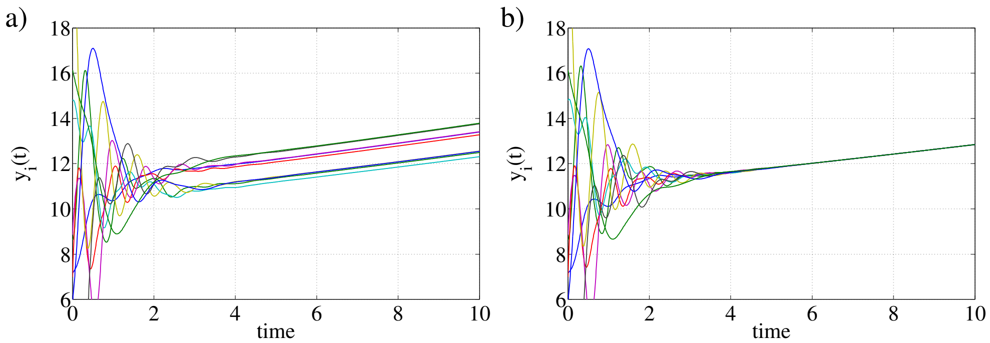


Figure 5.14.: Outputs evolution for the network analyzed in Example 5.7 with a) incoming algorithm $\mathbf{L}_{in}$ and b) outgoing algorithm $\mathbf{L}_{out}$.

Furthermore, from this matrix, an incoming feedback matrix $\mathbf{L}_{in}$ can also be defined by replacing the diagonal elements of $\mathbf{L}_{out}$ in such a way that the rows (and not the columns) add up to zero. Note that the signs of the off-diagonal elements of the feedback matrices are not necessarily positive.

In Figure 5.14 a simulation of the behavior of the network is shown under the influence of both algorithms with identical initial conditions. Note that in the case that the agents were simple integrators, the incoming algorithm would guarantee consensus. However, this is not the case for this network where the designed outgoing algorithm can successfully lead to consensus but not its incoming version. ■

5.2.3. Switched Algorithms

Most of current work on consensus is focused on switching communication topologies. There is a long list of examples, some of which are [9, 19, 22, 31, 41, 47, 50–52, 55, 57–61]. Different criteria can be applied to define the meaning of the discrete modes and how the systems switch from one discrete mode to other. Furthermore, different approaches can be used to deal with the switching characteristics of the systems. This makes consensus under switching restrictions a challenging research field, even when considering only integrator systems.

In this section we focus mainly on intended changes of the consensus algorithm that defines different operating discrete modes. The case where the consensus algorithm changes in an unwanted or unplanned way (due to communication faults for example) will be discussed at the end of this section. Attending the designer's freedom to specify an algorithm in the way it best fits his needs, the case of switched algorithms becomes relevant when, for some (technical) reason, a connected algorithm cannot be specified. For example, if the agents only support a limited number of incoming communication signals. Other reason to do this would be to avoid using a unique algorithm for tactical reasons. For example, if it is known that the communication process can be interrupted by an external "enemy" agent, switching from one controller to other would make more difficult for the external agent to identify the needed communication channels, thus improving the resilience of the network to the "attacks" of the agent. Strategic reasons can also justify a switching algorithm. For example, to approximate consensus in different sections of the network with a "dense" algorithm (with many edges between the agents of each section only), before switching to a less dense algorithm that connects all agents in the network. In this section, however, we do not emphasize the reasons why a given switched algorithm is considered but merely on conditions that guaranteed consensus under these circumstances. To do this, we consider the original results in Chapter 3 and the following definition of switching algorithm.

Definition 5.2.5. A *switching algorithm* is a control law $\mathbf{u} = \mathbf{L}_{\hat{\imath}}\mathbf{y}$ for an AAN that switches between M feedback matrices $\mathbf{L}_{\hat{\imath}} \in \mathbb{R}^{Nq\times Nq}$ at switching instants τ_k in an infinite (but known) *switching instants set* $S_\infty = \{\tau_1, \tau_2, ..., \tau_k, ...\}$. Each feedback matrix is associated with a *discrete mode* $q_{\hat{\imath}} \in Q = \{q_1, q_2, ..., q_M\}$.

Particularly, we investigate the case where, for each of the M discrete modes $q_{\hat{\imath}} \in Q$ defined in a set $Q = \{q_1, q_2, ..., q_M\}$, one particular feedback matrix $\mathbf{L}_{\hat{\imath}} = -\hat{L}(\mathscr{G}_{w,\hat{\imath}})$ is defined from a weighted undirected graph $\mathscr{G}_{w,\hat{\imath}}$; and, considering an organization matrix derived from a tree $\mathbf{T} = D'(\mathscr{T}^o)$, to avoid the influence of the states over the consensus error, the residual matrix is identically zero, *i.e.* $\mathbf{R}_A = \mathbf{TCA}\left(\mathbf{C}^+\mathbf{JC} + \mathbf{C}^\perp(\mathbf{C}^\perp)'\right) = \mathbf{0}$. In that case, the dynamics of target error are described by the following switched system:

$$\dot{\mathbf{e}} = \mathbf{TC}\left(\mathbf{BL}_{\hat{\imath}} + \mathbf{AC}^+\right)\mathbf{T}^+\mathbf{e} = \mathbf{G}_{\hat{\imath}}\mathbf{e}. \tag{5.40}$$

Clearly, the stability of this switched system can be addressed through the various cases studied in Chapter 3. As the exact switching instants of these system are decided as part of the consensus algorithm, of particular interest is the time dependent switching case with known residence times.

Defining a polytope $\mathbf{G}(\boldsymbol{\alpha})$ with vertices given by $\mathbf{G}_{\hat{\imath}}$, with $\hat{\imath} \in \{1, 2, ..., M\}$, if at an instant $\tau_k \in S_\infty = \{\tau_1, \tau_2, ...\}$ the system switches to mode $q_{\hat{\imath}} \in Q$, then the value of the error at instant $\tau_{k+1} \in S_\infty$ can be calculated as

$$\begin{aligned}\mathbf{e}(\tau_{k+1}) &= e^{\mathbf{G}(\boldsymbol{\alpha}_k)T_k}\mathbf{e}(\tau_k)\\ &= e^{\mathbf{G}_{\hat{\imath}}T_k}\mathbf{e}(\tau_k)\\ &= \boldsymbol{\Phi}_k\mathbf{e}(\tau_k),\end{aligned}$$

where the transition matrix $\boldsymbol{\Phi}_k$ associated with mode $q_{\hat{\imath}} \in Q$ is implicitly defined.

Periodic Algorithms

A first approach is to consider only deterministic switching in the form of a periodic process as introduced in Example 3.2. We will understand a periodic process as a switched systems where the only allowed sequence of discrete modes is $q_1 \to q_2 \to \cdots \to q_{M-1} \to q_M \to q_1 \to \cdots$ that repeats infinitely. That is, a system that can be represented by the automata shown in Figure 5.15.

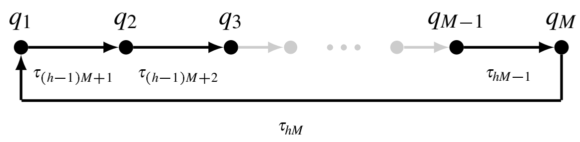


Figure 5.15.: General Periodic Process with M discrete modes.

Definition 5.2.6. A *periodic algorithm* is a switching algorithm that associates with each of the M discrete modes $q_{\hat{\imath}} \in Q = \{q_1, q_2, \ldots, q_M\}$, a feedback matrix $\mathbf{L}_{\hat{\imath}} \in \mathbb{R}^{Nq \times Nq}$ and a unique residence time $T_{(\hat{\imath})} := T_k = \tau_{k+1} - \tau_k \in \mathbb{R}^+$, for all $k \in \mathbb{N}$ where the algorithm is switched to mode $q_{\hat{\imath}}$; and that switches between the modes in a unique infinitely repetitive sequence $q_1 \to q_2 \to \cdots \to q_{M-1} \to q_M \to q_1 \to \cdots$.

A cycle of a periodic algorithm is one repetition of the switching sequence $q_1 \to q_2 \to \cdots \to q_{M-1} \to q_M \to q_1$. Note that as the residence times are unique and associated with the modes, the total time of a cycle is constant and given by $T_{cycle} = \sum_{\hat{\imath}=1}^{M} T_{(\hat{\imath})}$. Sampling the system every M switching instants, that is at instants τ_{hM} with $h \in \mathbb{N}$, the transition matrix of the entire cycle of the system can be obtained as:

$$\begin{aligned}\mathbf{\Psi}_{cycle} &:= \mathbf{\Psi}_{(h-1)M}^{hM} = \prod_{i=1}^{M} \mathbf{\Phi}_{hM+1-i} = \prod_{i=1}^{M} e^{\mathbf{G}(\boldsymbol{\alpha}_{hM+1-i})T_{hM+1-i}} \\ &= e^{\mathbf{G}_M T_{(M)}} e^{\mathbf{G}_{M-1} T_{(M-1)}} \cdots \; e^{\mathbf{G}_2 T_{(2)}} e^{\mathbf{G}_1 T_{(1)}}\end{aligned}$$

The implicit discrete time system defined when sampling at switching instants τ_{hM}, with $h \in \mathbb{N}$, is then linear:

$$\mathbf{e}(\tau_{(h+1)M}) = \mathbf{\Psi}_{cycle} \mathbf{e}(\tau_{hM}) \tag{5.41}$$

Considering Lemma 3.1, the stability of the discrete time system (5.41) proves that the AAN with the periodic algorithm leads to consensus. This can be verified considering a discrete time Lyapunov matrix $\mathbf{P} > 0$ as in Theorem 3.2, or simply by calculating the eigenvalues of $\mathbf{\Psi}_{cycle}$.

Clearly, the stability analysis could also be done by considering other switching instant sequences like in Theorem 3.3. Using the whole cycle, however, gives some hints on the convergence rate at which consensus is achieved. For this, let us define

$$\begin{aligned}\overline{\sigma} &:= \max\left\{\text{svd}\left\{\mathbf{\Psi}_{cycle}\right\}\right\} \\ \underline{\sigma} &:= \min\left\{\text{svd}\left\{\mathbf{\Psi}_{cycle}\right\}\right\},\end{aligned}$$

then,

$$\begin{aligned}&\underline{\sigma}^2 \mathbf{I} \leqslant \mathbf{\Psi}_{cycle}{}' \mathbf{\Psi}_{cycle} \leqslant \overline{\sigma}^2 \mathbf{I} \\ \iff &\underline{\sigma}^2 \mathbf{e}'(\tau_{hM}) \mathbf{e}(\tau_{hM}) \leqslant \mathbf{e}'(\tau_{hM}) \mathbf{\Psi}_{cycle}{}' \mathbf{\Psi}_{cycle} \mathbf{e}(\tau_{hM}) \leqslant \overline{\sigma}^2 \mathbf{e}'(\tau_{hM}) \mathbf{e}(\tau_{hM}) \\ \iff &\underline{\sigma} \|\mathbf{e}(\tau_{hM})\| \leqslant \|\mathbf{e}(\tau_{(h+1)M})\| \leqslant \overline{\sigma} \|\mathbf{e}(\tau_{hM})\|\end{aligned}$$

Assuming that the consensus error approaches the origin, from the previous expression a bound for the convergence rate of the norm of the error, similar to that in equation (5.10) for the case of loopless algorithms, can be obtained. Furthermore, not only a higher bound can be defined but also a lower one. By subtracting $\|\mathbf{e}(\tau_{hM})\|$ and dividing by $T_{cycle} > 0$ one obtains,

$$
\begin{aligned}
-\frac{1-\underline{\sigma}}{T_{cycle}}\|\mathbf{e}(\tau_{hM})\| \leqslant \frac{\|\mathbf{e}(\tau_{(h+1)M})\| - \|\mathbf{e}(\tau_{hM})\|}{T_{cycle}} \leqslant -\frac{1-\overline{\sigma}}{T_{cycle}}\|\mathbf{e}(\tau_{hM})\| \\
-\underline{\varsigma}\|\mathbf{e}(\tau_{hM})\| \leqslant \frac{\|\mathbf{e}(\tau_{(h+1)M})\| - \|\mathbf{e}(\tau_{hM})\|}{T_{cycle}} \leqslant -\overline{\varsigma}\|\mathbf{e}(\tau_{hM})\|,
\end{aligned}
\tag{5.42}
$$

where $\underline{\varsigma} := \frac{1-\underline{\sigma}}{T_{cycle}} > 0$ and $\overline{\varsigma} := \frac{1-\overline{\sigma}}{T_{cycle}} > 0$ represent, respectively, approximated lower and higher bounds for the convergence rate of the switched system in a similar sense that ς in section 5.1.1. Note that because of the negative sign in (5.42), $\underline{\varsigma} \geqslant \overline{\varsigma} \geqslant 0$.

It is important to notice that this bounds represent merely a long term mean approximation of the convergence rate at which the network can reach consensus. There is not guarantee that, at any given time, the convergence rate of the consensus error would stay within the limits. This can be easily understood by considering a two modes algorithm, where in one mode, the feedback matrix makes the norm of the error to increase, but this is compensated by the other mode where the norm decreases. If the convergence rate is instantaneously evaluated during the increasing mode, then it would be clearly out of the bounds. Furthermore, in the second mode, the convergence rate would also be out of the limits as the norm decreases faster to compensate the increase in the first mode. The bounds are however calculated considering both behaviors and therefore are average values for the overall convergence rate.

Probabilistic Algorithms

A periodic algorithm is based on a deterministic switching sequence and therefore it is a very special kind of consensus controller which is praised for its simplicity. In fact, a more complex kind of algorithms can be defined by allowing more switching sequences. The most extreme case would be to allow every $M(M-1)$ possible jumps between the discrete modes, associating with each one of them a non unitary probability as defined in Section 3.2.3. Several categories of "mixed" (deterministic and probabilistic) algorithms could be defined in between, by imposing unitary probability of some jumps like in Example 3.3.

Definition 5.2.7. A *probabilistic algorithm* is a switching algorithm that associates with each of the M discrete modes $q_{\hat{\imath}} \in Q = \{q_1, q_2, \ldots, q_M\}$, a feedback matrix $\mathbf{L}_{\hat{\imath}} \in \mathbb{R}^{Nq \times Nq}$; a unique residence time $T_{(\hat{\imath})} := T_k = \tau_{k+1} - \tau_k \in \mathbb{R}^+$, for all $k \in \mathbb{N}$ where the algorithm is switched to mode $q_{\hat{\imath}}$; and a probability vector $\boldsymbol{\pi}_{\hat{\imath}}^+ \in \Lambda_M \subset \mathbb{R}^N$ in such a way that the probability of switching from mode $q_{\hat{\imath}} \in Q$ to mode $q_{\hat{\jmath}} \in Q$ at instant τ_{k+1} is $0 \leqslant \left[\boldsymbol{\pi}_{\hat{\imath}}^+\right]_{\hat{\jmath}} \leqslant 1$.

Note that it is not necessary to relax the uniqueness property of the residence time as, when several known residence times are associated with the same feedback matrix and probability

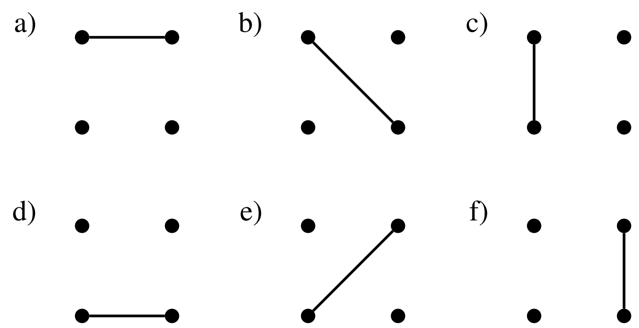


Figure 5.16.: Non Connected Graphs for Switched Algorithms in Example 5.8.

vector, a new discrete mode can be defined for each of the different values. This would, of course, increase the number of discrete modes and increase the complexity of the allowed switching sequences.

When $\mathbf{R}_A = \mathbf{0}$, consensus in an AAN under such an algorithm can be studied with the help of Theorem 3.6. Unlikely the periodic case, it is however difficult to state approximated bounds for the convergence rate of the system as no cycle matrix can be defined.

Example 5.8. Consider $N = 4$ identical systems described by

$$\mathbf{A}_i = \begin{bmatrix} 0.001 & 0.000 \\ 0.000 & 0.002 \end{bmatrix}, \quad \mathbf{B}_i = \begin{bmatrix} 1.000 & 0.500 \\ 0.002 & 1.000 \end{bmatrix}, \quad \mathbf{C}_i = \begin{bmatrix} 1.000 & 0.000 \\ 0.000 & 1.000 \end{bmatrix}.$$

For analysis, we consider an organization matrix derived from a star graph centered in the first agent: $\mathbf{T} = D'(\mathscr{T}^o) = \text{row}\{\mathbf{1}, \ -\mathbf{I}\}$. Note that, as $\mathbf{C}_i = \mathbf{I}$ and the systems are identical, the residual matrix $\mathbf{R}_A$ is identically zero.

Assume that the agents can only handle communication signals with one unique other agent at the same time. Therefore, switched algorithms are proposed based on the six graphs ($\mathscr{G}_a$, $\mathscr{G}_b$, $\mathscr{G}_c$, $\mathscr{G}_d$, $\mathscr{G}_e$, and $\mathscr{G}_f$) of Figure 5.16 and their respective feedback matrices:

$$\begin{aligned} &\mathbf{L}_a = -\hat{L}(\mathscr{G}_a), \mathbf{L}_b = -\hat{L}(\mathscr{G}_b), \mathbf{L}_c = -\hat{L}(\mathscr{G}_c), \\ &\mathbf{L}_d = -\hat{L}(\mathscr{G}_d), \mathbf{L}_e = -\hat{L}(\mathscr{G}_e), \mathbf{L}_f = -\hat{L}(\mathscr{G}_f). \end{aligned}$$

These matrices define six discrete modes of operation $Q = \{a, b, c, d, e, f\}$.

First, consider a periodic algorithm with identical residence times for each mode, $T_{(\hat{\imath})} = 0.2$, and the infinitely repeated sequence $\cdots \to a \to b \to c \to d \to e \to f \to a \to \cdots$. In this case, the whole cycle matrix $\mathbf{\Psi}_{cycle}$ of the system is such that

$$\text{eig}\left\{\mathbf{\Psi}_{cycle}\right\} = \{0.3425 \pm 0.0435i, 0.3582, 0.5784 \pm 0.0190i, 0.5817\}.$$

Clearly, this algorithm allows the network to reach consensus as the eigenvalues of the cycle matrix are all within the unitary disc. This can be verified by the simulation shown in Figure

5.17 a.1) and a.2) where, respectively, the first and second output of the systems are drawn for an arbitrary initial condition.

Now consider that the network stays in each modes $T_{(\hat{\imath})} = 0.2$ time units before randomly switching to other mode. The switching probabilities from each mode are given by:

$$\boldsymbol{\pi}_a^+ = \begin{bmatrix} 0.0000 \\ 0.2375 \\ 0.2375 \\ 0.0500 \\ 0.2375 \\ 0.2375 \end{bmatrix}, \boldsymbol{\pi}_b^+ = \begin{bmatrix} 0.2375 \\ 0.0000 \\ 0.2375 \\ 0.2375 \\ 0.0500 \\ 0.2375 \end{bmatrix}, \boldsymbol{\pi}_c^+ = \begin{bmatrix} 0.2375 \\ 0.2375 \\ 0.0000 \\ 0.2375 \\ 0.2375 \\ 0.0500 \end{bmatrix},$$

$$\boldsymbol{\pi}_d^+ = \begin{bmatrix} 0.0500 \\ 0.2375 \\ 0.2375 \\ 0.0000 \\ 0.2375 \\ 0.2375 \end{bmatrix}, \boldsymbol{\pi}_e^+ = \begin{bmatrix} 0.2375 \\ 0.0500 \\ 0.2375 \\ 0.2375 \\ 0.0000 \\ 0.2375 \end{bmatrix}, \boldsymbol{\pi}_f^+ = \begin{bmatrix} 0.2375 \\ 0.2375 \\ 0.0500 \\ 0.2375 \\ 0.2375 \\ 0.0000 \end{bmatrix}.$$

Applying Theorem (3.6), the following six conditions for stability can be stated:

$$\begin{aligned}
&\boldsymbol{\Phi}_a'\mathbf{P}(\boldsymbol{\pi}_a^+)\boldsymbol{\Phi}_a - \mathbf{P}_1 < 0,\\
&\boldsymbol{\Phi}_b'\mathbf{P}(\boldsymbol{\pi}_b^+)\boldsymbol{\Phi}_b - \mathbf{P}_2 < 0,\\
&\boldsymbol{\Phi}_c'\mathbf{P}(\boldsymbol{\pi}_c^+)\boldsymbol{\Phi}_c - \mathbf{P}_3 < 0,\\
&\boldsymbol{\Phi}_d'\mathbf{P}(\boldsymbol{\pi}_d^+)\boldsymbol{\Phi}_d - \mathbf{P}_4 < 0,\\
&\boldsymbol{\Phi}_e'\mathbf{P}(\boldsymbol{\pi}_e^+)\boldsymbol{\Phi}_e - \mathbf{P}_5 < 0,\\
&\boldsymbol{\Phi}_f'\mathbf{P}(\boldsymbol{\pi}_f^+)\boldsymbol{\Phi}_f - \mathbf{P}_6 < 0.
\end{aligned}$$

The monomials of the polynomial ,

$$\mathbf{P}(\boldsymbol{\alpha}) = \mathbf{P}_1\alpha_1 + \mathbf{P}_2\alpha_2 + \mathbf{P}_3\alpha_3 + \mathbf{P}_4\alpha_4 + \mathbf{P}_5\alpha_5 + \mathbf{P}_6\alpha_6 > 0,$$

are all positive definite matrices, and the transition matrices are defined as

$$\boldsymbol{\Phi}_{\hat{\imath}} = e^{\mathbf{TC}(\mathbf{BL}_{\hat{\imath}} + \mathbf{AC}^{-1})\mathbf{T}^+ T_{(\hat{\imath})}},$$

for all $\hat{\imath} \in \{a,b,c,d,e,f\}$. It can be numerically shown that the described conditions are feasible and therefore the network with this switching rule reaches consensus. For some randomly generated switching sequence, the evolution of the outputs of the systems with the probabilistic switched algorithm is shown for identical initial conditions as before in Figure 5.17 b.1) and b.2).

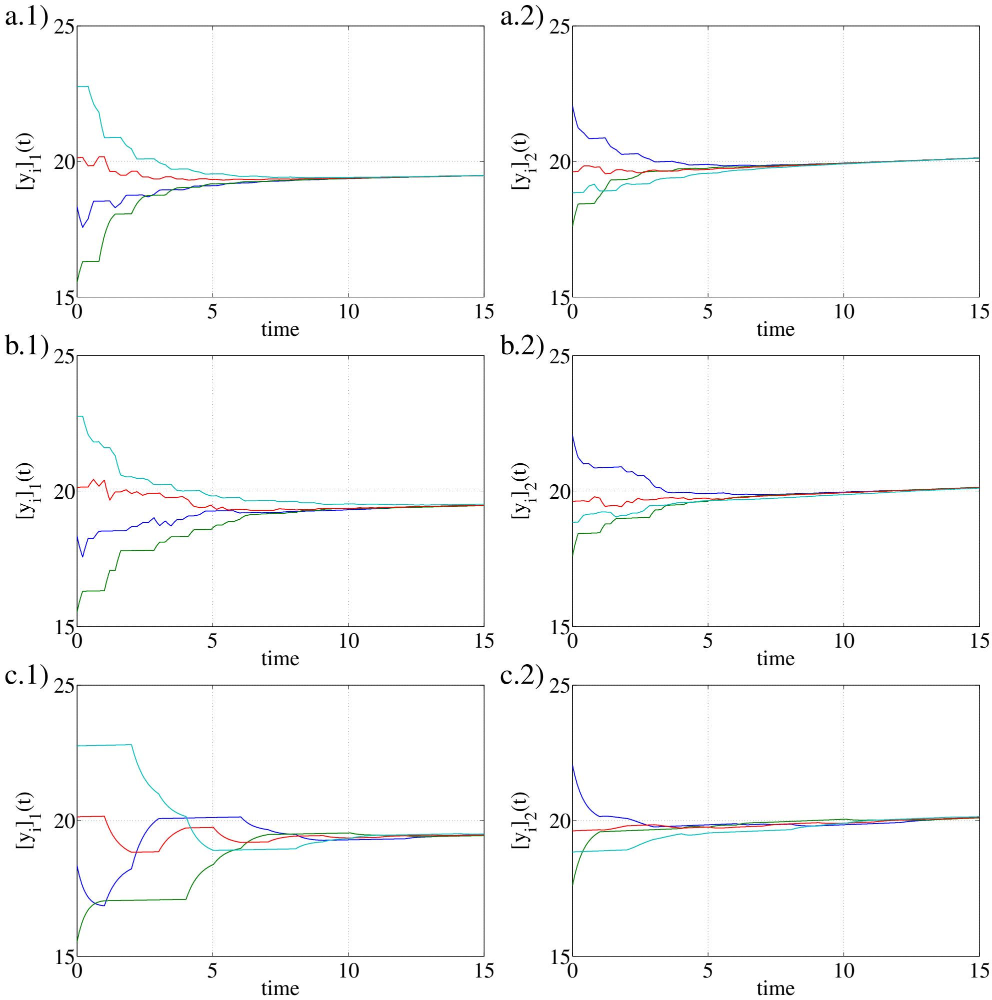


Figure 5.17.: First and Second Outputs Evolution of Agents in Example 5.8 with a) Periodic Switching Algorithm, b) Probabilistic Switching Algorithm, and c) Slower Periodic Switching Algorithm.

The convergence rate bounds for the periodic algorithm are given by $\overline{\varsigma} = 0.3243$ and $\underline{\varsigma} = 0.5963$. If however the same algorithm is implemented with $T_{(\hat{\imath})} = 1.0$, then the bounds become $\overline{\varsigma} = 0.0691$ and $\underline{\varsigma} = 0.1667$. This shows that by changing the residence time in the modes, the convergence rate at which consensus is reached can be changed. In this case, increasing the residence time makes the algorithm considerably slower as both bounds are closer to zero. The time response of the slower algorithm can be seen in Figure 5.17 c). Unfortunately for probabilistic algorithms, although it can be verified by simulation that changing the residence times will modify the convergence rate at which consensus is reached, it is not easy to estimate a priori this change like in the periodic case. ■

Unintended Switching

We understand as unintended switching the case where the feedback matrix switches between M variations of a known ideal or original feedback matrix (derived, for example, from a loopless weighted graph). Typically, only a few of these possible matrices allows the network to reach consensus while the majority only enforces consensus in some sections of the network.

Definition 5.2.8. An *unintended switching process* is such that it associates with each of the M discrete modes $q_{\hat{\imath}} \in Q = \{q_1, q_2, \ldots, q_M\}$, a feedback matrix $\mathbf{L}_{\hat{\imath}} \in \mathbb{R}^{Nq \times Nq}$; a non precisely known residence time $T_{(\hat{\imath})} := T_{(\hat{\imath})}^{min} + \Delta T_{(\hat{\imath})} := \tau_{k+1} - \tau_k \in \mathbb{R}^+$, for all $k \in \mathbb{N}$ where the algorithm is switched to mode $q_{\hat{\imath}}$, such that $T_{(\hat{\imath})}^{min} \in \mathbb{R}^+$ is known and $\Delta T_{(\hat{\imath})} \in [0, \Delta T_{(\hat{\imath})}^{max}]$; and a probability vector $\boldsymbol{\pi}_{\hat{\imath}}^+ \in \Lambda_M \subset \mathbb{R}^N$ in such a way that the probability of switching from mode $q_{\hat{\imath}} \in Q$ to mode $q_{\hat{\jmath}} \in Q$ at instant τ_{k+1} is $0 \leqslant [\boldsymbol{\pi}_{\hat{\imath}}^+]_{\hat{\jmath}} \leqslant 1$.

This kind of switching may occur as the result of communication or controller failure. In these cases, the ideal feedback matrix is instantaneously changed by a similar one selected from a vast number of possibilities. Given an ideal feedback matrix, the number of variations that can be considered as a discrete mode increase explosively with the number of agents, of edges of the original graph, and complexity of the accepted weight function for the edges. Furthermore, the probabilities of jumping from any of these modes to any other needs to be well estimated in order to describe the unintended switching process accurately.

Example 5.9. To illustrate the idea of explosion of modes, let us consider $N = 2$ agents with an algorithm derived of the unweighted undirected graph that connects both agents. In the ideal case, the consensus algorithm is described by matrix $\mathbf{L}_{01} \in \mathbb{R}^{Nq \times Nq}$ below. Note that this is a matrix with $N^2 = 4$ blocks. The diagonal ones represent feedback signals within an agent, while the off diagonal blocks represent signals that are exchanged between the agents. A simple failure of one of this blocks changes its value from $\pm\mathbf{I}$ to $\mathbf{0}$.

$$\mathbf{L}_{01} = \begin{bmatrix} -\mathbf{I} & \mathbf{I} \\ \mathbf{I} & -\mathbf{I} \end{bmatrix},$$

$$\mathbf{L}_{02} = \begin{bmatrix} -\mathbf{I} & \mathbf{0} \\ \mathbf{I} & -\mathbf{I} \end{bmatrix}, \mathbf{L}_{03} = \begin{bmatrix} -\mathbf{I} & \mathbf{I} \\ \mathbf{0} & -\mathbf{I} \end{bmatrix}, \mathbf{L}_{04} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{I} & -\mathbf{I} \end{bmatrix}, \mathbf{L}_{05} = \begin{bmatrix} -\mathbf{I} & \mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix},$$

$$\mathbf{L}_{06} = \begin{bmatrix} -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \end{bmatrix}, \mathbf{L}_{07} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I} & -\mathbf{I} \end{bmatrix}, \mathbf{L}_{08} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & -\mathbf{I} \end{bmatrix},$$

$$\mathbf{L}_{09} = \begin{bmatrix} -\mathbf{I} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \mathbf{L}_{10} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}, \mathbf{L}_{11} = \begin{bmatrix} -\mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \end{bmatrix},$$

$$\mathbf{L}_{12} = \begin{bmatrix} -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \mathbf{L}_{13} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \mathbf{L}_{14} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}, \mathbf{L}_{15} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \end{bmatrix},$$

$$\mathbf{L}_{16} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

If one block fails, then there are ${}^{N^2}C_1 = 4$ possible feedback matrices[3], $\mathbf{L}_{02}$, $\mathbf{L}_{03}$, $\mathbf{L}_{04}$, and $\mathbf{L}_{05}$, that describe four different scenarios. When two blocks fail simultaneously, there are ${}^{N^2}C_2 = 6$ matrices ($\mathbf{L}_{06}$ to $\mathbf{L}_{11}$). Three simultaneous failures induce ${}^{N^2}C_3 = 4$ new scenarios ($\mathbf{L}_{12}$ to $\mathbf{L}_{15}$) and four failures impose a sixteenth possible feedback matrix $\mathbf{L}_{16}$.

Even for this very simple case, the number of discrete modes that can be defined is considerably large in comparison with the number of agents. More sophisticated examples can also be named by considering weighted graphs where each edge is associated with a larger set of possible weight matrices. An interesting case would be that where the communication of only some of the signals between two agents is interrupted and not all of them. In that case, only some elements of the weights function would be switched to zero. By adding more agents and considering different ideal algorithms, the number of possible discrete modes can be easily incremented. In order to have a realistic description of a switching failure situation, the probability of jumping from any of these modes to any other must be correctly estimated along with the residence time at each mode.

[3] The symbol ${}^nC_k = \binom{n}{k} = n!/(k!(n-k)!)$ denotes Newton's binomial coefficient: "choose k out of n".

Furthermore, note that these matrices are not always easy to describe through graphs and therefore it is difficult to study them from that perspective. Also, only some of the matrices have the zero sum row property so it does not always hold that $\mathbf{L}_{\hat{\imath}}\mathbf{J} = \mathbf{0}$. ■

In the case of intended switching we have restricted the case to feedback matrices derived of loopless graphs so that $\mathbf{L}_{\hat{\imath}}\mathbf{J} = \mathbf{0}$ for all $\hat{\imath} \in Q$. This restriction is very convenient to make the residual matrix independent of the feedback matrices and therefore to avoid the influence of the states of the agents. However, this condition is not necessarily fulfilled in the case where the switching between feedback matrices occurs in an unintended way. In general, the dynamics of the error can be written as the following switched system

$$\begin{aligned}\dot{\mathbf{e}} &= \mathbf{TC}\left(\mathbf{BL}_{\hat{\imath}} + \mathbf{AC}^{+}\right)\mathbf{T}^{+}\mathbf{e} + \mathbf{TC}\left(\mathbf{BL}_{\hat{\imath}}\mathbf{JC} + \mathbf{A}\left(\mathbf{C}^{+}\mathbf{JC} + \mathbf{C}^{\perp}(\mathbf{C}^{\perp})'\right)\right)\mathbf{x} \\ &= \mathbf{G}_{\hat{\imath}}\mathbf{e} + \mathbf{R}_{\hat{\imath}}\mathbf{x}.\end{aligned} \tag{5.43}$$

Even when $\mathbf{R}_A = \mathbf{0}$, the states of the systems might influence the value of the consensus error. Therefore, to study consensus under this kind of switching, we must assume the existence of a security mechanism that ensures that $\|\mathbf{R}_{\hat{\imath}}\mathbf{x}\| < \varepsilon(t)$. In other words, a mechanism which ensures that the effect of the states over the consensus error can be neglected at any time.

This can be achieved from a supervisory perspective quite simply for networks where $\mathbf{R}_A = \mathbf{0}$, by forcing the feedback matrix to switch to a safe mode $\hat{\imath}$ where $\mathbf{L}_{\hat{\imath}}\mathbf{J} = \mathbf{0}$, if the system had stayed too long in a non safe mode $\hat{\jmath}$ where $\mathbf{L}_{\hat{\jmath}}\mathbf{J} \neq \mathbf{0}$. For example, in Example 5.9, if the communication line from the second agent to the first agent fails during a period longer than tolerated, that is, if the system stays at $\mathbf{L}_{02}$ for a long time, then the controller forces the feedback matrix to switch to the safe mode $\mathbf{L}_{07}$ until the failure is repaired and only then it switches back to $\mathbf{L}_{01}$. This assumption will make the description of the unintended switching process even more intricate that it would be without it.

In general, stability conditions for this case can be formulated with the help of Theorem 3.7. The main complication is to describe accurately the switching process due to the large number of discrete modes expected and the uncertainty on the residence time at each mode. Unfortunately, the stability conditions that can be formulated depend on the assumption that $\|\mathbf{R}_{\hat{\imath}}\mathbf{x}\| < \varepsilon(t)$ and therefore they cannot be regarded as a formal proof that consensus can be reached but merely show that it can be reasonably approximated.

Example 5.10. Consider $N = 3$ agents with dynamics described by:

$$\begin{aligned}\mathbf{A}_1 &= 0.02,\ \mathbf{B}_1 = 1.00,\ \mathbf{C}_1 = 0.20,\\ \mathbf{A}_2 &= 0.02,\ \mathbf{B}_2 = 1.50,\ \mathbf{C}_2 = 0.20,\\ \mathbf{A}_3 &= 0.02,\ \mathbf{B}_3 = 0.50,\ \mathbf{C}_3 = 0.20.\end{aligned}$$

Note that $\mathbf{R}_A = \mathbf{TCA}(\mathbf{C}^{+}\mathbf{JC} + \mathbf{C}^{\perp}(\mathbf{C}^{\perp})') = \mathbf{0}$, but the agents are not identical. The agents search consensus through a nominal algorithm described as the negative Laplacian of the unweighted undirected graph with edges between the first and the second agents and the first and the third agent. That is, matrix $\mathbf{L}_{01}$ below. To study consensus the organization matrix $\mathbf{T} = \text{row}\{\mathbf{1}, -\mathbf{I}\}$ will be considered.

First consider the case where failures may occur only in the used communication channels between the agents and not in the feedback from one agent to itself. These signals correspond to the non diagonal entries of the feedback matrix. Then, there are only four possible failures in the system: either $[\mathbf{L}_{01}]_{12}$, $[\mathbf{L}_{01}]_{13}$, $[\mathbf{L}_{01}]_{21}$ or $[\mathbf{L}_{01}]_{23}$ switch from 1 to 0. Furthermore, the agents are capable of detecting a fault almost immediately and change their individual feedback gain (the diagonal numbers in the algorithm matrix) to always match the zero sum row property. Therefore, it can be considered that every time a failure occurs, the system switches instantaneously to an operation mode $\hat{\imath}$ where $\mathbf{L}_{\hat{\imath}}\mathbf{J} = \mathbf{0}$. Considering that up to four successive failures may occur, this situation is described by $M = 16$ discrete operation modes in $Q = \{01, 02, ..., 15, 16\}$, each one associated with one of the sixteen matrices listed below.

Ideal Algorithm:

$$\mathbf{L}_{01} = \begin{bmatrix} -2 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix},$$

One Failure:

$$\mathbf{L}_{02} = \begin{bmatrix} -1 & 0 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix}, \mathbf{L}_{03} = \begin{bmatrix} -1 & 1 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix},$$

$$\mathbf{L}_{04} = \begin{bmatrix} -2 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}, \mathbf{L}_{05} = \begin{bmatrix} -2 & 1 & 1 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

Two Failures:

$$\mathbf{L}_{06} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix}, \mathbf{L}_{07} = \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}, \mathbf{L}_{08} = \begin{bmatrix} -1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{L}_{09} = \begin{bmatrix} -1 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}, \mathbf{L}_{10} = \begin{bmatrix} -1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{L}_{11} = \begin{bmatrix} -2 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

Three Failures:

$$\mathbf{L}_{12} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}, \mathbf{L}_{13} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{L}_{14} = \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{L}_{15} = \begin{bmatrix} -1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

Four Failures:

$$\mathbf{L}_{16} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

If a failure occurs, and so the system switches to a mode $\hat{\imath} \neq 01$, then the probability that the failure is solved after an uncertain residence time $T_{(\hat{\imath})} \in [T_{(\hat{\imath})}^{min}, T_{(\hat{\imath})}^{min} + \Delta T_{(\hat{\imath})}^{max}]$ is 50%. The probability that after this period, another failure occurs is also 50%. In principle, all four different failures might occur with the same probability. However the probabilities that, at a switching instant, a failure is repaired and another different failure occurs, two different failures occur, or two different failures are repaired, are considered to be neglectable. This means that from a given mode $\hat{\imath} \in Q$, some other modes are not reacheable in one jump. This can be seen graphically in Figure 5.18 where a double arrowed edge between modes $\hat{\imath}$ and $\hat{\jmath}$ represents that a jump may occur from $\hat{\imath}$ to $\hat{\jmath}$ and *vice versa*. Jumping downwards in the graph represents the occurance of a failure and jumping upwards, that the failure is repaired. The probabilities of these jumps are stated, for each level of arrows, as a tuple (p_{dw}, p_{up}) at the right side of the figure, where p_{dw} is the probability of jumping downwards in the figure and p_{up} of jumping upwards. For more clarity, the different probabilities are also shown in matrix $\mathbf{\Pi}$ where the future probability vectors, $\boldsymbol{\pi}_{\hat{\imath}}^{+}$, for each mode, are implicitly defined. Element $[\mathbf{\Pi}]_{\hat{\imath}\hat{\jmath}} = \left[\boldsymbol{\pi}_{\hat{\imath}}^{+}\right]_{\hat{\jmath}}$ should be understood as the probability of switching from mode $\hat{\imath}$ to mode $\hat{\jmath}$.

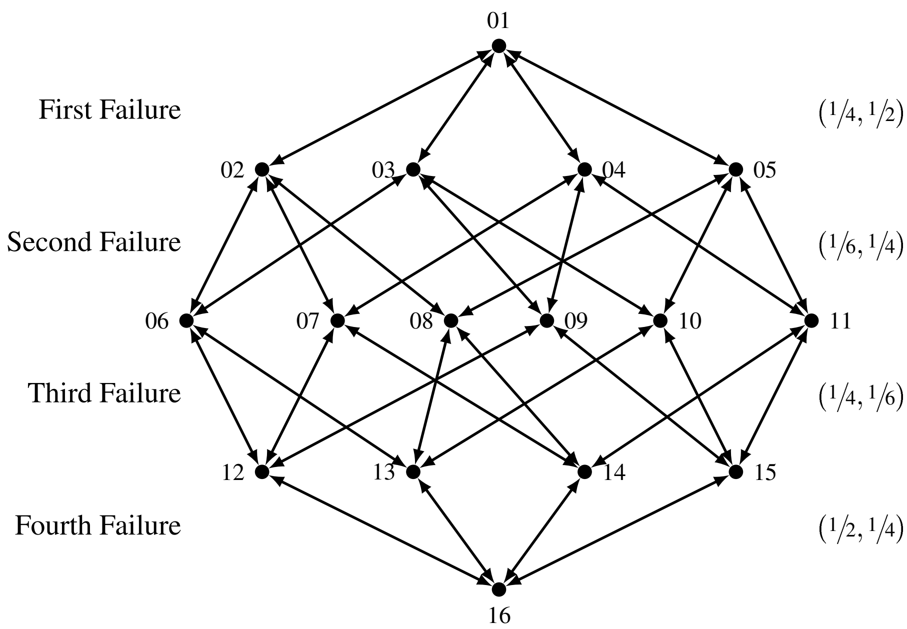


Figure 5.18.: Switching Graph for System in Example 5.10.

$$
\begin{aligned}
\boldsymbol{\Pi} &= \operatorname{col}\left\{(\boldsymbol{\pi}_{\hat{\imath}}^{+})'\right\}_{\hat{\imath}\in Q} \\
&= \begin{bmatrix}
0 & 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/2 & 0 & 0 & 0 & 0 & 1/6 & 1/6 & 1/6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/2 & 0 & 0 & 0 & 0 & 1/6 & 0 & 0 & 1/6 & 1/6 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/2 & 0 & 0 & 0 & 0 & 0 & 1/6 & 0 & 1/6 & 0 & 1/6 & 0 & 0 & 0 & 0 & 0 \\
1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 1/6 & 0 & 1/6 & 1/6 & 0 & 0 & 0 & 0 & 0 \\
0 & 1/4 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 1/4 & 0 & 0 & 0 \\
0 & 1/4 & 0 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 1/4 & 0 & 0 \\
0 & 1/4 & 0 & 0 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 1/4 & 0 & 0 \\
0 & 0 & 1/4 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 & 1/4 & 0 \\
0 & 0 & 1/4 & 0 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 1/4 & 0 \\
0 & 0 & 0 & 1/4 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 1/4 & 0 \\
0 & 0 & 0 & 0 & 0 & 1/6 & 1/6 & 0 & 1/6 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 \\
0 & 0 & 0 & 0 & 0 & 1/6 & 0 & 1/6 & 0 & 1/6 & 0 & 0 & 0 & 0 & 0 & 1/2 \\
0 & 0 & 0 & 0 & 0 & 0 & 1/6 & 1/6 & 0 & 0 & 1/6 & 0 & 0 & 0 & 0 & 1/2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/6 & 1/6 & 1/6 & 0 & 0 & 0 & 0 & 1/2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 & 0
\end{bmatrix}.
\end{aligned}
$$

As for the residence times, we consider that for all $\hat{\imath} \in Q \backslash \{01\}$ these are described by identical bounds: $T_{(\hat{\imath})}^{min} = 1.0$ and $\Delta T_{(\hat{\imath})}^{max} = 0.5$. For the nominal mode 01, we consider a fixed residence $T_{(\hat{\imath})} = 1.5$.

With this long specification of the switching process, Theorem 3.7 can be applied to investigate consensus. This leads to one $q(N-1) \times q(N-1) = 2 \times 2$ matrix inequality and 15

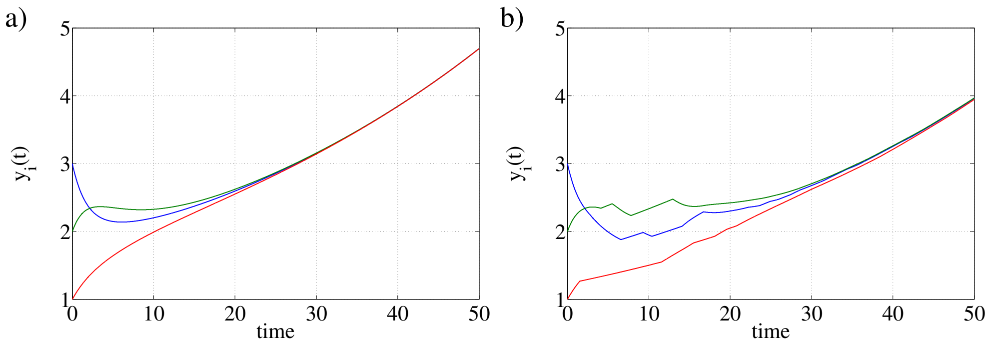


Figure 5.19.: Output evolution of the network in Example 5.10 under a) Ideal non-switching Algorithm, and b) Unintended Switching Algorithm.

matrix inequalities of dimension $3q(N-1) \times 3q(N-1) = 6 \times 6$. This inequalities consider 16 Lyapunov 2×2 matrix variables, 15 $\mathbf{X}$-type 2×2 matrix variables, 15 $\mathbf{Y}$-type 2×2 matrix variables, and 15 η-type scalar variables. These inequalities can be tested numerically for feasibility to verify that consensus can be reached with this switching specification. Figure 5.19 shows the behavior of the network in the ideal case without failures and with a random switching sequence that fulfills the specified properties. ■

Example 5.11. Now consider the same $N = 3$ agents but in another switching scenario. This time, if a failure occurs, the controllers at each agent do not react instantaneously. Therefore, a short gap of time is considered where $\mathbf{L}_{\hat{\imath}}\mathbf{J} \neq \mathbf{0}$ before the algorithm is switched again to a "safe" mode. In principle all four failures may occur. However, for simplicity, we will consider that, given a failure that breaks the connection from agent i to agent j, then the probability that a second failure occurs is neglectable except for the one that breaks the connection from agent j to agent i. A third failure cannot occur. In the case of a second failure, the controllers also need some time to move to a safe algorithm. Only from safe algorithms, the network can jump again to the ideal algorithm. This specification leads to 15 discrete modes associated with the matrices listed below.

Ideal Algorithm:

$$\mathbf{L}_{01} = \begin{bmatrix} -2 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix},$$

One Failure:

$$\mathbf{L}_{02} = \begin{bmatrix} -2 & 0 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix}, \ \mathbf{L}_{03} = \begin{bmatrix} -2 & 1 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix},$$

$$\mathbf{L}_{04} = \begin{bmatrix} -2 & 1 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix}, \ \mathbf{L}_{05} = \begin{bmatrix} -2 & 1 & 1 \\ 1 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix},$$

One Failure - Safe Modes:

$$\mathbf{L}_{06} = \begin{bmatrix} -1 & 0 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix}, \ \mathbf{L}_{07} = \begin{bmatrix} -1 & 1 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix},$$

$$\mathbf{L}_{08} = \begin{bmatrix} -2 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}, \ \mathbf{L}_{09} = \begin{bmatrix} -2 & 1 & 1 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

Two Failures:

$$\mathbf{L}_{10} = \begin{bmatrix} -1 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix}, \ \mathbf{L}_{11} = \begin{bmatrix} -1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix},$$

$$\mathbf{L}_{12} = \begin{bmatrix} -2 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}, \ \mathbf{L}_{13} = \begin{bmatrix} -2 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

Two Failures - Safe Modes:

$$\mathbf{L}_{14} = \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}, \ \mathbf{L}_{15} = \begin{bmatrix} -1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

The possible switching sequences are described graphically in Figure 5.20 and the switch-

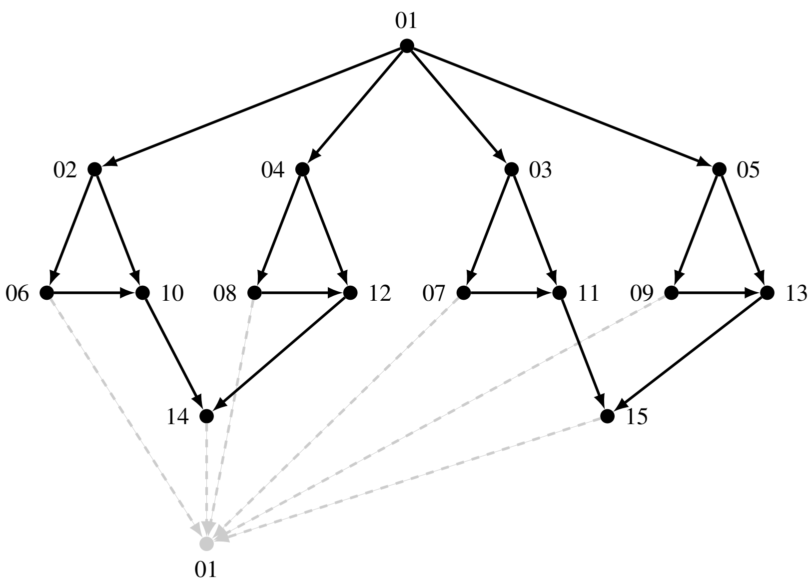


Figure 5.20.: Switching Graph for System in Example 5.11.

ing probabilities are given by the following matrix,

$$\boldsymbol{\Pi} = \operatorname{col}\left\{(\boldsymbol{\pi}_{\hat{\imath}}^{+})'\right\}_{\hat{\imath}\in Q}$$

$$= \begin{bmatrix}
0 & 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 & 0 & 3/4 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 & 0 & 3/4 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 & 0 & 3/4 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 & 0 & 3/4 & 0 & 0\\
1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3/4 & 0 & 0 & 0 & 0 & 0\\
1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3/4 & 0 & 0 & 0 & 0\\
1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3/4 & 0 & 0 & 0\\
1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3/4 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}.$$

Ignoring the switching residual matrix $\mathbf{R}_{\hat{\imath}}$, with a fixed residence time for the ideal mode, $T_{(1)} = 1.5$, and,

$$\begin{aligned}
T_{(\hat{\imath})}^{min} &= 0.10,\ \ \Delta T_{(\hat{\imath})}^{max} = 0.050, \text{ for } \hat{\imath} \in \{2,3,4,5\},\\
T_{(\hat{\imath})}^{min} &= 0.20,\ \ \Delta T_{(\hat{\imath})}^{max} = 0.100, \text{ for } \hat{\imath} \in \{6,7,8,9\},\\
T_{(\hat{\imath})}^{min} &= 0.05,\ \ \Delta T_{(\hat{\imath})}^{max} = 0.025, \text{ for } \hat{\imath} \in \{10,11,12,13\},\\
T_{(\hat{\imath})}^{min} &= 0.30,\ \ \Delta T_{(\hat{\imath})}^{max} = 0.150, \text{ for } \hat{\imath} \in \{14,15\},
\end{aligned}$$

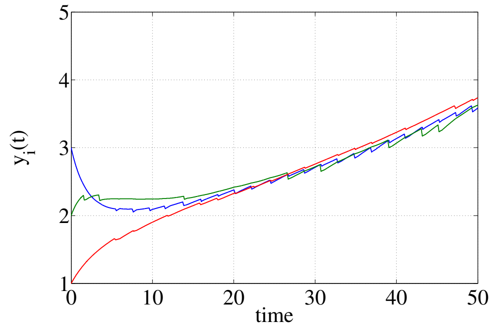


Figure 5.21.: Output evolution of the network in Example 5.11 under Unintended Switching Algorithm.

the inequalities resulting of Theorem 3.7 can be tested feasible. However, due to "unsafe" modes where $\mathbf{L}_{\hat{\imath}}\mathbf{J} \neq \mathbf{0}$, this is not a prove that consensus can be reached. In Figure 5.21 a simulation of the network under a random sequence of jumps with specified probabilities is shown. From here is clear that, at moments, the effect of the unsafe modes drives the system away from the consensus objective although a general tendency to consensus can be observed. This effect can, eventually, move the agents too far away from each other making impossible reaching consensus in the close future. This is more relevant when the state dynamics of the network are unstable. ■

5.3. Generalized Dynamics

In this section, some direct generalizations of the previous discussions are shown. First, each agent is generalized by including a proportional term in the transfer function matrix of each agent. Then the case of higher order dynamics is studied. In all these cases, expressions for the dynamics of the error and the states are derived. Finally, a discussion on the effect of communication dynamics over the consensus error is stated.

5.3.1. Direct Input over Output

Up to now, only dynamical input/output relationships have been considered. That is, systems where $\mathbf{y}_i = \mathbf{C}_i\mathbf{x}_i$. Consider now a different kind of agent defined as:

Definition 5.3.1. An autonomous *agent with direct output* (DA) is an autonomous agent $i \in \mathscr{V}$ with individual dynamics described by:

$$\begin{aligned} \dot{\mathbf{x}}_i &= \mathbf{A}_i\mathbf{x}_i + \mathbf{B}_i\mathbf{u}_i \\ \mathbf{y}_i &= \mathbf{C}_{ii}\mathbf{x}_i + \mathbf{D}_i\mathbf{u}_i \end{aligned} \tag{5.44}$$

With $\mathbf{A}_i \in \mathbb{R}^{n_i\times n_i}$, $\mathbf{B}_i \in \mathbb{R}^{n_i\times q}$, row full rank $\mathbf{C}_i \in \mathbb{R}^{q\times n_i}$, and a matrix $\mathbf{D}_i \in \mathbb{R}^{q\times q}$ that models a direct relationship between the inputs and the outputs of the system.

Note that we consider here that the number of input of each system is the same as the number of outputs. A network composed only of this kind of agents will be referred to as a Direct Agents Network.

Definition 5.3.2. A *Direct Agents Network* (DAN) is a network composed only of DA with dynamics described by:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \end{aligned} \tag{5.45}$$

Where, $\mathbf{A} = \operatorname{diag}\{\mathbf{A}_i\}_{i=1}^N$, $\mathbf{B} = \operatorname{diag}\{\mathbf{B}_i\}_{i=1}^N$, $\mathbf{C} = \operatorname{diag}\{\mathbf{C}_i\}_{i=1}^N$ and $\mathbf{D} = \operatorname{diag}\{\mathbf{D}_i\}_{i=1}^N$.

The consensus error described through an organization matrix $\mathbf{T} = D'(\mathscr{T}^o)$ with a directed tree $\mathscr{T}^o$ is:

$$\mathbf{e} = \mathbf{T}\mathbf{y} \implies \mathbf{y} = \mathbf{T}^+\mathbf{e} + \mathbf{J}\mathbf{y}$$

A consensus algorithm derived from an undirected graph without selfloops $\mathbf{L} = -\hat{L}(\mathscr{G}_w)$ is considered so that:

$$\mathbf{u} = \mathbf{L}\mathbf{y} \implies \dot{\mathbf{u}} = \mathbf{L}\dot{\mathbf{y}} \tag{5.46}$$

$$\mathbf{u} = \mathbf{L}\left(\mathbf{T}^+\mathbf{e} + \mathbf{J}\mathbf{y}\right) = \mathbf{L}\mathbf{T}^+\mathbf{e} \implies \dot{\mathbf{u}} = \mathbf{L}\mathbf{T}^+\dot{\mathbf{e}} \tag{5.47}$$

The dynamics of the error are given by:

$$\dot{\mathbf{e}} = \mathbf{TCAx} + \mathbf{TCBLT}^{+}\mathbf{e} + \mathbf{TD}\dot{\mathbf{u}}$$

Using (5.46),

$$\begin{aligned} \dot{\mathbf{u}} &= \mathbf{L}\dot{\mathbf{y}} = \mathbf{L}\left(\mathbf{C}\dot{\mathbf{x}} + \mathbf{D}\dot{\mathbf{u}}\right) \\ \left(\mathbf{I} - \mathbf{LD}\right)\dot{\mathbf{u}} &= \mathbf{LCAx} + \mathbf{LCBLT}^{+}\mathbf{e} \\ \dot{\mathbf{u}} &= \left(\mathbf{I} - \mathbf{LD}\right)^{-1}\mathbf{LC}\left(\mathbf{Ax} + \mathbf{BLT}^{+}\mathbf{e}\right), \end{aligned}$$

assuming that $\left(\mathbf{I} - \mathbf{LD}\right)^{-1}$ exists. So,

$$\begin{aligned} \dot{\mathbf{e}} &= \mathbf{TCAx} + \mathbf{TCBLT}^{+}\mathbf{e} + \mathbf{TD}\left(\mathbf{I} - \mathbf{LD}\right)^{-1}\mathbf{LC}\left(\mathbf{Ax} + \mathbf{BLT}^{+}\mathbf{e}\right) \\ \dot{\mathbf{e}} &= \mathbf{T}\left[\mathbf{I} + \mathbf{D}\left(\mathbf{I} - \mathbf{LD}\right)^{-1}\mathbf{L}\right]\mathbf{C}\left[\mathbf{Ax} + \mathbf{BLT}^{+}\mathbf{e}\right]. \end{aligned}$$

Note that $\mathbf{I} + \mathbf{D}\left(\mathbf{I} - \mathbf{LD}\right)^{-1}\mathbf{L} = \left(\mathbf{I} - \mathbf{DL}\right)^{-1}$ (see Proposition A.10 in the Appendix), and so:

$$\dot{\mathbf{e}} = \mathbf{T}\left(\mathbf{I} - \mathbf{DL}\right)^{-1}\mathbf{CBLT}^{+}\mathbf{e} + \mathbf{T}\left(\mathbf{I} - \mathbf{DL}\right)^{-1}\mathbf{CAx}. \tag{5.48}$$

Considering the right pseudoinverse $\mathbf{C}^{+} \in \mathbb{R}^{n \times Nq}$, so that $\mathbf{CC}^{+} = \mathbf{I}$ and $\mathbf{C}^{+}\mathbf{C} = \mathbf{I} - \mathbf{C}^{\perp}(\mathbf{C}^{\perp})'$, with $\mathbf{CC}^{\perp} = \mathbf{0}$ and $(\mathbf{C}^{\perp})'\mathbf{C}^{\perp} = \mathbf{I}$, then

$$\begin{aligned} \mathbf{y} = \mathbf{Cx} + \mathbf{Du} &\implies \mathbf{C}^{+}\mathbf{y} = \mathbf{x} - \mathbf{C}^{\perp}(\mathbf{C}^{\perp})'\mathbf{x} + \mathbf{C}^{+}\mathbf{Du} \\ &\implies \mathbf{x} = \mathbf{C}^{+}\left(\mathbf{T}^{+}\mathbf{e} + \mathbf{Jy}\right) + \mathbf{C}^{\perp}(\mathbf{C}^{\perp})'\mathbf{x} - \mathbf{C}^{+}\mathbf{DLT}^{+}\mathbf{e} \\ &\implies \mathbf{x} = \mathbf{C}^{+}\left(\mathbf{I} - \mathbf{DL} + \mathbf{JDL}\right)\mathbf{T}^{+}\mathbf{e} + \left(\mathbf{C}^{+}\mathbf{JC} + \mathbf{C}^{\perp}(\mathbf{C}^{\perp})'\right)\mathbf{x}. \end{aligned}$$

Therefore, (5.48) becomes:

$$\begin{aligned} \dot{\mathbf{e}} =& \mathbf{T}\left(\mathbf{I} - \mathbf{DL}\right)^{-1}\mathbf{C}\left(\mathbf{BL} + \mathbf{AC}^{+}\left(\mathbf{I} - \mathbf{DL} + \mathbf{JDL}\right)\right)\mathbf{T}^{+}\mathbf{e} + \ldots \\ & \ldots + \mathbf{T}\left(\mathbf{I} - \mathbf{DL}\right)^{-1}\mathbf{CA}\left(\mathbf{C}^{+}\mathbf{JC} + \mathbf{C}^{\perp}(\mathbf{C}^{\perp})'\right)\mathbf{x}. \end{aligned} \tag{5.49}$$

Alternatively, using (5.47) the error dynamics can be expressed as

$$\begin{aligned} \dot{\mathbf{e}} &= \mathbf{TCAx} + \mathbf{TCBLT}^{+}\mathbf{e} + \mathbf{TDLT}^{+}\dot{\mathbf{e}} \\ \left(\mathbf{I} - \mathbf{TDLT}^{+}\right)\dot{\mathbf{e}} &= \mathbf{TCAx} + \mathbf{TCBLT}^{+}\mathbf{e} \\ \dot{\mathbf{e}} &= \left(\mathbf{I} - \mathbf{TDLT}^{+}\right)^{-1}\mathbf{TCAx} + \left(\mathbf{I} - \mathbf{TDLT}^{+}\right)^{-1}\mathbf{TCBLT}^{+}\mathbf{e} \end{aligned}$$

Note that $\left(\mathbf{I} - \mathbf{TDLT}^{+}\right)^{-1} = \mathbf{I} + \mathbf{TD}\left(\mathbf{I} - \mathbf{LT}^{+}\mathbf{TD}\right)^{-1}\mathbf{LT}^{+}$ and as $\mathbf{LT}^{+}\mathbf{T} = \mathbf{L}$,

$$\begin{aligned} \left(\mathbf{I} - \mathbf{TDLT}^{+}\right)^{-1}\mathbf{T} &= \left[\mathbf{I} + \mathbf{TD}\left(\mathbf{I} - \mathbf{LT}^{+}\mathbf{TD}\right)^{-1}\mathbf{LT}^{+}\right]\mathbf{T} \\ \left(\mathbf{I} - \mathbf{TDLT}^{+}\right)^{-1}\mathbf{T} &= \mathbf{T}\left[\mathbf{I} + \mathbf{D}\left(\mathbf{I} - \mathbf{LD}\right)^{-1}\mathbf{L}\right] \\ \left(\mathbf{I} - \mathbf{TDLT}^{+}\right)^{-1}\mathbf{T} &= \mathbf{T}\left(\mathbf{I} - \mathbf{DL}\right)^{-1} \end{aligned}$$

And from here equation (5.49) can easily be obtained.

As $\mathbf{u} = \mathbf{L}\mathbf{y} = \mathbf{L}(\mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}) \implies \mathbf{u} = (\mathbf{I} - \mathbf{L}\mathbf{D})^{-1}\mathbf{L}\mathbf{C}\mathbf{x}$, the dynamics of the error can be written as:

$$\dot{\mathbf{x}} = \left(\mathbf{A} + \mathbf{B}(\mathbf{I} - \mathbf{L}\mathbf{D})^{-1}\mathbf{L}\mathbf{C}\right)\mathbf{x} \tag{5.50}$$

Note that when $\mathbf{D} = \mathbf{0}$, equations (5.49) and (5.50) are equivalent to what has been studied before. However, the introduction of matrix $\mathbf{D}$ makes these expressions to be strongly non linear with respect to the algorithm matrix $\mathbf{L}$. Therefore, although it can be used for consensus analysis, it is not straight forward to derive LMI conditions suitable for algorithm design.

For analysis simplicity, the following matrices can be defined:

$$\mathbf{G}_D = \mathbf{T}(\mathbf{I} - \mathbf{D}\mathbf{L})^{-1}\mathbf{C}\left(\mathbf{B}\mathbf{L} + \mathbf{A}\mathbf{C}^+(\mathbf{I} - \mathbf{D}\mathbf{L} + \mathbf{J}\mathbf{D}\mathbf{L})\right)\mathbf{T}^+$$
$$\mathbf{R}_D = \mathbf{T}(\mathbf{I} - \mathbf{D}\mathbf{L})^{-1}\mathbf{C}\mathbf{A}\left(\mathbf{C}^+\mathbf{J}\mathbf{C} + \mathbf{C}^\perp(\mathbf{C}^\perp)'\right)$$
$$\mathbf{A}_{cl,D} = \mathbf{A} + \mathbf{B}(\mathbf{I} - \mathbf{L}\mathbf{D})^{-1}\mathbf{L}\mathbf{C}$$

From here, similar arguments as those exposed in Section 5.1.2 can be used to determinate if consensus can be reached in specific cases. Particularly, necessary conditions are that matrix $\mathbf{G}_D$ is Hurwitz and that the signal $\mathbf{r}_D = \mathbf{R}_D\mathbf{x}$ is either zero or vanishes with time.

Example 5.12. Consider $N = 10$ agents with identical individual dynamics given by:

$$\mathbf{A}_i = \begin{bmatrix} -0.1 & 0.1 \\ 0.1 & 0.0 \end{bmatrix}, \mathbf{B}_i = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}, \mathbf{C}_i = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}, \mathbf{D}_i = 1.$$

That is, the same as network e) in Example 5.3 but with $\mathbf{D} = \mathbf{I}$ instead of zero. In that example it was shown that considering the two consensus algorithms, $\mathbf{L}_g = -\hat{L}(\mathscr{G})$ and $\mathbf{L}_t = -\hat{L}(\mathscr{T}_w)$, and the organization $\mathbf{T} = D(\mathscr{T}^o)$ derived from Figure 5.5, the network reaches dynamic consensus as the residual signal $\mathbf{r} = \mathbf{R}_A\mathbf{x}$ vanishes with time even though $\mathbf{x}$ does not. However, including matrix $\mathbf{D} = \mathbf{I}$, the residual signal $\mathbf{r}_D = \mathbf{R}_D\mathbf{x}$ does not vanishes although $\mathbf{G}_D$ is Hurwitz and $\mathbf{A}_{cl,D}$ has only one unstable eigenvalue in both cases, which implies that consensus in the long term cannot be reached. This can be seen in the simulation results depicted in Figure 5.22 where the errors between the outputs of the systems ($\mathbf{e} = \mathbf{T}\mathbf{y}$) are shown under the same conditions as in Example 5.3. ■

5.3.2. Higher Order Dynamics

Up to this point, a fundamental assumption in all the discussed cases is that the product $\mathbf{C}\mathbf{B}$ is full rank. In an AAN, that is only possible if for all $i \in \mathscr{V}$, $\mathbf{C}_i\mathbf{B}_i \neq \mathbf{0}$. If this is not the case, then there are some elements of the error $\mathbf{e}$ in an AAN that do not depend on the consensus algorithm $\mathbf{L}$. In the extreme case when $\mathbf{C}\mathbf{B} = \mathbf{0}$, then $\mathbf{G}_A = \mathbf{T}\mathbf{C}\left(\mathbf{B}\mathbf{L} + \mathbf{A}\mathbf{C}^+\right)\mathbf{T}^+ = \mathbf{T}\mathbf{C}\mathbf{A}\mathbf{C}^+\mathbf{T}^+$ allowing the systems to reach consensus dependent only on the characteristics of matrices $\mathbf{A}$

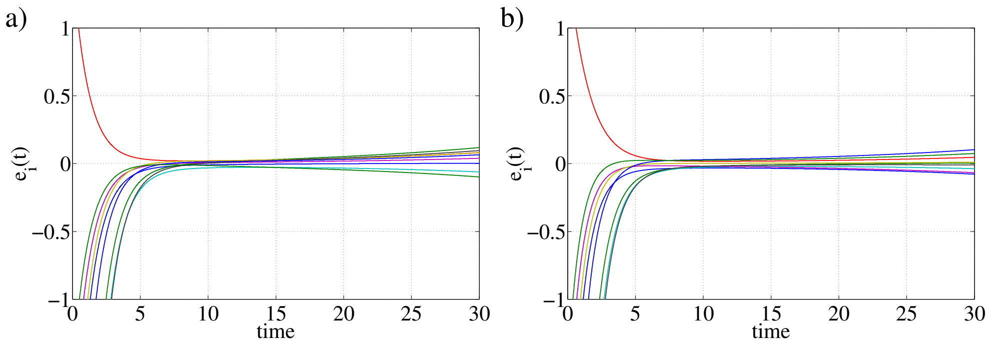


Figure 5.22.: Error evolution of the network in Example 5.12 under a) algorithm $\mathbf{L}_g = -\hat{L}(\mathscr{G})$ and b) $\mathbf{L}_t = -\hat{L}(\mathscr{T}_w)$.

and $\mathbf{C}$. The rate at which consensus is then reached does not depend on the chosen algorithm but merely on the dynamics of the agents.

This situation is not as unusual as desired. Consider for example a network composed only of AA with dynamics described by the canonical second order transfer function:

$$y_i(s) = \frac{k_i\omega_i}{s^2 + 2\omega_i\zeta_i s + \omega_i^2} u_i(s).$$

Their controllable canonical realization in the space of states is given by the matrices:

$$\mathbf{A}_i = \begin{bmatrix} 0 & 1 \\ -\omega_i^2 & -2\omega_i\zeta_i \end{bmatrix}, \mathbf{B}_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \mathbf{C}_i = \begin{bmatrix} k_i\omega_i & 0 \end{bmatrix}. \tag{5.51}$$

Clearly, $\mathbf{C}_i\mathbf{B}_i = \mathbf{0} \implies \mathbf{CB} = \mathbf{0}$ implying that consensus depends only on the poles of each agent. It can be further shown, that for transfer function of relative order greater than one (*i.e.*, where the difference between the number of poles and zeros of the transfer function is more that one), then the product $\mathbf{C}_i\mathbf{B}_i$ is always zero and consensus, for these agents, cannot be directly controlled. This fact makes consensus especially vulnerable to perturbations, uncertainties and other contradicting control objectives.

In an intuitive way, a strategy to deal with this problem is to "decrease" the order of the plant. That is, to include derivative terms in the control strategy so that the overall transfer function of each agent has a relative order of exactly one. This reminds naturally of a proportional-derivative (PD) controller and can, in practice, present the same implementation restrictions. Namely, the derivative of a quantity can only be approximated by analogous circuits but not exactly determined. However, in the special case where the states of the agents are available for feedback, the derivative of the output can be easily calculated. Therefore, in this section we would assume that each agent can obtain the exact value of the derivative of its output and communicate it to other agents without problems.

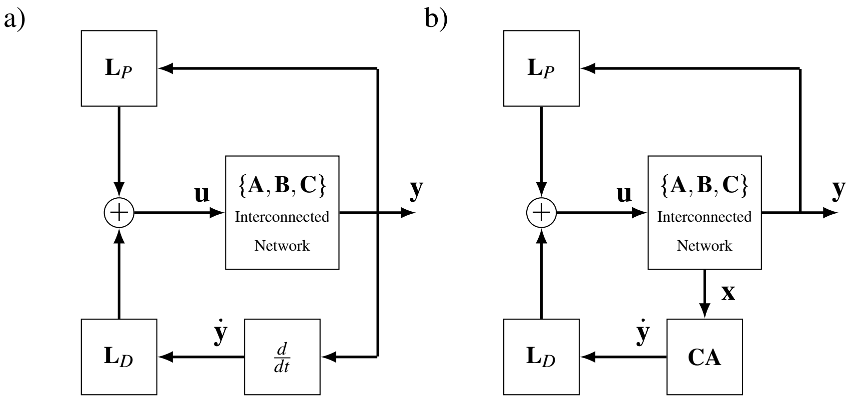


Figure 5.23.: Proportional Derivative Consensus Algorithm a) with Ideal Derivative Block, b) with States Feedback Implementation.

A PD-type consensus algorithm is proposed in the block diagram of Figure 5.23 a). Here the input of the agents is composed of the sum of a proportional and a derivative signal described by the equation

$$\mathbf{u} = \mathbf{L}_P\mathbf{y} + \mathbf{L}_D\dot{\mathbf{y}}. \tag{5.52}$$

Where $\mathbf{L}_P = -\hat{L}(\mathscr{G}_P) \in \mathbb{R}^{Nq\times Nq}$ and $\mathbf{L}_D = -\hat{L}(\mathscr{G}_D) \in \mathbb{R}^{Nq\times Nq}$ are two (possibly different) feedback matrices obtained from the undirected weighted graphs $\mathscr{G}_P$ and $\mathscr{G}_D$. Note that if $\mathbf{CB} = \mathbf{0}$,

$$\dot{\mathbf{y}} = \mathbf{C}\dot{\mathbf{x}} = \mathbf{CAx} + \mathbf{CBu} = \mathbf{CAx}.$$

Therefore, if there is access to the states vector $\mathbf{x}$, the derivative of the outputs can be precisely calculated by a simple matrix multiplication as in Figure 5.23 b). For an AAN, this calculation can be distributed in the agents as the product $\mathbf{CA} = \operatorname{diag}\{\mathbf{C}_i\mathbf{A}_i\}_{i\in\mathscr{V}}$ is block diagonal. Under this assumption it is not necessary to approximate the derivative of the output to use it for feedback. In more restrictive cases when the vector $\mathbf{x}$ is not available, a states observer strategy can still be followed to obtain a good approximation of the derivative of the output without the need to approximate it directly from the output.

Note that the inverse relationships between the error defined by an organization matrix $\mathbf{T} = D'(\mathscr{T}^o)$ and the output of the system, and their respective derivatives are

$$\mathbf{e} = \mathbf{Ty} \implies \mathbf{y} = \mathbf{T}^+\mathbf{e} + \mathbf{Jy}$$
$$\dot{\mathbf{e}} = \mathbf{T}\dot{\mathbf{y}} \implies \dot{\mathbf{y}} = \mathbf{T}^+\dot{\mathbf{e}} + \mathbf{J}\dot{\mathbf{y}}.$$

From here, equation (5.52) can be rewritten as

$$\begin{aligned}\mathbf{u} &= \mathbf{L}_P\mathbf{y} + \mathbf{L}_D\dot{\mathbf{y}}\\ &= \mathbf{L}_P(\mathbf{T}^+\mathbf{e} + \mathbf{Jy}) + \mathbf{L}_D(\mathbf{T}^+\dot{\mathbf{e}} + \mathbf{J}\dot{\mathbf{y}})\\ &= \mathbf{L}_P\mathbf{T}^+\mathbf{e} + \mathbf{L}_D\mathbf{T}^+\dot{\mathbf{e}}.\end{aligned}$$

For the consensus analysis of the network with the described strategy, it is not enough to obtain the expression of the derivative of the error as the product $\mathbf{CB} = \mathbf{0}$ makes it independent of the value of the input. Indeed,

$$\dot{\mathbf{e}} = \mathbf{TC}\dot{\mathbf{x}} = \mathbf{TCAx} + \mathbf{TCBu} = \mathbf{TCAC}^{+}\mathbf{T}^{+}\mathbf{e} + \mathbf{TCA}(\mathbf{C}^{+}\mathbf{JC} + \mathbf{C}^{\perp}(\mathbf{C}^{\perp})')\mathbf{x}.$$

To solve this problem, consider the fact that if a system reaches consensus, then it stays in consensus. *i.e.* $\mathbf{e} = \mathbf{0} \Rightarrow \dot{\mathbf{e}} = \mathbf{0}$ as shown at the end of Section 4.2.3. This implies that imposing stability of the aggregation of vector $\mathbf{e}$ and its derivative, is in fact not more restrictive than imposing stability of only $\mathbf{e}$. To make use of this, let us introduce the following auxiliary variable:

$$\mathbf{z} := \dot{\mathbf{e}} = \mathbf{T}\dot{\mathbf{y}} = \mathbf{TC}\dot{\mathbf{x}} = \mathbf{TCAx}$$

Its derivative is,

$$\begin{aligned}
\dot{\mathbf{z}} &= \mathbf{TCA}\dot{\mathbf{x}} = \mathbf{TCA}^2\mathbf{x} + \mathbf{TCABu} \\
&= \mathbf{TCA}^2\mathbf{C}^{+}\mathbf{T}^{+}\mathbf{e} + \mathbf{TCA}^2\left(\mathbf{C}^{+}\mathbf{JC} + \mathbf{C}^{\perp}(\mathbf{C}^{\perp})'\right)\mathbf{x} + \mathbf{TCAB}\left(\mathbf{L}_P\mathbf{T}^{+}\mathbf{e} + \mathbf{L}_D\mathbf{T}^{+}\mathbf{z}\right) \\
&= \mathbf{TCA}\left(\mathbf{AC}^{+} + \mathbf{BL}_P\right)\mathbf{T}^{+}\mathbf{e} + \mathbf{TCABL}_D\mathbf{T}^{+}\mathbf{z} + \mathbf{TCA}^2\left(\mathbf{C}^{+}\mathbf{JC} + \mathbf{C}^{\perp}(\mathbf{C}^{\perp})'\right)\mathbf{x}
\end{aligned}$$

From here, consensus can be studied by testing stability of the following enlarged system:

$$\begin{bmatrix} \dot{\mathbf{e}} \\ \dot{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{TCA}\left(\mathbf{AC}^{+} + \mathbf{BL}_P\right)\mathbf{T}^{+} & \mathbf{TCABL}_D\mathbf{T}^{+} \end{bmatrix} \begin{bmatrix} \mathbf{e} \\ \mathbf{z} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{TCA}^2\left(\mathbf{C}^{+}\mathbf{JC} + \mathbf{C}^{\perp}(\mathbf{C}^{\perp})'\right) \end{bmatrix}\mathbf{x}$$

Note that the system matrix in the previous expression depends on the feedback matrices $\mathbf{L}_P$ and $\mathbf{L}_D$ only if $\mathbf{CAB} \neq \mathbf{0}$. If that is not the case, then consensus cannot be controlled. Again, this happens when the relative order of the transfer functions of the agents is too large. For example, in the controllable canonical realization of a third order transfer function, $\mathbf{C}_i\mathbf{A}_i\mathbf{B}_i = \mathbf{0}$. Intuitively, the explanation of this is simple. Introducing one derivative block in the algorithm decreases the relative order of the plant in only one unit. Thus, to completely control consensus in a network with agents of relative order $o \in \mathbb{N}$, $o-1$ derivative terms should be included in the feedback law of higher order agents. Furthermore, stability then needs to be verified not only over the first derivative of the error, but on the first $o-1$ derivatives. For simplicity, we will not expressly address these higher order cases.

In the cases where the product $\mathbf{TCA}^2\left(\mathbf{C}^{+}\mathbf{JC} + \mathbf{C}^{\perp}(\mathbf{C}^{\perp})'\right)\mathbf{x}$ does not vanish, then the dynamics of the states need also to be taken into account as in the previous sections. For a PD-type algorithm, replacing (5.52) into the expression of the dynamics of the states $\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}$, with $\mathbf{CB} = \mathbf{0}$, one obtains directly:

$$\begin{aligned}
\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{B}\left(\mathbf{L}_P\mathbf{Cx} + \mathbf{L}_D\mathbf{C}\dot{\mathbf{x}}\right) \\
&= \mathbf{Ax} + \mathbf{BL}_P\mathbf{Cx} + \mathbf{BL}_D\mathbf{C}\left(\mathbf{Ax} + \mathbf{Bu}\right) \\
&= \left(\mathbf{A} + \mathbf{BL}_P\mathbf{C} + \mathbf{BL}_D\mathbf{CA}\right)\mathbf{x}.
\end{aligned}$$

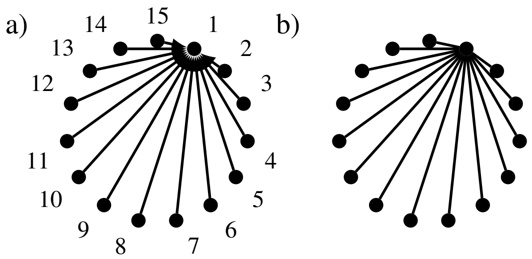


Figure 5.24.: a) Organization Tree $\mathscr{T}^o$, b) Unweighted Tree $\mathscr{T}$ for Example 5.13.

Example 5.13. Consider $N = 15$ identical double order agents with dynamics described by the matrices in (5.51). The parameter values of the agents are given by $k_i = 3.00$, $\zeta_i = 0.25$ and $\omega_i = 0.10$, $\forall i \in \mathscr{V}$. The transfer function of the agents has poles given by $\operatorname{eig}\{\mathbf{A}_i\} = \{-0.0250 \pm 0.0968i\}$. That is, the systems are stable but oscillatory.

The organization matrix to be considered is derived from the directed tree in Figure 5.24 a). The proportional and derivative gains of the PD-type algorithm are obtained from weighted graphs $\mathscr{T}_P$ and $\mathscr{T}_D$, that have the same shape of the unweighted graph in Figure 5.24 b), with weight functions that gives to all edges in $\mathscr{T}_P$ the weight $w_P = 10$ and to all edges in $\mathscr{T}_D$ the weight $w_D = 100$.

In this case we obtain that the system matrix,

$$\mathbf{G}_{PD} := \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{TCA}\left(\mathbf{AC}^+ + \mathbf{BL}_P\right)\mathbf{T}^+ & \mathbf{TCABL}_D\mathbf{T}^+ \end{bmatrix},$$

is Hurwitz as it only has real eigenvalues from which the largest is $\lambda_{PD} = -0.1$. Even though $\mathbf{R}_A \neq \mathbf{0}$, the residual signal $\mathbf{r} = \mathbf{R}_A\mathbf{x}$ vanishes as matrix $\mathbf{A} + \mathbf{BL}_P\mathbf{C} + \mathbf{BL}_D\mathbf{CA}$ is also Hurwitz, but with conjugate complex eigenvalues. From this analysis, it can be concluded that consensus will be reached with the proposed PD-type algorithm.

This is verified by the simulations shown in Figure 5.25 a) and b), where the evolution of the outputs of the network are shown for an arbitrary initial condition with an ideal derivative block in a) and a states feedback block in b). In Figure 5.25 c), the response of the network with only the proportional part of the controller is shown (that is, when $\mathbf{L}_D = \mathbf{0}$). In d) the response of the network with only the derivative part is shown (when $\mathbf{L}_P = \mathbf{0}$). From these figures it is clear that consensus cannot be reached successfully only by considering the proportional or derivative part of the algorithm. ∎

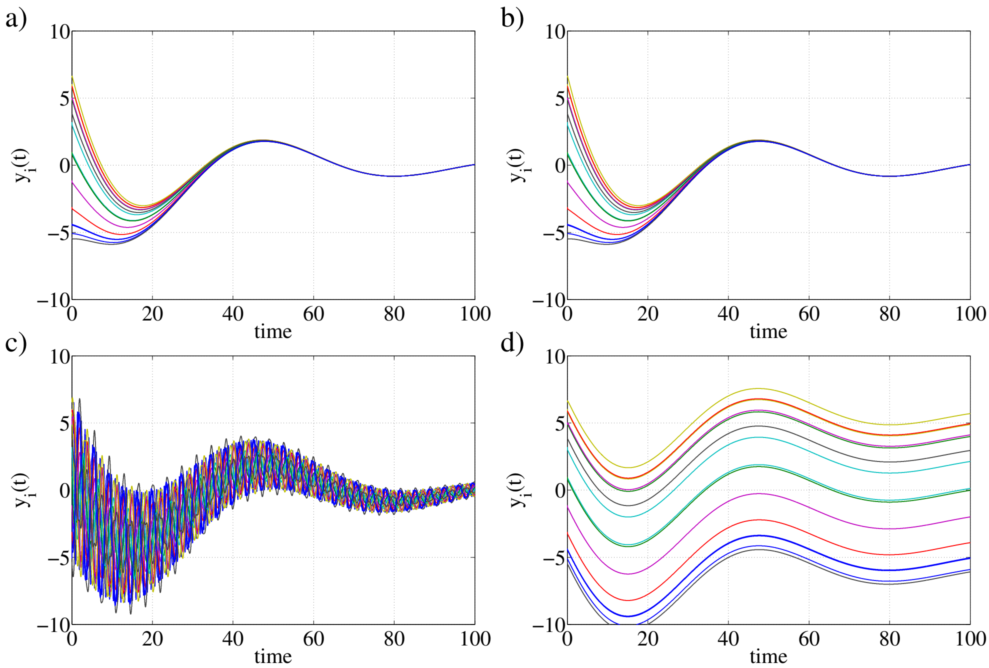


Figure 5.25.: Output evolution of the network in Example 5.13 under a) PD-type algorithm with ideal derivative block, b) PD-type algorithm with states feedback, c) P-type algorithm, and d) D-type algorithm with states feedback.

5.3.3. Communication Constrains

Up to this point, and in most of the consensus related references, consensus is studied making emphasis on the algorithmic properties of the control feedback. That is to answer the question *which information needs to be shared in order to reach consensus?* This however does not describe the information sharing process. In this sense, a feedback law $\mathbf{u} = \mathbf{L}\mathbf{y}$ as studied before, does not model the communications links between the agents, but merely describes which signals need to be shared. Even though a consensus algorithm implies in most cases that certain signals need to be communicated between the agents, the communication process itself is assumed to be ideal and only dynamical behaviors of the agents are considered. However, in more realistic scenarios, it is sometimes necessary to explicitly model communication constrains.

To study the role of communication on a given network, with a given consensus algorithm, is to answer the question, *how the information is shared?* Which dynamical processes affect the information while it travels from one agent to the other. This fundamental difference between what is shared and how it is shared, is sometimes overseen making the study of such problems confusing. The idea of the separation between an intended algorithm and the communication behavior can already be observed in Section 5.2.3 where switching algorithms are studied. In the unintended switching case, we deal with the possibility that communication channels temporary fail, preventing to implement the chosen nominal algorithm.

This case also gives some hints on the main complication of studying communication dynamics separately. In general, even for networks that reach consensus in the ideal case, when the communication dynamics are considered, the residual signal is modified by the dynamics of the communication channels (*e.g.* by the failures in the unintended switching case). Thus, there is no guarantee that consensus can be reached. As with AANs, it is difficult to characterize consensus *a priori* by a single numerical indicator. This is further complicated as the residual signal might not only depend on the characteristics of the network or the channels, but also on the chosen algorithm.

Coming back to the general model described in Section 4.1.1, to study the communication process is to deal with the characteristics of the block labeled *Communication Channels* in Figure 4.1. This block can be modeled with more detail as a collection of point-to-point communication channels that can be subjected to different kind of dynamical behavior such as switching failures, filtering, noise, uncertainties, or delays. In a network of N agents, there are N^2 possible point-to-point communication channels counting the feedback channel of each system to itself. Whether these channels are used by the consensus algorithm or not, is independent of their dynamical characteristics and therefore a differentiation between the algorithm and the implementation of the communication process must be made.

A communication channel $(i, j) \in \mathscr{V} \times \mathscr{V}$ from agent i to agent j is a (possibly) dynamical system such that its input is a measurement of the output of system i and its output is an input for the controller of agent j. The set of all N^2 possible communication channels is denoted $\mathscr{H} = \mathscr{V} \times \mathscr{V}$. For convenience of notation, as in [35], a communication channel can be labeled

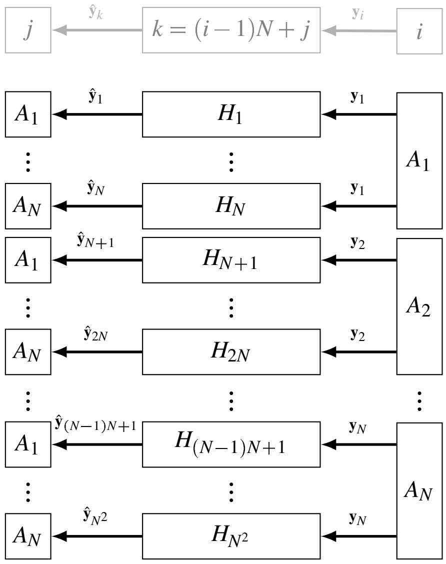


Figure 5.26.: Communication channels.

by an auxiliary index $k=(i-1)N+j \in \{1,\ldots,N^2\}$, where i is the system sending a signal to system j through the communication channel k. Note that as $(i,j)\in\mathscr{V}\times\mathscr{V}$, the pair (i,j) can always be obtained from $k=(i-1)N+j$, *i.e.* $(i,j) \iff k$.

From here, we define N^2 vectors $\hat{\mathbf{y}}_k\in\mathbb{R}^q$ that represent the communicated signals from system i to system j. A schematic representation of the communication channels between agent i to agent j, and their respective labeling, can be seen in Figure 5.26. Taking this into consideration, the feedback signal $\mathbf{u}$ can be characterized by the control law

$$\mathbf{u}=\hat{\mathbf{L}}\hat{\mathbf{y}}$$

where $\hat{\mathbf{y}}=\mathrm{col}\,\{\hat{\mathbf{y}}_k\}_{k=1}^{N^2}\in\mathbb{R}^{N^2q}$ and $\hat{\mathbf{L}}\in\mathbb{R}^{Nq\times N^2q}$ is obtained from an ideal consensus algorithm described by $\mathbf{L}\in\mathbb{R}^{Nq\times Nq}$ in equation (5.2) in the following way:

$$\hat{\mathbf{L}}=\sum_{i=1}^{N}\mathbf{s}_i\mathbf{s}_i'\mathbf{L}\mathbf{S}_i \tag{5.53}$$

Where, $\mathbf{s}_i\in\mathbb{R}^{Nq\times q}$ is a block column vector of N blocks with zeros everywhere but in the i-th block element that is the identity matrix $\mathbf{I}_q$, so that $\mathbf{1}=\sum_{i=1}^{N}\mathbf{s}_i$. Additionally, $\mathbf{S}_i=\mathrm{diag}\,\{\mathbf{s}_{\hat{\imath}}'\}_{\hat{\imath}=1}^{N}$ so that $\mathbf{S}=\sum_{i=1}^{N}\mathbf{S}_i=\mathrm{diag}\,\{\mathbf{1}'\}_{\hat{\imath}=1}^{N}$. $\hat{\imath}\in\mathscr{V}$ is an auxiliary index different from $i\in\mathscr{V}$. From here,

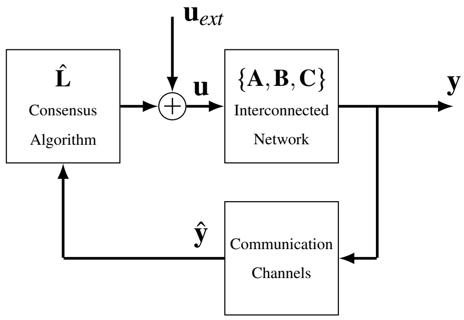


Figure 5.27.: Consensus under communication constrains block diagram.

the feedback signal $\mathbf{u}$ can be written as:

$$\mathbf{u} = \begin{bmatrix} -\Delta_{11}\hat{\mathbf{y}}_1 & \mathbf{W}_{12}\hat{\mathbf{y}}_2 & \cdots & \mathbf{W}_{1N}\hat{\mathbf{y}}_N \\ \mathbf{W}_{21}\hat{\mathbf{y}}_{N+1} & -\Delta_{22}\hat{\mathbf{y}}_{N+2} & \cdots & \mathbf{W}_{2N}\hat{\mathbf{y}}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{W}_{N1}\hat{\mathbf{y}}_{(N-1)N+1} & \mathbf{W}_{N2}\hat{\mathbf{y}}_{(N-1)N+2} & \cdots & -\Delta_{NN}\hat{\mathbf{y}}_{N^2} \end{bmatrix} \mathbf{1}_{N\times 1} \tag{5.54}$$

Note the similarity with the expression in equation (5.2). In the ideal case where all communication channels are modeled as the identity function, the feedback signal becomes $\mathbf{u} = \hat{\mathbf{L}}\hat{\mathbf{y}} = \mathbf{L}\mathbf{y}$. A schematic representation of a non ideal consensus algorithm can be seen in Figure 5.27.

By expressly modeling each of the N^2 signals $\hat{\mathbf{y}}_k$, consensus can be studied under various cases including filtering, delays, noise and uncertainties. As can be supposed from the definition of matrix $\hat{\mathbf{L}}$, this is often laborious and demanding in notation. However, the main problem is that the communicated signals are dynamical functions of the outputs of the systems, *i.e.* $\hat{\mathbf{y}} = Com(t,\mathbf{y})$. Therefore, there is no guarantee that the feedback signal $\mathbf{u}$ can be expressed as a function of the error only, even when in the ideal case $\mathbf{u} = \mathbf{L}\mathbf{T}^+\mathbf{e} + \mathbf{L}\mathbf{J}\mathbf{y} = \mathbf{L}\mathbf{T}^+\mathbf{e}$. In general, $\mathbf{u}$ is a dynamical function of $\hat{\mathbf{L}}$, $\mathbf{e}$, $\mathbf{x}$, and time. Thus, a residual signal, different from that of the ideal case, is always present in the expression for the dynamics of the error, $\dot{\mathbf{e}} = \mathbf{TCBu} + \mathbf{TCAC}^+\mathbf{T}^+\mathbf{e} + \mathbf{TCA}\left(\mathbf{C}^+\mathbf{JC} + \mathbf{C}^\perp(\mathbf{C}^\perp)'\right)\mathbf{x}$. As seen in the AAN case, this makes it difficult to study consensus in general.

Example 5.14. To exemplify the influence of the states in the search for consensus under different dynamical communication cases, consider $N = 4$ identical agents described $\forall i \in \{1,2,3,4\}$ by

$$\mathbf{A}_i = \begin{bmatrix} 0.01 & 0.00 \\ 0.00 & -0.02 \end{bmatrix}, \ \mathbf{B}_i = \begin{bmatrix} 1.00 & 0.50 \\ 0.30 & 1.00 \end{bmatrix}, \ \mathbf{C}_i = \mathbf{I}_{2\times 2}.$$

Note that, as all agents are identical and matrix $\mathbf{C}_i$ is square and non-singular. Thus, the residual matrix in the ideal case is identically zero for any organization: $\mathbf{R}_A = \mathbf{TCAC}^{-1}\mathbf{JC} = \mathbf{TJCAC}^{-1}\mathbf{C} = \mathbf{0}$.

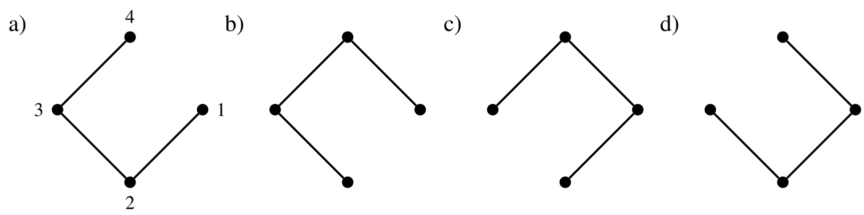


Figure 5.28.: Algorithmic undirected unweighted graphs $\mathscr{G}_i$, with $i \in \{a,b,c,d\}$, for Example 5.14.

Consider four algorithms derived as negative Laplacian matrices of the loopless unweighted undirected graphs in Figure 5.28: $\mathbf{L}_i = -\hat{L}(\mathscr{G}_i)$ with $i \in \{a,b,c,d\}$. In the ideal case, all these algorithms make the respective system matrix $\mathbf{G}_A = \mathbf{TC}\left(\mathbf{BL} + \mathbf{AC}^+\right)\mathbf{T}^+$ to be Hurwitz, therefore they allow to reach consensus. We consider also an organization centered in the first agent with a matrix $\mathbf{T} = \text{row}\{\mathbf{1}, -\mathbf{I}\}$.

As all agents are the same and the algorithms are derived from isomorphic graphs, any given set of initial conditions can be permuted in such a way that the time response of the network under different algorithms is equivalent. That is, that each trajectory followed by the agents under the action of one algorithm, is exactly replicated by other agent when other algorithm is considered. This can be seen in Figure 5.29. Note that the four plots, corresponding to the four different controllers, are identical except for the agents assigned to each initial condition (*i.e.*, the colors of the lines are different). Observe further that in all cases, the network reaches consensus.

Now consider that the communication channels are modeled so that only feedback channels are ideal. Consider that the rest of the channels is modeled by first order dynamical filters:

$$\dot{\hat{\mathbf{y}}}_k = -\frac{1}{\tau_k}\hat{\mathbf{y}}_k + \frac{1}{\tau_k}\mathbf{y}_i,$$

with $k = (i-1)N + j$, and

$$\tau_k = \begin{cases} 4.00, & (i,j) \in \{(1,2),(2,1)\} \\ 2.00, & (i,j) \in \{(2,3),(3,2)\} \\ 2.50, & (i,j) \in \{(3,4),(4,3)\} \\ 1.50, & (i,j) \in \{(1,4),(4,1)\} \end{cases}$$

Because of the dynamics of the channels, the network behaves differently as in the ideal case. This can be seen in Figure 5.30 where the time responses of the network with the non-ideal algorithms are depicted under identical initial conditions for the states and zero initial conditions for the dynamical communication channels. From the plots it is not clear if the networks reach consensus in all cases.

Consider now that the non ideal channels are described through delays of the input signal such that

$$\hat{\mathbf{y}}_k(t) = \begin{cases} \mathbf{y}_i(t-0.90), \ (i,j) \in \{(1,2),(2,1)\} \\ \mathbf{y}_i(t-0.30), \ (i,j) \in \{(2,3),(3,2)\} \\ \mathbf{y}_i(t-0.50), \ (i,j) \in \{(3,4),(4,3)\} \\ \mathbf{y}_i(t-0.20), \ (i,j) \in \{(1,4),(4,1)\} \end{cases},$$

with $k = (i-1)N + j$.

Stability conditions can be formulated for the time delayed error, when the residual signal is neglected. This can be done in an LMI framework through Lyapunov-Krasovskii functions. Some examples in this area are [85, 92–94, 97, 110, 114]. However, in spite of these conditions, under identical simulation conditions as in the ideal case, from Figure 5.31 it is also not clear that the network reaches consensus for all equivalent algorithms. This shows that the effect of the states and the residual signal cannot be simply ignored to study consensus.

Slightly different is the case where the channels are affected by noise. In this case,

$$\hat{\mathbf{y}}_k = \mathbf{y}_i + \nu_k \mathbf{n}_k,$$

where $k = (i-1)N + j$, $\mathbf{n}_k \in \mathbb{R}^q$ are unknown white noise vectors that are uncorrelated with each other, and ν_k is a known parameter of the channel that represents the standard deviation of the noise. A large value for ν_k implies a noisier channel. Here, the noise signal can be considered as an external randomly distributed perturbation added to the transmitted signal and therefore an H_∞-norm argument can be followed to categorize different algorithms by defining a transfer function between the aggregation of the noise signals and the error.

For this example, the non-ideal channels are such that they are affected by noise with the following parameters

$$\nu_k = \begin{cases} 0.1, \ (i,j) \in \{(1,2),(2,1)\} \\ 0.3, \ (i,j) \in \{(2,3),(3,2)\} \\ 0.8, \ (i,j) \in \{(3,4),(4,3)\} \\ 0.7, \ (i,j) \in \{(1,4),(4,1)\} \end{cases}.$$

The network is simulated under identical conditions as before but considering the noisy channels. This can be seen in Figure 5.32. Clearly, the presence of noise makes it impossible to reach consensus, however choosing different algorithms makes it possible to minimize this effect. In particular, the simulation indicators *ISD* and *IAD* in all four cases are shown in Table 5.4 along with the H_∞-norm of the transfer function between the noise and the error. It is clear that a better performance is explained by a lower value of the norm.

From these examples it becomes clear that the effect of the residual signals cannot be simply neglected when studying consensus under communication constrains. This makes it hard to

Table 5.4.: Performance of different algorithms under noisy communication in Example 5.14.

Algorithm	$\|H_{ne}(s)\|_\infty$	$ISD(30)$	$IAD(30)$
$\mathbf{L}_a$	1.597554	26.761442	65.060689
$\mathbf{L}_b$	1.822130	26.877189	65.749193
$\mathbf{L}_c$	1.084465	25.901879	63.286071
$\mathbf{L}_d$	1.254586	26.217052	63.852563

predict the behavior of the network in advance by only considering the parameters of the dynamical behavior of the communications channels. Further development needs to take place to study these issues in more detail. ■

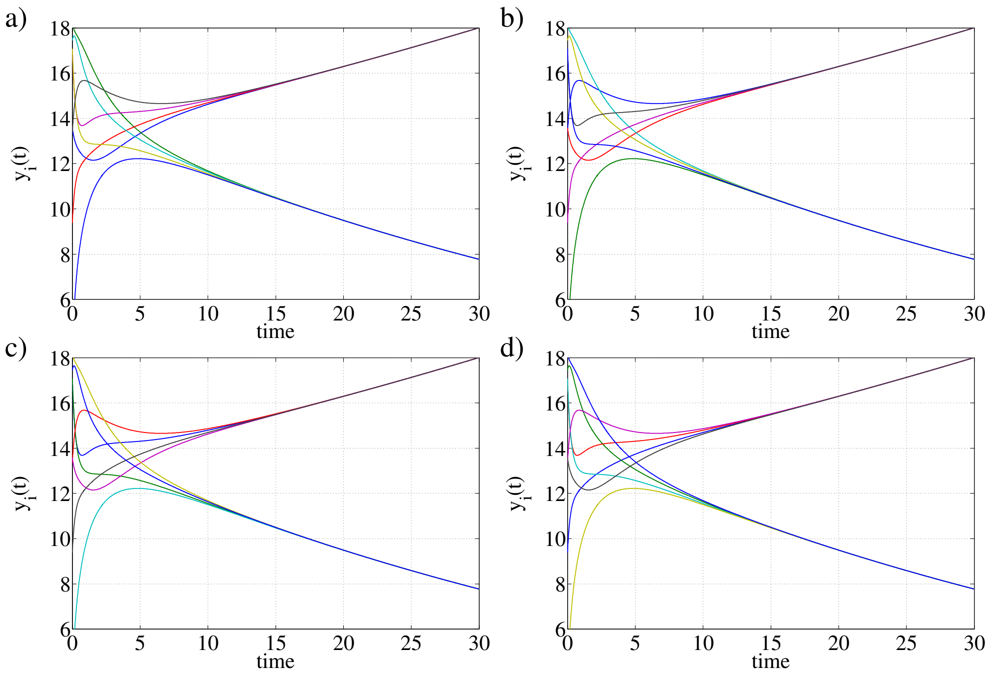


Figure 5.29.: Outputs evolution for the network analyzed in Example 5.14 with ideal communication and algorithms a) $\mathbf{L}_a$, b) $\mathbf{L}_b$, c) $\mathbf{L}_c$, and d) $\mathbf{L}_d$.

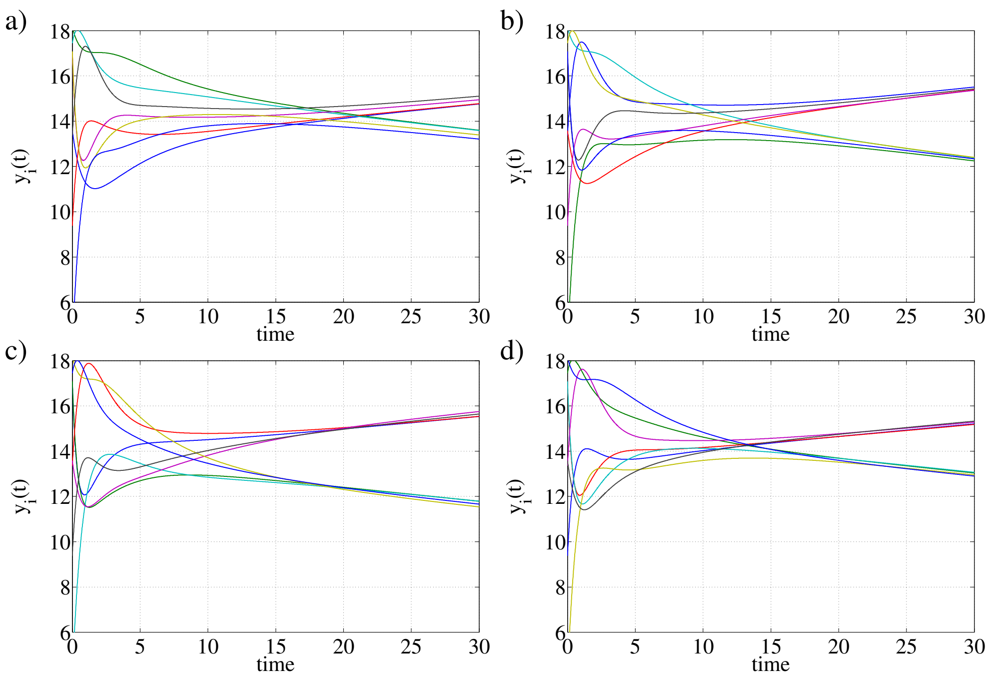


Figure 5.30.: Outputs evolution for the network analyzed in Example 5.14 with filtered communication and algorithms a) $\mathbf{L}_a$, b) $\mathbf{L}_b$, c) $\mathbf{L}_c$, and d) $\mathbf{L}_d$.

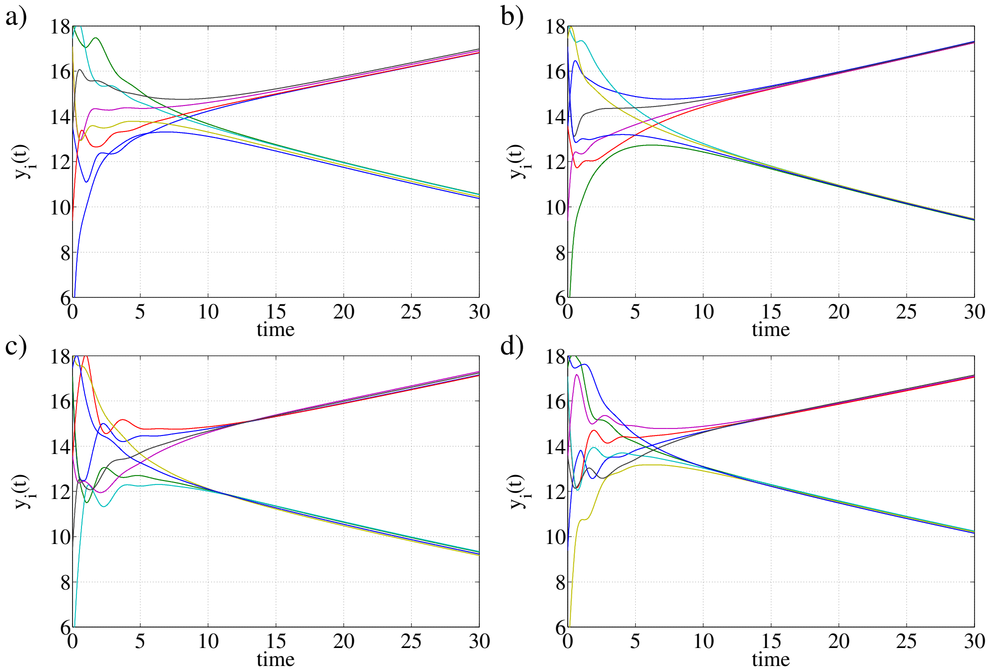


Figure 5.31.: Outputs evolution for the network analyzed in Example 5.14 with delayed communication and algorithms a) $\mathbf{L}_a$, b) $\mathbf{L}_b$, c) $\mathbf{L}_c$, and d) $\mathbf{L}_d$.

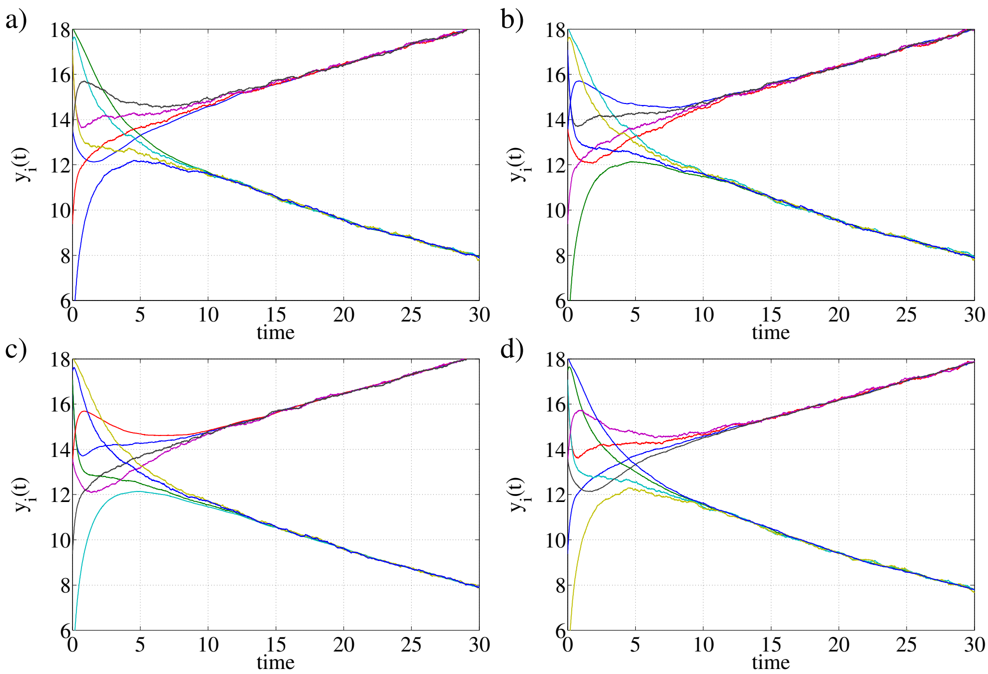


Figure 5.32.: Outputs evolution for the network analyzed in Example 5.14 with noisy communication and algorithms a) $\mathbf{L}_a$, b) $\mathbf{L}_b$, c) $\mathbf{L}_c$, and d) $\mathbf{L}_d$.

6
Power Consensus in Microgrids

Microgrids are a promising solution for the integration of renewable power sources into the existing energetic matrix and for energy supply of remote areas, where significant social impact can be achieved [177]. Therefore, during recent years, this technology has garnered a lot of attention from the control community. Some recent examples of publications related to the many technical challenges of microgrids are [174, 178–180, 185]. This chapter is related to the thesis [182] and the associated papers [181, 183, 184]. However, a different perspective is followed here as the control of microgrids is seen as an application of the concepts studied in the previous chapters and not as a goal in itself.

Besides frequency and voltage stability of the grid, an important control topic is (active or reactive) *power sharing*. This is understood as the ability of all connected three-phase generation units to supply the demanded steady state power in a predefined proportion. That is, to share the demanded load at an equivalent rate of the total capability of each generation unit. The advantage of achieving this control objective lies in the possibility of predefining operational set points for all generation units in order to avoid overload or operation under the allowed minimum. This can be translated into a consensus problem where the generation units play the role of the agents and the normalized power is the output variable.

In traditional synchronous machine based grids, *active* power sharing is achieved by what is known as *droop control*. This is a set of proportional controllers, each implemented in every generation unit, without communication between them. A similar approach has been proposed by researchers for AC inverter based microgrids. In this chapter, we first adopt this approach to study active power sharing, extending the results to analyze and design possible communication links between different generation units. In the second part of this chapter, the case of *reactive* power sharing is studied by following the approach in [181], where communication between the inverters is proposed as an alternative to *voltage droop control*. In both cases, the problem is formulated in a consensus based framework.

6.1. Microgrid Modeling

A microgrid can be interpreted as an undirected graph $\mathscr{G} = (\mathscr{V}, \mathscr{E}, w)$ where each electrical node $i \in \mathscr{V}$ is a vertex and the transmission lines are weighted edges. For the analysis of a microgrid, it is a common assumption to consider grids composed only of active nodes, *i.e.* nodes where voltage regulation can be done. This can be achieved by a Kron reduction procedure as described in [172]. It is also assumed that the grid is connected and therefore every active node $i \in \mathcal{V}$ has at least one neighbor node $j \in \mathcal{N}_i$.

At every active node $i \in \mathcal{V}$, a balanced load described by a resistance R_{ii} in series with an inductance L_{ii} will be considered. The corresponding complex impedance is defined in *fasor* notation as $Z_{ii} = R_{ii} + \hat{\imath}\omega_o L_{ii}$, with $\hat{\imath} = \sqrt{-1}$ the imaginary unit, $\omega_o = 2\pi f_o$, and f_o the nominal operation frequency of the grid. The power factor at each load is

$$p_{f,i} = \cos\left(\arctan\left(\frac{\omega_o L_{ii}}{R_{ii}}\right)\right).$$

A transmission line between nodes $i \in \mathcal{V}$ and $j \in \mathcal{V}$ will also be assumed as a balanced impedance composed by a resistance R_{ij} in series with an inductance L_{ij}. Note that always $R_{ij} = R_{ji}$ and $L_{ij} = L_{ji}$. In fasor notation, the associated complex impedance is $Z_{ij} = R_{ij} + \hat{\imath}\omega_o L_{ij}$. It is assumed that these line parameters can be estimated with reasonable accuracy due to the small size of a microgrid.

6.1.1. Power Flow

When the electric angle ψ of a three-phase balanced sinusoidal signal $\mathbf{x}^{abc}$ is described by a constant frequency ω and a phase shift angle $\delta(t)$, *i.e.* $\psi(t) = \omega t + \delta(t)$, then it can be represented in an equivalent rotatory reference frame as $\mathbf{x}^{dq} = [x^q,\ x^d]'$ by means of Park's transformation [175]. See Appendix B.1 for details.

In this notation, if $\mathbf{i}_i^{abc}$ is the current injected by an inverter in an active node $i \in \mathcal{V}$ and $\mathbf{v}_i^{abc}$ the voltage at that node, then the active power injected by the inverter can be defined by $P_i = i_i^d v_i^d + i_i^q v_i^q$. Analogously, the reactive power injected by the inverter is defined as $Q_i = i_i^q v_i^d - i_i^d v_i^q$.

Dynamical relationships as function of the node voltages for the current can be obtained from a circuital analysis of the grid. From here, if the transient behavior of the line dynamics is neglected, expressions for the active and reactive power injected by each inverter can be obtained as functions of the voltage angle δ_i and amplitude V_i as in the following expressions.

$$\begin{aligned} P_i(t) = {} & \frac{3}{2}\left[\frac{R_{ii}}{R_{ii}^2 + \omega^2 L_{ii}^2} + \sum_{j\in\mathcal{N}_i}\frac{R_{ij}}{R_{ij}^2 + \omega^2 L_{ij}^2}\right] V_i^2(t) + \dots \\ & - \frac{3}{2}\sum_{j\in\mathcal{N}_i}\frac{R_{ij}}{R_{ij}^2 + \omega^2 L_{ij}^2} V_i(t)V_j(t)\cos\left(\delta_i(t) - \delta_j(t)\right) + \dots \\ & + \frac{3}{2}\sum_{j\in\mathcal{N}_i}\frac{\omega L_{ij}}{R_{ij}^2 + \omega^2 L_{ij}^2} V_i(t)V_j(t)\sin\left(\delta_i(t) - \delta_j(t)\right) \end{aligned} \tag{6.1}$$

$$
\begin{aligned}
Q_i(t) = \frac{3}{2}\left[\frac{\omega L_{ii}}{R_{ii}^2+\omega^2 L_{ii}^2}+\sum_{j\in\mathcal{N}_i}\frac{\omega L_{ij}}{R_{ij}^2+\omega^2 L_{ij}^2}\right]V_i^2(t)+\ldots \\
-\frac{3}{2}\sum_{j\in\mathcal{N}_i}\frac{\omega L_{ij}}{R_{ij}^2+\omega^2 L_{ij}^2}V_i(t)V_j(t)\cos\left(\delta_i(t)-\delta_j(t)\right)+\ldots \\
-\frac{3}{2}\sum_{j\in\mathcal{N}_i}\frac{R_{ij}}{R_{ij}^2+\omega^2 L_{ij}^2}V_i(t)V_j(t)\sin\left(\delta_i(t)-\delta_j(t)\right)
\end{aligned} \tag{6.2}
$$

Where $V_i(t)$ is the amplitude of the sinusoidal voltage wave at the output of inverter $i \in \mathcal{V}$ (not its RMS value). A detailed development to obtain such models can be seen in Appendix B.2. In practice, power is measured through first order filters with time constants $\tau_{p,i}$ and $\tau_{q,i}$:

$$
\begin{aligned}
\tau_{p,i}\dot{P}_{m,i} &= P_i - P_{m,i} \\
\tau_{q,i}\dot{Q}_{m,i} &= Q_i - Q_{m,i}.
\end{aligned}
$$

However, when the time constants are sufficiently small these dynamics can be ignored.

6.1.2. Voltage Source Inverter Model

The model presented in this section is based on the thesis [182] and the associated papers [181, 183, 184]. Other sources include, *e.g.*, [178, 180]. In general, a Voltage Source Inverter (VSI) is a highly non-linear system where the frequency and the amplitude of the output voltage signal can be controlled through two different inputs. However, under some assumptions, a relative simple model can be used to used to describe its approximated behavior. A VSI, considering internal control loops, switching modulation and appropriate filtering, can be modeled by the following set of equations:

$$
\begin{aligned}
\dot{\psi}_i(t) &= u^{\psi_i}(t) \\
\tau_{V_i}\dot{V}_i &= -V_i + u^{V_i},
\end{aligned}
$$

Where $\psi_i(t) = \omega t + \delta_i(t)$ and V_i are, respectively, the electric angle and voltage amplitude of the output AC voltage signal in the *abc* representation; u^{ψ_i} is a control frequency input, and u^{V_i} a control amplitude input. This model assumes that each inverter is equipped with some DC storage unit, large enough to increase and decrease the AC power output in a certain range. Usually it is considered that the time constant $\tau_{V_i} \ll 1$, so that the previous equation is simply reduced to:

$$
\dot{\psi}_i(t) = u^{\psi_i}(t) \tag{6.3}
$$

$$
V_i = u^{V_i}. \tag{6.4}
$$

In the following sections, the problem of power sharing will be treated separately for active and reactive power.

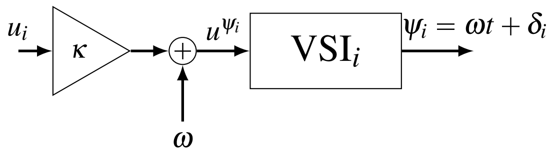


Figure 6.1.: VSI frequency control.

6.2. Active Power Sharing

6.2.1. Control Strategy

The model for power flow in equation (6.1) is highly non linear. Therefore, a suitable simplification should be considered to treat the problem of power sharing with the developed consensus tools. In the case that the voltage amplitude at every node is constant and equal (that is $\forall i \in \mathscr{V}, V_i(t) = V$) and the angle differences are small ($|\delta_i(t) - \delta_j(t)| \ll 1$) so that $\sin(\delta_i(t) - \delta_j(t)) \approx \delta_i(t) - \delta_j(t)$ and $\cos(\delta_i(t) - \delta_j(t)) \approx 1$, then one can obtain the following affine approximation:

$$P_i \quad \approx \quad P_{ii} + \sum_{j \in \mathscr{N}_i} Q_{ij} \left(\delta_i(t) - \delta_j(t)\right),$$

where

$$P_{ii} = \frac{3}{2} \frac{R_{ii}}{R_{ii}^2 + \omega^2 L_{ii}^2} V^2, \quad Q_{ij} = \frac{3}{2} \frac{\omega L_{ij}}{R_{ij}^2 + \omega^2 L_{ij}^2} V^2.$$

Note that $Q_{ij} = Q_{ji}$ and that this approximation is only accurate if the derivative of the electric angles ψ_i of every inverter is approximately equal to the nominal value ω.

The inverter is operated around the nominal frequency ω and therefore its input is modified by:

$$u^{\psi_i}(t) = \omega + \kappa u_i(t),$$

where $\kappa > 0$ is a constant equal for all inverters thought to limit the magnitude of signals u_i after feedback. If this constant is too large, then the model assumptions $\psi = \omega t + \delta$ and $\delta_i \approx \delta_j$ might not be satisfied in closed loop. Note that $u^{\psi_i}(t) = \dot{\psi}_i(t) = \omega + \dot{\delta}_i(t)$ what implies that $\dot{\delta}_i(t) = \kappa u_i(t)$. This control strategy can be seen in Figure 6.1.

A so called *droop controller* is commonly defined through a proportional gain $k_{p,i}$ such that $u_i = -k_{p,i} P_{m,i}$ with $P_{m,i}$ the measured power injected at node i. This kind of controller will be treated later as a special case. A more general model can be obtained defining the phase angle as state variable $x_i := \delta_i = \psi_i - \omega t$ and the normalized power as output variable $y_i := \frac{1}{\chi_i} P_i$. Here $\chi_i > 0$ is the power sharing constant of each inverter. A practical choice of the proportional constants would be $\chi_i = S_i$ where S_i is the nominal power rating of the inverter at node $i \in \mathcal{V}$. With this, the normalized active power injected by inverter i can be written as:

$$\begin{aligned}
\dot{x}_i &= \kappa u_i \\
y_i &= \frac{1}{\chi_i} P_{ii} + \frac{1}{\chi_i} \sum_{j \in \mathcal{N}_i} Q_{ij} \left(x_i - x_j \right).
\end{aligned}$$

6.2.2. Aggregated Microgrid Active Power Control Model

Given the model of each VSI, the whole grid can be characterized as a MIMO system by the following compact equations:

$$\begin{aligned}
\dot{\mathbf{x}} &= \kappa \mathbf{u} \\
\mathbf{y} &= \mathbf{F}\mathbf{C}\mathbf{x} + \mathbf{d}
\end{aligned} \tag{6.5}$$

where the elements of matrix $\mathbf{C}$ are given $\forall i, j \in \mathcal{V}$ by:

$$[\mathbf{C}]_{ij} = \begin{cases} \sum_{k \in \mathcal{N}_i} Q_{ik} & \text{if } \ i = j, \\ -Q_{ij} & \text{if } \ j \in \mathcal{N}_i, \\ 0 & \text{i.o.c.} \end{cases}$$

and,

$$\begin{aligned}
\mathbf{F} &= \operatorname{diag}\left\{1/\chi_i\right\}_{i \in \mathcal{V}}, \\
\mathbf{x} &= \operatorname{col}\left\{x_i\right\}_{i \in \mathcal{V}} = \operatorname{col}\left\{\delta_i\right\}_{i \in \mathcal{V}}, \\
\mathbf{y} &= \operatorname{col}\left\{y_i\right\}_{i \in \mathcal{V}} = \operatorname{col}\left\{P_i/\chi_i\right\}_{i \in \mathcal{V}}, \\
\mathbf{u} &= \operatorname{col}\left\{u_i\right\}_{i \in \mathcal{V}}, \\
\mathbf{d} &= \operatorname{col}\left\{P_{ii}/\chi_i\right\}_{i \in \mathcal{V}}.
\end{aligned}$$

Note that, as $Q_{ij} = Q_{ji}$, matrix $\mathbf{C} = \mathbf{C}' \in \mathbb{R}^{N \times N}$ is the Laplacian matrix of a weighted undirected graph without self loops. Consequently, the rows of this matrix always add up to zero (*i.e.* $\forall i \in \mathcal{V}$, $\sum_{j=1}^{N} [\mathbf{C}]_{ij} = 0$) and therefore $\mathbf{C}\mathbf{1} = \mathbf{0}$. That is, the described model for a microgrid is a CAN as studied in Section 5.2.1 with $\mathbf{A} = \mathbf{0}$, $\mathbf{B} = \kappa\mathbf{I}$, $q = 1$, and submitted to the action of an external perturbation $\mathbf{d}$ that represents the per unit load at each node.

Also note that the elements of matrix $\mathbf{C}$ are only dependent on the parameters of the electric lines but not on the loads. It is useful to define the per unit load change rate $\mathbf{w} = \frac{d}{dt}\mathbf{d}$.

6.2.3. Active Power Sharing

Active power sharing is understood as the property that, for any $i, j \in \mathscr{V}$, it holds true that (*e.g.*[181]):

$$\lim_{t\to+\infty} \frac{P_i(t)}{\chi_i} = \lim_{t\to+\infty} \frac{P_j(t)}{\chi_j}.$$

This objective remains naturally to consensus as defined in this thesis, and therefore the same methodology can be followed. That is, to define an organization matrix $\mathbf{T} = D'(\mathscr{T}^o) \in \mathbb{R}^{(N-1)\times N}$ from a strictly directed graph $\mathscr{T}^o$. From here, a target error vector can be defined as:

$$\mathbf{e} = \mathbf{T}\mathbf{y} \tag{6.6}$$

And power sharing can be redefined as the asymptotic convergence of this vector:

$$\text{Active Power sharing} \iff \lim_{t\to+\infty} \|\mathbf{e}\| = 0$$

6.2.4. Consensus Based Control

Consider a consensus algorithm

$$\mathbf{u} = \mathbf{L}\mathbf{y},$$

where $\mathbf{L} \in \mathbb{R}^{N\times N}$ is obtained as the negative Laplacian of an undirected weighted graph $\mathscr{G}_w$ with identically weighted selfloops:

$$\mathbf{L} = -L(\mathscr{G}_w) = -\hat{L}(\mathscr{G}_w) - l\mathbf{I},$$

with $l > 0$ a design constant. Note that,

$$\begin{aligned}\mathbf{L}\mathbf{T}^+\mathbf{T} &= -\hat{L}(\mathscr{G}_w)\mathbf{T}^+\mathbf{T} - l\mathbf{T}^+\mathbf{T} \\ &= -\hat{L}(\mathscr{G}_w) - l\mathbf{I} + l\mathbf{J} \\ &= \mathbf{L} + l\mathbf{J}\end{aligned}$$

Hence $\mathbf{L} = \mathbf{L}\mathbf{T}^+\mathbf{T} - l\mathbf{J}$. Therefore, the feedback signal $\mathbf{u}$ can be rewritten as:

$$\mathbf{u} = \mathbf{L}\mathbf{T}^+\mathbf{e} - l\mathbf{J}\mathbf{y}$$

From (6.5) and (6.6), it then follows that

$$\dot{\mathbf{e}} = \mathbf{T}\dot{\mathbf{y}} = \kappa\mathbf{T}\mathbf{F}\mathbf{C}\mathbf{L}\mathbf{T}^+\mathbf{e} - \kappa\frac{l}{N}\mathbf{T}\mathbf{F}(\mathbf{C}\mathbf{1})\mathbf{1}'\mathbf{y} + \mathbf{T}\mathbf{w}$$

Using the fact that $\mathbf{C1} = \mathbf{0}$, then finally

$$\dot{\mathbf{e}} = \kappa\mathbf{TFCLT}^{+}\mathbf{e} + \mathbf{Tw} \tag{6.7}$$

From here, it is possible to treat power sharing as the stability problem of system (6.7) under the action of the perturbation $\mathbf{w}$ in the same way as for a CAN with a looped algorithm. That is, consensus can be reached by considering algorithms derived of graphs with only selfloops and no connections between the nodes. In particular, the so called *droop controller* for active power sharing can be represented by $\mathbf{L} = -L((\mathscr{V},\mathscr{S},w))$, where $\mathscr{S} = \{(i,i)\}_{i\in\mathscr{V}}$ is a set of selfloops and the weight function $w : \mathscr{S} \mapsto \mathbb{R}^{+}$ is such that $\forall e \in \mathscr{S}$, $w(e) = l \in \mathbb{R}^{+}$. However, from equation (6.7), there is no mathematical reason to restrict the analysis only to algorithms without communication edges. That is, matrix $\mathbf{TFCLT}^{+}$ can be Hurwitz even without using a droop controller strategy.

As $\mathbf{w}$ is an external signal that depends only on the behavior of the load, the H_∞-norm of the transfer function matrix $H_{we}(s) = \left(s\mathbf{I} - \kappa\mathbf{TFCLT}^{+}\right)^{-1}\mathbf{T}$ between the normalized change rate of the load and the consensus error can be interpreted as a load change accuracy measurement and can be used to characterize the behavior of the grid. Similar to what has been done in the previous chapters, the following procedure can be stated to design algorithms that guaranteed certain performance of the network.

Theorem 6.1 (Active Power - Load Change Sensitivity)**.** *Given a microgrid modeled by equation (6.5) over the set of nodes $\mathscr{V}$, an unweighted undirected tree $\mathscr{T} = (\mathscr{V},\mathscr{E})$, a corresponding organization matrix $\mathbf{T} = D'(\mathscr{T}^{o}) \in \mathbb{R}^{(N-1)\times N}$, and scalar values $l > 0$ and $\gamma > 0$; a consensus algorithm described by $\mathbf{L} = -L((\mathscr{V},\mathscr{E}\cup\mathscr{S},w)) = -\mathbf{T}'\mathbf{WT} - l\mathbf{I}$, with a weight function defined by $w((i,j)) = w_{ij} > 0$ for $(i,j) \in \mathscr{E}$, and $w((i,i)) = w_{ii} = l$ for equally weighted selfloops in $\mathscr{S} = \{(i,i)\}_{i\in\mathscr{V}}$, and a guaranteed sensitivity under load change given by $\|H_{we}(s)\|_\infty \leqslant \gamma$, can be designed through the feasibility problem of LMI (6.8) over diagonal variable matrices $\mathbf{Q} > 0$ and $\mathbf{Z}$ of appropriate dimensions. In that case, $\mathbf{W} = \mathbf{ZQ}^{-1} = \mathrm{diag}\{w_{ij}\}_{(i,j)\in\mathscr{E}}$ represents the weights of the corresponding edges of the tree.*

$$\begin{bmatrix} -\kappa\mathbf{TFCT}'\mathbf{Z} - \kappa\mathbf{ZTCFT}' - l\kappa\mathbf{TFCT}^{+}\mathbf{Q} - l\kappa\mathbf{Q}(\mathbf{T}^{+})'\mathbf{CFT}' & \mathbf{T} & \mathbf{Q} \\ \star & -\gamma^2\mathbf{I} & \mathbf{0} \\ \star & \star & -\mathbf{I} \end{bmatrix} < 0 \tag{6.8}$$

Proof. Apply the BRL over system (6.7) and define $\mathbf{L} = -\mathbf{T}'\mathbf{WT} - l\mathbf{I}$ and $\mathbf{Z} = \mathbf{WQ}$. The diagonal structure of $\mathbf{W}$ is assured by the imposition that the variables $\mathbf{Z}$ and $\mathbf{Q} > 0$ are also diagonal. □

Note that additional structure conditions can be imposed to matrix $\mathbf{W}$. In particular, imposing certain diagonal elements of $\mathbf{Z}$ to be zero, results in forcing the corresponding edges to be weighted by zero. That is, to design algorithms that can be represented with less than $N-1$

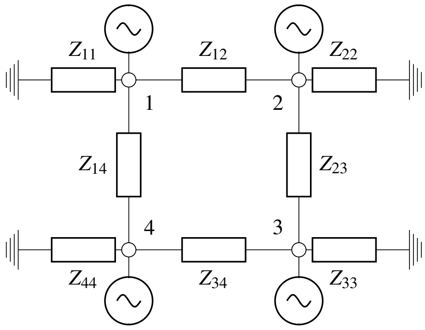


Figure 6.2.: Circuital Description of Microgrid in Example 6.1.

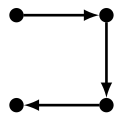


Figure 6.3.: Organization Strict Digraph $\mathscr{T}^o$ in Example 6.1.

edges between different agents. As the microgrid is a CAN, such a strategy can be followed and still reach consensus if $l \neq 0$.

In Theorem 6.1, it is considered that the weight value of the selfloops is known. However, inequality (6.8) can be modified to treat the value $l > 0$ also as a variable. We will not explicitly consider this case because typical microgrid setups work already with predefined droop controllers that take into account other practical and implementation issues. The objective of this result is then only to design information exchange schemes between the agents so that the performance of the network with respect to $\|H_{we}(s)\|_\infty$ can be improved.

Example 6.1. The objective of this simulation example is to evaluate the performance of different consensus algorithms under uncontrollable load changes. Consider the microgrid depicted in Figure 6.2. The nominal parameters of the lines and loads are given in Table 6.1 and the parameters of the inverters in Table 6.2. The load parameters are only an estimation of the nominal value and change with time. In the figures, the nodes in the graphs are labeled clockwise with node 1 at the upper left corner, like in Figure 6.2. The organization to be considered is given by Figure 6.3 and the corresponding matrix

$$\mathbf{T} = D'(\mathscr{T}^o) = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}.$$

We consider the consensus algorithms described by the undirected graphs in Figure 6.4. Controller a) represents simple droop control with gain $l = 1$. The next three controllers also

Table 6.1.: Line and Load Nominal Parameters for Microgrid in Example 6.1.

	$R_{ij}[\Omega]$	$L_{ij}[mH]$
Z_{12}	1.630	3.2900
Z_{23}	0.140	0.2540
Z_{34}	0.580	1.3400
Z_{14}	0.128	0.1324
Z_{11}	264.4689	276.6963
Z_{22}	211.7525	196.5919
Z_{33}	230.9099	184.2106
Z_{44}	198.9333	128.5817

Table 6.2.: Inverters Parameters for Example 6.1.

i	$S_i = \chi_i$[MVA]	$\tau_{p,i}$[s]	κ	V_{RMS}[kV]	f[Hz]
1	5.50	0.012	0.01	20	50
2	6.50	0.015			
3	6.30	0.011			
4	7.00	0.018			

include the same droop control but add links between the nodes. Controller b) uses the same edges as the graph that describes the microgrid. On the other hand, controller c) considers only the edges that are not part of the microgrid. Finally, controller d) considers all the possible edges between the nodes.

Additionally, we consider Theorem 6.1 with $l = 1$, and $\gamma = 0.10$ for e) and $\gamma = 0.25$ for f), to design the weighted graphs depicted in Figure 6.4 e) and f), where the labels over the edges represent their weights. Moreover, graphs g) and h) describe unweighted loopless algorithms that can also be used for consensus. Note that graph h) is not connected. The looped algorithms are characterized by matrices $\mathbf{L}_i = -L(\mathscr{G}_i) = -\mathbf{I} - \hat{L}(\mathscr{G}_i)$, $i \in \{a,b,c,d,e,f\}$, while the last two algorithms by $\mathbf{L}_i = -\hat{L}(\mathscr{G}_i)$, $i \in \{g,h\}$.

The evaluation of the algorithms considers the load profiles depicted in Figure 6.5. Here, the load active power demand and power factor are given for a period of 60 seconds at nominal frequency. In Table 6.3, the H$_\infty$-norm of the transfer function between the load change rate and the consensus error is shown along with the simulation indicators *ISD* and *IAD* at $t = 60s$ for the described load profiles. It is clear that the predictions given by $\|H_{we}(s)\|_\infty$ are corroborated in general by the values of *ISD* and *IAD*. Note that the performance of the feedbacks is better when the number of unweighted communication links is larger. Nevertheless,

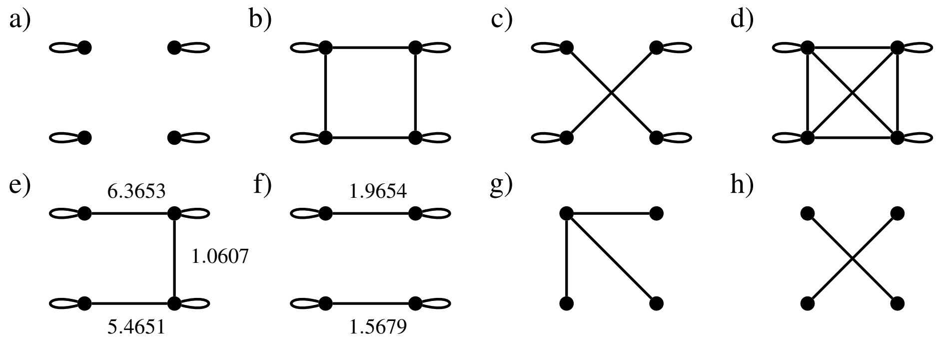


Figure 6.4.: Selflooped Graphs $\mathscr{G}_i$, with $i \in \{a,b,c,d,e,f,g,h\}$, for Algorithms in Example 6.1.

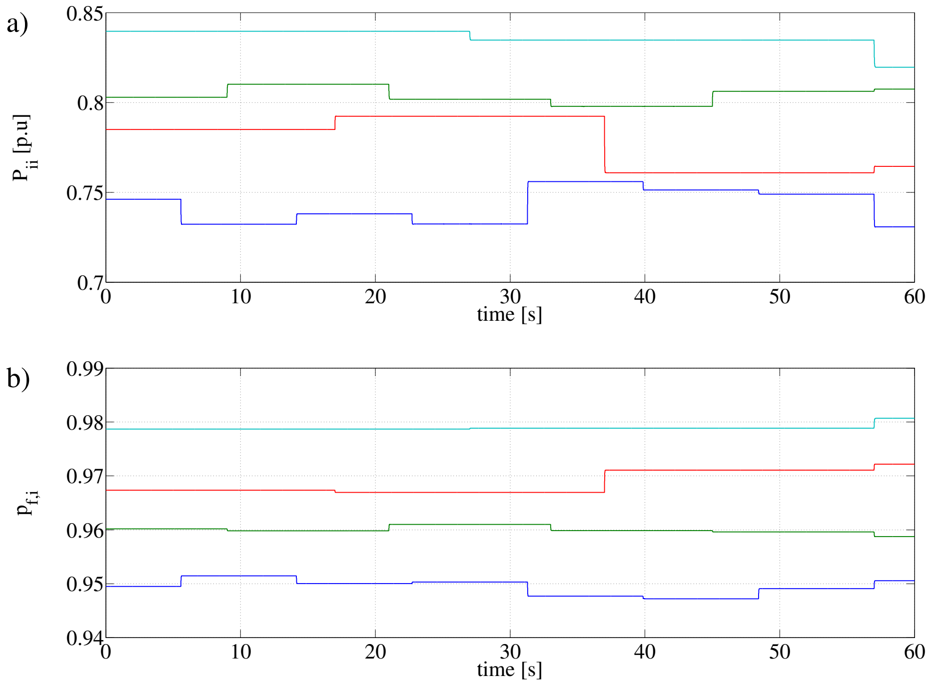


Figure 6.5.: Load behavior in Example 6.1. a) Per unit load active power $\bar{P}_{ii}$, b) Power factor $p_{f,i}$.

Table 6.3.: Numeric and Simulation Results in Example 6.1.

$\mathbf{L}_i$	$\|H_{we}(s)\|_\infty$	$ISD(60s)\times 10^3$	$IAD(60s)\times 10^3$
a)	0.7109	42.906810	82.200626
b)	0.2362	14.358624	27.533427
c)	0.2504	16.225617	28.489495
d)	0.1422	8.877602	16.828561
e)	0.0555	5.203841	8.868942
f)	0.1563	11.623881	19.131669
g)	0.5406	32.499430	57.306885
h)	∞	38.140610×10^2	76.261980×10^2

using the designed weighted feedback e), leads to a better performance than the fully connected unweighted feedback. Note that feedback f) has a similar performance as feedback d) even though it has only two communication edges. The loopless algorithm h), as it is not connected, shows the worst performance as, in that case matrix $\mathbf{TFCLT}^+$ is not Hurwitz because it has a zero eigenvalue. On the contrary, as algorithm g) is connected, it does indeed force the network to reach consensus. Its performance could be easily improved by considering appropriate weights or more links between the inverters.

The results in the table are complemented by Figure 6.6 where the output per unit powers of the inverters with the "best" and "worst" looped algorithms are drawn. When no links between the inverters are considered, the effect of load changes clearly compromise power sharing in a larger way. Figure 6.7 shows the response of the network with the loopless algorithms g) and h). It is clear that the non connected algorithm cannot ensure consensus. ■

6.3. Reactive Power Sharing

6.3.1. Control Strategy

Contrary to the case of active power where the power is controlled through the voltage frequency input, in the case of reactive power, control is done usually by varying the amplitude of the voltage. To simplify equation (6.2), a common assumption is that the angle differences are small so that $\sin(\delta_i(t)-\delta_j(t)) \approx \delta_i(t)-\delta_j(t) \approx 0$ and $\cos(\delta_i(t)-\delta_j(t)) \approx 1$. This can be relaxed to assume some possibly time variant difference without much changes in the following developments.

Moreover, it is assumed that the voltage amplitude of each inverter can only be modified around the nominal value V by a signal v_i so that $V_i = (1+v_i)V$ with $|v_i| \ll 1$. The maximum absolute value of the variable v_i depends on local regulations, however a realistic bound is

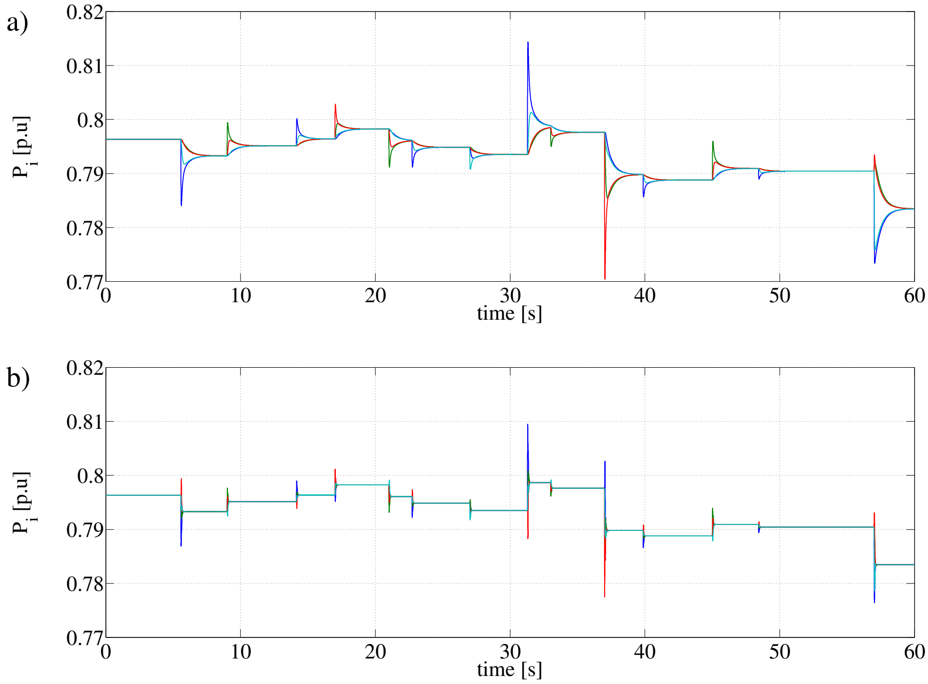


Figure 6.6.: Generated Active Power in [p.u.] in Example 6.1. a) Algorithm $\mathbf{L}_a$, b) Algorithm $\mathbf{L}_e$.

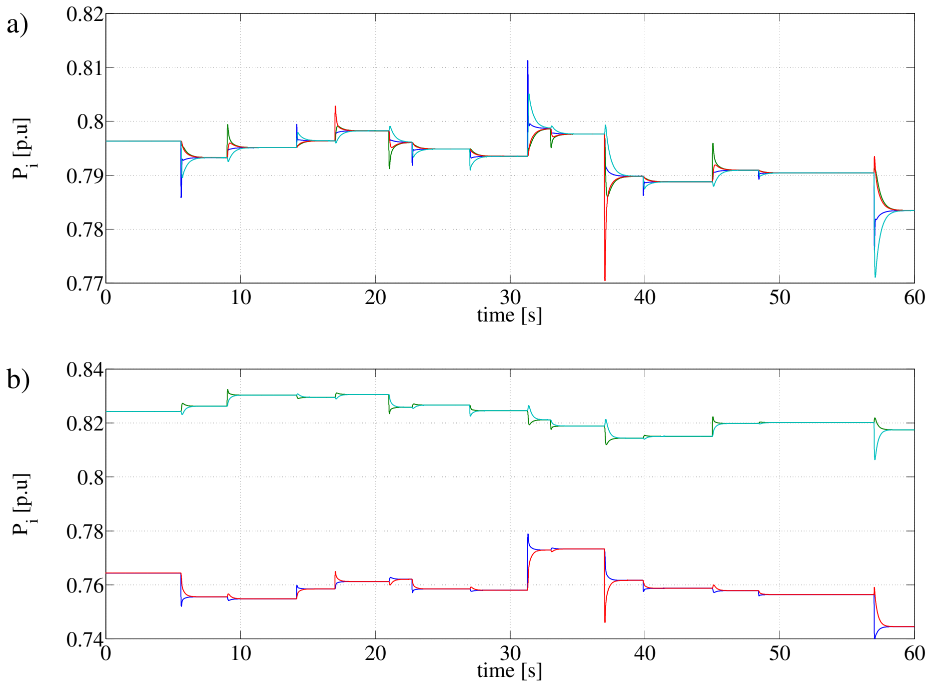


Figure 6.7.: Generated Active Power in [p.u.] in Example 6.1. a) Algorithm $\mathbf{L}_g$, b) Algorithm $\mathbf{L}_h$.

$|\nu_i| \leqslant 1.5\%$. With these simplifications, the reactive power expression can be approximated by a quadratic equation:

$$Q_i \approx Q_{ii}(1+\nu_i)^2 + \sum_{j\in\mathcal{N}_i} Q_{ij}\,(1+\nu_i)\,(\nu_i-\nu_j)\,, \tag{6.9}$$

where

$$Q_{ii} = \frac{3}{2}\frac{\omega L_{ii}}{R_{ii}^2+\omega^2 L_{ii}^2}V^2,\;\; Q_{ij} = \frac{3}{2}\frac{\omega L_{ij}}{R_{ij}^2+\omega^2 L_{ij}^2}V^2.$$

Note that, again, $Q_{ij} = Q_{ji}$ and that this approximation is only accurate if the electric angles ψ_i of every inverter change slowly around the nominal value ωt. Due to the fact that the differences between ν_i and ν_j are small, (6.9) is strongly dominated by the term $Q_{ii}(1+\nu_i)^2$. That is, in open loop, the reactive power injected at a node is mainly explained by the load consumption at the same node and the effect of the rest of the grid is low.

Expression (6.9) is clearly non linear. If the nominal values of load resistance $R_{ii,0}$ and load impedance $L_{ii,0}$ can be reliably estimated, and the load perturbation

$$\Delta Q_{ii} := Q_{ii} - Q_{ii,0} = Q_{ii} - \frac{3}{2}\frac{\omega L_{ii,0}}{R_{ii,0}^2+\omega^2 L_{ii,0}^2}V^2$$

is reasonably small, (6.9) can be linearized around $\nu_i = 0, \forall i \in \mathcal{V}$ to obtain the following affine expression:

$$Q_i \approx Q_{ii,0} + \Delta Q_{ii} + 2Q_{ii,0}\nu_i + \sum_{j\in\mathcal{N}_i} Q_{ij}\,(\nu_i-\nu_j)\,. \tag{6.10}$$

A *voltage droop control* strategy, understood as a proportional controller $u^{V_i} = V + k_{d,i}Q_i$, has been proposed [178, 180] to reach reactive power sharing imitating the strategy for active power sharing. However, it has been shown that this kind of control does not achieve this control objective [181]. This is because the injection of reactive power at a node is affected mainly by the load at that node and not by the effect of the neighbors of the node. That is why, following the approach in [181], we will consider an integral control strategy described by the following set of equations:

$$\begin{aligned} u^{V_i} &= V + w_i \\ \dot{w}_i &= V k_i u_i, \end{aligned}$$

where $k_i \in \mathbb{R}^+$ is a design parameter and u_i is a control signal that depends on the information of the rest of the inverters of the grid. This control strategy can be seen in Figure 6.8.

For the system composed of the inverter and its integral controller, we can define the state variable $x_i = \nu_i$. The controlled variable is the per unit reactive power $y_i = \frac{1}{\lambda_i}Q_i$ injected to the grid by the VSI. Here $\lambda_i > 0$ is the power sharing constant of each inverter. As in the active

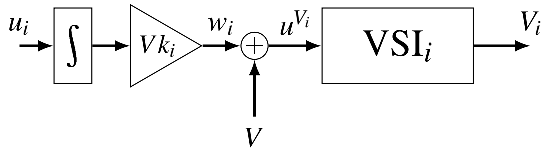


Figure 6.8.: VSI Voltage Amplitude Control.

power case, a practical choice of these constants would be $\lambda_i = \chi_i = S_i$ where S_i is the nominal power rating of the inverter at node $i \in \mathcal{V}$ [181]. However this is not a necessary condition and therefore we distinguish two potentially different constants, χ_i for active power and λ_i for reactive power. As $V_i = (1+\nu_i)V = (1+x_i)V$, a dynamic expression for the dynamics of the consensus agent are given $\forall i \in \mathcal{V}$ by

$$\begin{aligned} \dot{x}_i &= k_i u_i \\ y_i &= \frac{1}{\lambda_i} Q_{ii,0} + \frac{1}{\lambda_i}\Delta Q_{ii} + 2\frac{1}{\lambda_i} Q_{ii,0} x_i + \frac{1}{\lambda_i}\sum_{j\in\mathcal{N}_i} Q_{ij}\left(x_i - x_j\right), \end{aligned} \tag{6.11}$$

where ΔQ_{ii} plays the role of an external dynamical perturbation.

6.3.2. Aggregated Microgrid Reactive Power Control Model

As in the active power case, the whole network can be modeled as a connected network in the following compact way:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{F}\left(\mathbf{C} + 2\mathbf{D}_0\right)\mathbf{x} + \mathbf{d}_0 + \Delta\mathbf{d} \end{aligned} \tag{6.12}$$

Where the aggregated vectors and matrices are defined as:

$$\begin{aligned} &\mathbf{x} := \mathrm{col}\{x_i\}_{i\in\mathcal{V}} = \mathrm{col}\{\nu_i\}_{i\in\mathcal{V}}, && \mathbf{y} := \mathrm{col}\{y_i\}_{i\in\mathcal{V}} = \mathrm{col}\{Q_i/\lambda_i\}_{i\in\mathcal{V}}, && \mathbf{u} := \mathrm{col}\{u_i\}_{i\in\mathcal{V}}, \\ &\mathbf{B} := \mathrm{diag}\{k_i\}_{i\in\mathcal{V}}, && \mathbf{F} := \mathrm{diag}\{1/\lambda_i\}_{i\in\mathcal{V}}, && \\ &\mathbf{D}_0 := \mathrm{diag}\{Q_{ii,0}\}_{i\in\mathcal{V}}, && \Delta\mathbf{d} := \mathbf{F}\mathrm{col}\{\Delta Q_{ii}\}_{i\in\mathcal{V}}, && \mathbf{d}_0 := \mathbf{F}\mathbf{D}_0\mathbf{1}, \end{aligned}$$

and the matrix $\mathbf{C} = \mathbf{C}' \in \mathbb{R}^{N\times N}$ identical to the one of the active power case. That is,

$$[\mathbf{C}]_{ij} = \begin{cases} \sum_{k\in\mathcal{N}_i} Q_{ik} & \text{if } i = j, \\ -Q_{ij} & \text{if } j \in \mathcal{N}_i, \\ 0 & \text{i.o.c.} \end{cases}.$$

Note that this interconnected network is not a CAN, as defined in Section 5.2.1, nor an AAN but a mixture between both as matrix $\mathbf{C}+2\mathbf{D}_0$ is the Laplacian matrix of an undirected graph with selfloops. Also, note that $\mathbf{D}_0$ depends only on the nominal load. The per unit reactive power vector is defined as $\mathbf{d} = \mathbf{d}_0 + \Delta\mathbf{d}$ and its derivative over time, the per unit load change rate, as $\mathbf{w} = \frac{d}{dt}\mathbf{d} = \frac{d}{dt}\Delta\mathbf{d}$.

Reactive power sharing is defined analogously as active power sharing. That is, for all $i,j \in \mathscr{V}$,

$$\lim_{t\to+\infty} \frac{Q_i(t)}{\lambda_i} = \lim_{t\to+\infty} \frac{Q_j(t)}{\lambda_j}$$

This is equivalent to the convergence to the origin of a consensus error $\mathbf{e} = \mathbf{T}\mathbf{y}$ defined, as usual, through an organization matrix $\mathbf{T} = D'(\mathscr{T}^o) \in \mathbb{R}^{(N-1)\times N}$ derived from a strictly directed graph $\mathscr{T}^o$:

$$\text{Reactive Power sharing} \iff \lim_{t\to+\infty} \|\mathbf{e}\| = 0$$

Where, $\mathbf{e} = \mathbf{T}\mathbf{y}$.

6.3.3. Consensus Based Control

As usual, a consensus algorithm for the described network is defined as a proportional feedback:

$$\mathbf{u} = \mathbf{L}\mathbf{y}.$$

Self-looped Controller

A selflooped controller considers only the reactive power measurement of each inverter for feedback. That is, without sharing signals between the inverters. A controller in such a way realizes the negative Laplacian matrix of an undirected graph, where the only edges are equally weighted selloops. This kind of controller can be addressed as *voltage droop controller* because of the analogy with the frequency droop controller. In this section, we briefly show why this kind of controllers should be avoided by considering the quadratical model of the agents. Without loss of generality, the consensus algorithm matrix may take the form $\mathbf{L} = -\mathbf{I}$.

Considering the quadratical model for the reactive power, with state variable $x_i = v_i$ and output variable $y_i = \frac{1}{\lambda_i} Q_i$, each inverter can be characterized by:

$$\begin{aligned} \dot{x}_i &= k_i u_i \\ y_i &= \frac{1}{\lambda_i} Q_{ii} (1+x_i)^2 + \frac{1}{\lambda_i} \sum_{j\in\mathscr{N}_i} Q_{ij} (1+x_i)(x_i - x_j) \end{aligned}$$

Then, the microgrid as a quadratic MIMO system can be written in compact way as

$$\dot{\mathbf{x}} = \mathbf{B}\mathbf{u}$$
$$\mathbf{y} = \mathbf{F}\mathbf{D}\,(\mathbf{I}+\mathbf{X})^2\,\mathbf{1} + \mathbf{F}(\mathbf{I}+\mathbf{X})\mathbf{C}\mathbf{X}\mathbf{1}$$

Where $\mathbf{X} = \operatorname{diag}\{x_i\}_{i\in\mathscr{V}}$ and $\mathbf{D} = \operatorname{diag}\{Q_{ii}\}_{i\in\mathscr{V}}$. Note that $\mathbf{d} = \mathbf{F}\mathbf{D}\mathbf{1}$ and $\mathbf{x} = \mathbf{X}\mathbf{1}$. After feedback, the dynamics of the system become:

$$\dot{\mathbf{x}} = \mathbf{B}\mathbf{L}\mathbf{F}\mathbf{D}\,(\mathbf{I}+\mathbf{X})^2\,\mathbf{1} + \mathbf{B}\mathbf{L}\mathbf{F}(\mathbf{I}+\mathbf{X})\mathbf{C}\mathbf{X}\mathbf{1}$$

Considering that $|x_i| \ll 1$, then $\mathbf{I}+\mathbf{X} \approx \mathbf{I}$ and so

$$\dot{\mathbf{x}} \approx \mathbf{B}\mathbf{L}\mathbf{F}\,(\mathbf{D}+\mathbf{C}\mathbf{X})\,\mathbf{1}$$

Due to the fact that the differences between x_i and x_j are small, then the term $\mathbf{C}\mathbf{X}\mathbf{1}$ can be neglected when compared to $\mathbf{D}\mathbf{1}$. In that case:

$$\dot{\mathbf{x}} \approx \mathbf{B}\mathbf{L}\mathbf{F}\mathbf{D}\mathbf{1} \tag{6.13}$$

Replacing the self looped controller and considering that $\mathbf{B}$, $\mathbf{F}$ and $\mathbf{D}$ are diagonal positive definite matrices, the following inequality holds element-wise:

$$\dot{\mathbf{x}} \approx -\mathbf{B}\mathbf{d} < \mathbf{0}_{N\times 1}$$

That is, when a selflooped controller is considered and in the neighborhood of the operation point $|x_i| \ll 1$, the node voltage is constantly decreasing with a speed proportional to the reactive load. This drives the system away from the operation point what is not acceptable for regulation reasons. In this sense, a selflooped controller should be avoided and other strategies must be used.

Loopless Laplacian Controller

Consider that the consensus feedback is given as the negative Laplacian matrix of an undirected weighted loopless graph $\mathscr{G}_w$, *i.e.*,

$$\mathbf{L} = -\hat{L}(\mathscr{G}_w).$$

Note that for this kind of consensus algorithms, equation (6.13) does not imply the negativity of the node voltage change rate, as this speed depends on the normalized difference between the reactive loads and because $\mathbf{L}$ has at least one zero eigenvalue.

Because of the zero row sum property of the Laplacian matrix, the feedback signal $\mathbf{u}$ can be rewritten as

$$\mathbf{u} = \mathbf{L}\mathbf{T}^{+}\mathbf{e}.$$

From here, the dynamics of the error using the affine dynamic model of the microgrid can be written as

$$\dot{\mathbf{e}} = \mathbf{TF}\left(\mathbf{C}+2\mathbf{D}_0\right)\mathbf{BLT}^{+}\mathbf{e}+\mathbf{Tw}. \tag{6.14}$$

Reactive power sharing can then be treated as the stability problem of the linear system (6.14) under the influence of the external perturbation $\mathbf{w}$. When the operating point is either not precisely known, or the consensus algorithm is studied over (infinitely) many operating points within a range, this can be modeled by $\mathbf{D}_0 = \mathbf{M}+\mathbf{NU}$ with $\mathbf{M}$ and $\mathbf{N}$ precisely known diagonal matrices and $\mathbf{U}$ a diagonal matrix such that $\mathbf{U}^2 \leqslant \varepsilon^2\mathbf{I}$, with $\varepsilon > 0$ a given scalar.

We can interpret the H_∞-norm of the transfer function matrix

$$H_{we}(s) = \left(s\mathbf{I}-\mathbf{TF}\left(\mathbf{C}+2\mathbf{D}_0\right)\mathbf{BLT}^{+}\right)^{-1}\mathbf{T}$$

between $\mathbf{w}$ and $\mathbf{e}$, respectively its least upper bound γ_{min} over all diagonal $\mathbf{U}$ satisfying $\mathbf{U}^2 \leqslant \varepsilon^2\mathbf{I}$ for a given $\varepsilon > 0$, as an indicator of load change accuracy. This is expressed in the following theorem.

Theorem 6.2 (Reactive Power - Load Change Sensitivity)**.** *In a microgrid described by equation (6.12) with uncertainties over the loads described by $\mathbf{D}_0 = \mathbf{M}+\mathbf{NU}$ with $\mathbf{M}$ and $\mathbf{N}$ precisely known diagonal matrices and $\mathbf{U}$ a diagonal matrix such that $\mathbf{U}^2 \leqslant \varepsilon^2\mathbf{I}$, with $\varepsilon > 0$, for given $\mathbf{T} = D'(\mathscr{T}^o)$ and $\mathbf{L} = -\hat{L}(\mathscr{G}_w)$, the sensitivity of the reactive power sharing control law with respect to load changes can be characterized by a scalar value $\gamma_{min} = \sqrt{\mu_{min}} > 0$ that can be computed by solving the optimization problem*

$$\mu_{min} = \inf\left\{\mu \in \mathbb{R}^{+} \mid LMI\ (6.15)\ \wedge\ \mathbf{Q}=\mathbf{Q}' > 0 \wedge\ \alpha > 0\right\}.$$

$$\left[\begin{array}{c|cc|c} \begin{array}{c}\mathbf{TF}\left(\mathbf{C}+2\mathbf{M}\right)\mathbf{BLT}^{+}\mathbf{Q}+\mathbf{Q}(\mathbf{T}^{+})'\mathbf{LB}\left(\mathbf{C}+2\mathbf{M}\right)\mathbf{FT}'+\\ 4\varepsilon^2\alpha\mathbf{TF}^2\mathbf{N}^2\mathbf{T}'\end{array} & \mathbf{T} & \mathbf{Q} & \mathbf{Q}(\mathbf{T}^{+})'\mathbf{LB} \\ \hline \star & -\mu\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \star & \star & -\mathbf{I} & \mathbf{0} \\ \hline \star & \star & \star & -\alpha\mathbf{I} \end{array}\right] < 0 \tag{6.15}$$

Proof. Applying the BRL to system (6.14) leads to the following matrix inequality:

$$\begin{bmatrix} \mathbf{TF}\left(\mathbf{C}+2\mathbf{D}_0\right)\mathbf{BLT}^{+}\mathbf{Q}+\mathbf{Q}(\mathbf{T}^{+})'\mathbf{LB}\left(\mathbf{C}+2\mathbf{D}_0\right)\mathbf{FT}'+\mathbf{QQ} & \mathbf{T} \\ \star & -\mu\mathbf{I} \end{bmatrix} < 0 \tag{6.16}$$

with $\mu = \gamma^2$. Inserting $\mathbf{D}_0 = \mathbf{M}+\mathbf{NU}$, and considering the Cross Products proposition A.15, then the uncertain terms related to matrix $\mathbf{U}$ can be bounded to get an upper bound for inequality (6.16):

$$\left[\begin{array}{c|c} \begin{array}{c}\mathbf{TF}\left(\mathbf{C}+2\mathbf{M}\right)\mathbf{BLT}^{+}\mathbf{Q}+\mathbf{Q}(\mathbf{T}^{+})'\mathbf{LB}\left(\mathbf{C}+2\mathbf{M}\right)\mathbf{FT}'\\ +\mathbf{QQ}+4\varepsilon^2\alpha\mathbf{TFNNFT}'+\frac{1}{\alpha}\mathbf{Q}(\mathbf{T}^{+})'\mathbf{LBBLT}^{+}\mathbf{Q}\end{array} & \mathbf{T} \\ \hline \star & -\mu\mathbf{I} \end{array}\right] < 0.$$

Table 6.4.: Inverters Parameters in Example 6.2.

i	$S_i = \lambda_i$[MVA]	k_i[1/s]	$\tau_{V,i}$[s]	$\tau_{q,i}$[s]
1	5.50	0.0022	0.011	0.012
2	6.50	0.0018	0.013	0.015
3	6.30	0.0024	0.012	0.011
4	7.00	0.0016	0.016	0.018

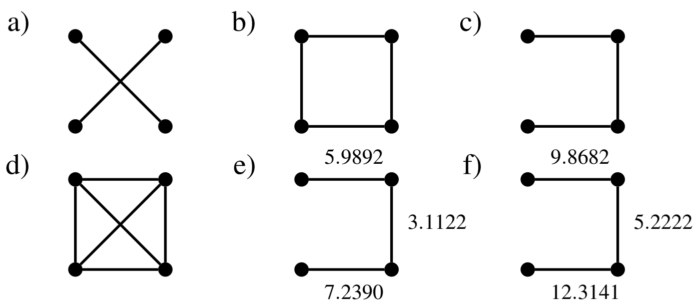


Figure 6.9.: Loopless Graphs $\mathscr{G}_i$, with $i \in \{a,b,c,d,e,f\}$, for Algorithms in Example 6.2.

Applying Schur's complement on the quadratic terms, we finally obtain LMI (6.15). □

Note that LMI (6.15) can be used, as in the active power case, to design tree shaped algorithms by imposing $\mathbf{L} = -\mathbf{T}'\mathbf{W}\mathbf{T}$, with $\mathbf{W} \in \mathbb{R}^{N\times N}$ diagonal, defining an auxiliary linearization variable $\mathbf{Z} = \mathbf{W}\mathbf{Q} \iff \mathbf{W} = \mathbf{Z}\mathbf{Q}^{-1}$ and imposing $\mathbf{Z}$ and $\mathbf{Q} > 0$ to be diagonal.

Example 6.2. Consider the same microgrid as in Example 6.1 with the same load and line nominal parameters. The additional parameters of the inverters are given in Table 6.4. It is also known that the operation point given by the load in the table can vary within a range of $\pm 3\%$. That is, $\mathbf{M} = \mathbf{N} = \operatorname{diag}\{Q_{ii,0}\}_{i\in\mathscr{V}_M}$ and $\varepsilon = 0.03$. The study of consensus is done through the same organization as in Example 6.1.

We consider the four arbitrary algorithms corresponding to the undirected loopless graphs in Figure 6.9 a), b), c) and d). Additionally, through LMI (6.15), the tree shaped algorithms associated with the weighted graphs depicted in Figure 6.4 e) and f) are designed with $\mu = \gamma^2 = 0.50$ for e) and $\mu = \gamma^2 = 0.30$ for f). The corresponding feedback matrices are given by $\mathbf{L}_i = -L(\mathscr{G}_i)$, $i \in \{a,b,c,d,e,f\}$.

The evaluation of the algorithms is done under the exactly same conditions as in Example 6.1. In Table 6.5 the values obtained by Theorem 6.2 are depicted along with the simulation indicators *ISD* and *IAD* at $t = 60s$. Similar conclusions as with the active power case can be drawn. The table is complemented by Figure 6.10 where the injected reactive power in p.u. of selected cases is shown. From Figure 6.10 b) and c), it is clear that the weighted control law

Table 6.5.: Numeric and Simulation Results in Example 6.2.

$\mathbf{L}_i$	γ_{min}	$ISE(60s)\times 10^3$	$IAE(60s)\times 10^3$
a)	–	14.0622×10^2	28.0261×10^2
b)	1.8271	59.8256	114.3103
c)	1.8729	70.5439	123.2655
d)	0.9186	33.2349	63.1663
e)	0.3148	14.2585	24.2328
f)	0.1876	9.6138	16.2864

f) behaves better than the fully connected but unweighted algorithm d).

Note that the performance of feedback a) cannot be evaluated by the indicators as the graph related to the controller is not connected and because the dynamics of the reactive power injected at each inverter are strongly decoupled. Indeed, in Figure 6.10 a), even though changes in any load are reflected as perturbations for other nodes, their influence is not enough to strongly modify the output power of all inverters. Only the connected nodes of the graph reach this goal regardless of the structure of the grid. This differs greatly from the case of active power sharing where the coupling between each node through the grid is large enough to ensure power sharing. ■

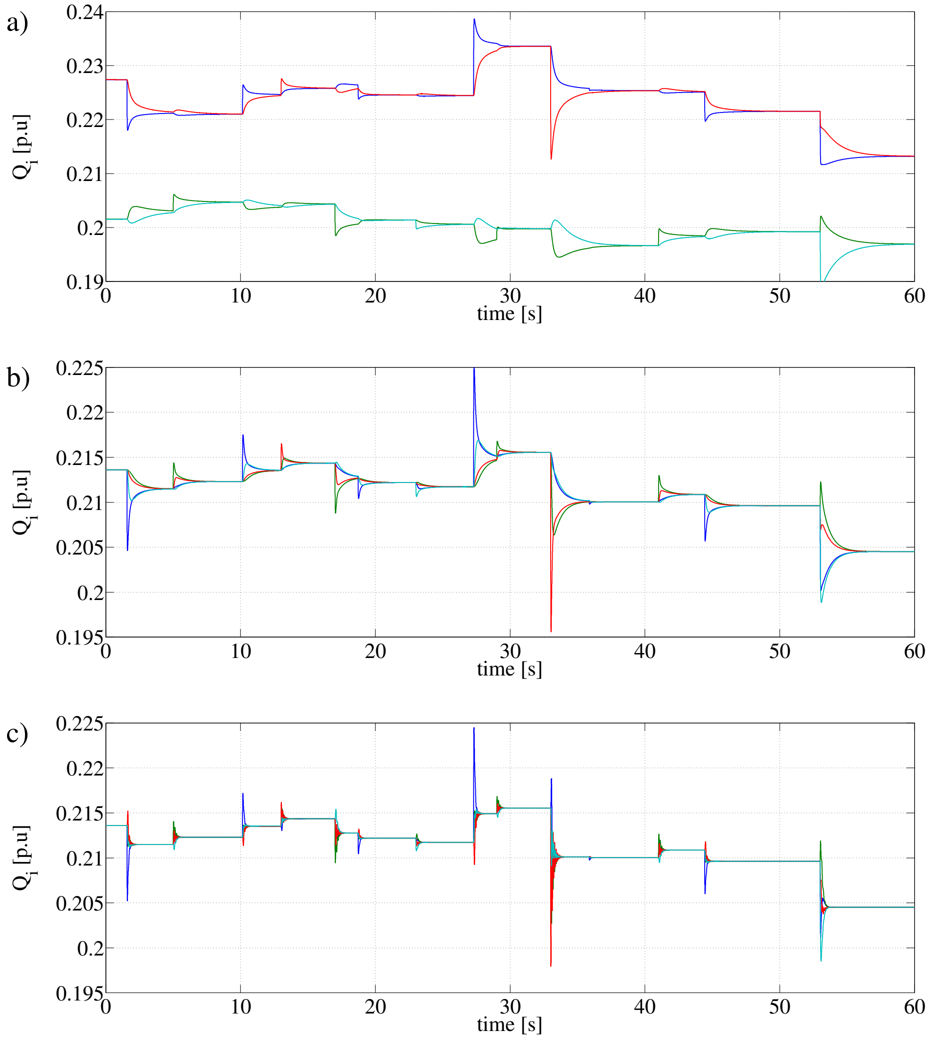


Figure 6.10.: Generated Reactive Power [p.u.] in Example 6.2. a) Algorithm $\mathbf{L}_a$, b) Algorithm $\mathbf{L}_d$, c) Algorithm $\mathbf{L}_f$.

7 Formation Control

The themes of consensus and formation control in multiagents systems are strongly related and are usually studied together like in [22, 29]. Most of the interest in the formation problem is focused on applications in mobile vehicles and robots, *e.g.* [3, 8, 11, 12, 24, 39, 45], where the dynamical models of the systems are derived from movement laws and leader-follower strategies are used in a mostly application oriented way. The link between formation control and the consensus problem is suggested through the use of graph theoretical methods in papers as [7, 42, 44]. However, most of the theoretically oriented approach to the formation problem is done by means of consensus analysis on networks composed solely by double integrators systems, [1, 13, 16, 27, 30, 32, 62]. This special kind of dynamics ease the consensus analysis on the networks greatly, but are difficult to generalize into more complex scenarios.

In general, the formation problem as understood here deals with a network where the agents do not search for consensus over their outputs (which can be interpreted as the agents' "velocity"), but over the integral of the outputs (which can be interpreted as the agents' "position"). Furthermore, they usually do not aim to agree on the exact same position, what in a mobile robots setup for example would not be physically possible, but on a given pattern or "formation". Implicitly, it is assumed that the integration process is independent of the internal characteristics of the agents, in the same sense that, for example, the position of a vehicle on a plane depends on its instantaneous speed, but not on the internal dynamic process to drive at this speed. That is, the position of the vehicle is independent of if it has one or several engines, if these are electric or combustion drives, or any technological characteristic of the individual system. Furthermore, one could exchange one vehicle by other with completely different dynamics, and the integration process would remain the same. From this perspective, the agents can be interpreted as the actuators for an integration process.

The most direct example of such formation problem are mobile robots commanded to move from one point to other maintaining a circular, linear or triangular formation. The active power sharing problem studied in Chapter 6 could also be considered as a formation problem, as consensus is not searched on the frequency space of the inverters but on a function of the electrical angle which is the integral of the frequency.

7.1. The Formation Control Problem

Consider N autonomous agents with aggregated dynamics given by the usual representation:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x}. \end{aligned} \tag{7.1}$$

Where $\mathbf{A} = \operatorname{diag}\{\mathbf{A}_i\}_{i\in\mathscr{V}} \in \mathbb{R}^{n\times n}$, and matrices $\mathbf{B} = \operatorname{diag}\{\mathbf{B}_i\}_{i\in\mathscr{V}} \in \mathbb{R}^{n\times p}$ and $\mathbf{C} = \operatorname{diag}\{\mathbf{C}_i\}_{i\in\mathscr{V}} \in \mathbb{R}^{Nq\times n}$ are full rank. We are also interested in the integral expression

$$\mathbf{z}_i = \int_{t_0}^{t} \mathbf{y}_i dt.$$

Clearly, with $\mathbf{z} = \operatorname{col}\{\mathbf{z}_i\}_{i=1}^{N}$ and $\mathbf{y}(t_0) = \mathbf{0}$,

$$\dot{\mathbf{z}} = \mathbf{y}. \tag{7.2}$$

We assume that $\mathbf{z}$ cannot be directly measured and therefore is not available for feedback. Only relative information of the differences between each $\mathbf{z}_i$ is available.

Because of the immediate comparison with an agent moving in a two or three dimensional space, the additional states $\mathbf{z}$ can be interpreted as the “position” of the agents, while the output $\mathbf{y}$ can be considered their “velocity”. This does however not mean that the study is restricted to only such a problem. Note that $\mathbf{z}_i$ can only be controlled by modifying $\mathbf{y}_i$ through the input $\mathbf{u}_i$ and so, in that sense, the agent is an “actuator” for the position process.

Consider the network described by (7.1) and (7.2). If this network is small in the sense of what was defined in Section 4.1.2, it can be analyzed from a centralized perspective considering information on all the elements of the network. The objective of the controller is to ensure that the agents can reach a given distribution in the space where vector $\mathbf{z}$ is defined. We assume that a centralized relative measurement of the vector $\mathbf{z}$ is available and defined through an organization transformation:

$$\mathbf{e}_0 = \mathbf{T}\mathbf{z}. \tag{7.3}$$

With $\mathbf{T} = D'(\mathscr{T}^o) \in \mathbb{R}^{(N-1)q\times Nq}$. As $\mathbf{T}^{+}\mathbf{e}_0 = \mathbf{T}^{+}\mathbf{T}\mathbf{z} = \mathbf{z} - \mathbf{J}\mathbf{z}$,

$$\mathbf{z} = \mathbf{T}^{+}\mathbf{e}_0 + \mathbf{J}\mathbf{z}. \tag{7.4}$$

Observe that $\mathbf{J}\mathbf{z} = \mathbf{1}E(\mathbf{z})$, where $E(\mathbf{z}) = \frac{1}{N}\sum_{i\in\mathscr{V}} \mathbf{z}_i$ is the unknown mean value of all $\mathbf{z}_i$.

In principle, the following analysis of a network can be done with any arbitrary organization matrix and not necessarily with the one that defines the relative measurements vector $\mathbf{e}_0$. Nevertheless, for simplicity, we assume in this section that the analysis matrix is the same as the one that defines the error.

It is of interest for the central controller that the systems reach in stationary state a fixed formation in $\mathbb{R}^q$ described by a known set of relative differences $\mathbf{p} = \mathbf{T}\bar{\mathbf{z}} \in \mathbb{R}^{(N-1)q}$, where $\bar{\mathbf{z}} = \mathrm{col}\{\bar{\mathbf{z}}_i\}_{i\in\mathscr{V}}$, with $\bar{\mathbf{z}}_i \in \mathbb{R}^q$, is an unknown vector that describes the desired formation of a network in the unknown absolute reference frame.

From here, similar as with the consensus objective, the formation problem can be defined as the stability of the difference between the relative measurements and the reference position.

Definition 7.1.1. A network is said to reach a desired *formation*, described by $\mathbf{p}$, if the vector

$$\mathbf{e} = \mathbf{e}_0 - \mathbf{p},$$

asymptotically approaches zero. That is,

$$\text{Formation} \iff \lim_{t\to+\infty} \|\mathbf{e}\| = 0.$$

Note that when $\mathbf{e} = \mathbf{T}\mathbf{z} - \mathbf{p} = \mathbf{0}$ we have that $\mathbf{z} = \mathbf{T}^{+}\mathbf{p} + \mathbf{J}\mathbf{z}$. From here, when $\mathbf{p}$ is constant, $\dot{\mathbf{z}} = \mathbf{J}\dot{\mathbf{z}}$. Therefore $\mathbf{y} = \dot{\mathbf{z}} = \mathbf{J}\mathbf{y} \iff \mathbf{y} = \mathbf{1}\mathbf{v}$, for some $\mathbf{v} \in \mathbb{R}^q$. That is, reaching formation implies that the agents reach consensus on their outputs.

We are further interested in imposing that, in absence of external inputs, the states of the agents, $\mathbf{x}$, asymptotically approaches the origin. That is,

$$\lim_{t\to+\infty} \|\mathbf{x}\| = 0.$$

This requirement is in principle a particular and restrictive case, however it does make sense when the agents are human-made physical systems. Indeed, it is expectable that, in open loop, the agents have stable dynamics. In closed loop, this second objective must be imposed if we aim to further control the behavior of the network as a whole once formation is achieved. If it were not the case, simple additional objectives as moving all the agents, while keeping the formation, from one region to other, would not be easy as the unbounded states might reflex on the outputs.

7.2. Control Strategy

7.2.1. Central Control Strategy Analysis

Consider the following given feedback law

$$\mathbf{u} = \mathbf{L}\mathbf{y} + \mathbf{H}(\mathbf{e}_0 - \mathbf{p}) + \mathbf{u}_{ext}, \tag{7.5}$$

with known feedback matrices $\mathbf{L} \in \mathbb{R}^{Nq\times Nq}$ and $\mathbf{H} \in \mathbb{R}^{Nq\times(N-1)q}$. Signal $\mathbf{u}_{ext}$ is an external input used to arbitrary modify the velocity of the agents and will be treated as a perturbation for the formation objective. This control strategy assumes that signals $\mathbf{u}_i = \mathbf{s}_i'\mathbf{u}$, available

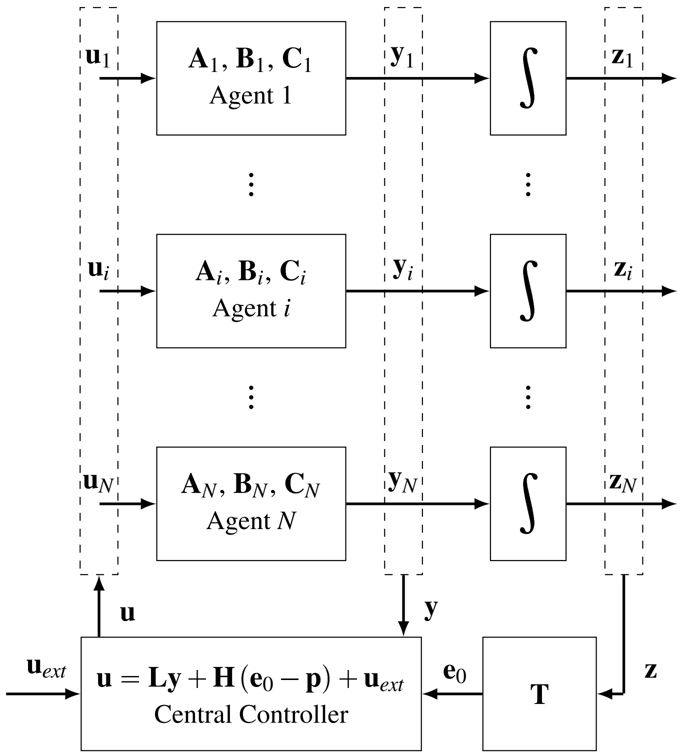


Figure 7.1.: Centralized Formation Control Strategy.

at the central controller level, are communicated to each agent. Furthermore, the outputs of the systems and the external inputs need also to be communicated to the central instance. A schematic representation of this control strategy can be seen in Figure 7.1. Note that the need of communication is implicitly necessary in this case.

If we want to avoid the active exchange of signals between the agents, we can assume that the matrix feedback $\mathbf{L}$ is block diagonal so that there is a feedback of $\mathbf{y}_i$ from each agent only to itself. However, any general matrix can be considered. In particular, a Laplacian structure derived from a weighted graph $\mathscr{G}_w$ can also be used.

With this control law, the dynamics of the network and the error can be written jointly as:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{e}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}+\mathbf{BLC} & \mathbf{BH} \\ \mathbf{TC} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{e} \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \mathbf{u}_{ext} + \begin{bmatrix} \mathbf{0} \\ -\mathbf{I} \end{bmatrix} \mathbf{w},$$

with $\mathbf{w} = \dot{\mathbf{p}}$. Defining the following matrices and vectors,

$$\tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ \mathbf{e} \end{bmatrix}, \tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{TC} & \mathbf{0} \end{bmatrix}, \tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}, \tilde{\mathbf{B}}_w = \begin{bmatrix} \mathbf{0} \\ -\mathbf{I} \end{bmatrix},$$

$$\tilde{\mathbf{C}} = \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \bar{\mathbf{C}} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix}, \tilde{\mathbf{L}} = \begin{bmatrix} \mathbf{L} & \mathbf{H} \end{bmatrix},$$

the expression can be written more compactly as

$$\begin{aligned} \dot{\tilde{\mathbf{x}}} &= \left(\tilde{\mathbf{A}} + \tilde{\mathbf{B}}\tilde{\mathbf{L}}\tilde{\mathbf{C}}\right)\tilde{\mathbf{x}} + \tilde{\mathbf{B}}\mathbf{u}_{ext} + \tilde{\mathbf{B}}_w\mathbf{w} \\ \mathbf{e} &= \bar{\mathbf{C}}\tilde{\mathbf{x}} \end{aligned} \tag{7.6}$$

Clearly, the dynamic behavior of $\tilde{\mathbf{x}}$ depends on the eigenvalues of matrix

$$\tilde{\mathbf{A}}_{cl} := \tilde{\mathbf{A}} + \tilde{\mathbf{B}}\tilde{\mathbf{L}}\tilde{\mathbf{C}} \in \mathbb{R}^{(n+(N-1)q)\times(n+(N-1)q)}.$$

Therefore, for known matrices $\mathbf{L}$ and $\mathbf{H}$, and in absence of external inputs, simply calculating the eigenvalues of $\tilde{\mathbf{A}}_{cl}$ is enough to determine if the network can reach the specified formation and the states converge to the origin. Observe that the lower right block of matrix $\tilde{\mathbf{A}}_{cl}$ is always zero.

When $\tilde{\mathbf{A}}_{cl}$ is Hurwitz, the H_∞-norm of the transfer function matrix from $\mathbf{u}_{ext}$ to $\mathbf{e}$, $\gamma_{ext} = \|H_{ext}(s)\|_\infty$, can be interpreted as an indicator of formation sensitivity to external input changes. Similarly, the sensitivity to reference changes can be evaluated by the norm of the transfer function matrix from $\mathbf{w}$ to $\mathbf{e}$, $\gamma_w = \|H_{we}(s)\|_\infty$.

7.2.2. Relative Position Feedback Design

In the case where the feedback matrices are to be designed, the problem is in the form of an output feedback design problem as those in, for example, [83, 89, 91], and stability conditions in terms of linear matrix inequalities (LMI) and Lyapunov functions can be stated.

Determining H with known L.

With the matrices defined for system (7.6), the problem of design of the feedback matrices $\mathbf{L}$ and $\mathbf{H}$ such that $\tilde{\mathbf{A}}+\tilde{\mathbf{B}}\tilde{\mathbf{L}}\tilde{\mathbf{C}}$ is Hurwitz can be solved with the help of Theorem A.17 in the Appendix. The resulting procedure is however difficult to apply when structural restrictions on $\mathbf{L}$ are assumed. Therefore, we will first assume that this matrix is known and block diagonal and that $\mathbf{A}_{cl} := \mathbf{A}+\mathbf{BLC}$ is Hurwitz. In that way, we can concentrate on determining $\mathbf{H}$ only.

When $\mathbf{w}=\mathbf{0}$, the system can be rewritten as

$$\begin{aligned}\dot{\tilde{\mathbf{x}}} &= \left(\bar{\mathbf{A}}+\tilde{\mathbf{B}}\mathbf{H}\bar{\mathbf{C}}\right)\tilde{\mathbf{x}}+\tilde{\mathbf{B}}\mathbf{u}_{ext}\\ \mathbf{e} &= \bar{\mathbf{C}}\tilde{\mathbf{x}},\end{aligned} \tag{7.7}$$

with

$$\tilde{\mathbf{x}} = \begin{bmatrix}\mathbf{x}\\ \mathbf{e}\end{bmatrix},\ \bar{\mathbf{A}} = \begin{bmatrix}\mathbf{A}_{cl} & \mathbf{0}\\ \mathbf{TC} & \mathbf{0}\end{bmatrix},\ \tilde{\mathbf{B}} = \begin{bmatrix}\mathbf{B}\\ \mathbf{0}\end{bmatrix},\ \bar{\mathbf{C}} = \begin{bmatrix}\mathbf{0} & \mathbf{I}_{(N-1)q}\end{bmatrix}.$$

This leads to the following design procedure for $\mathbf{H}$, which is based in the results stated in [89, 91] and the course [103].

Theorem 7.1. *For a given scalar $\gamma>0$ and a given known matrix $\mathbf{R}\in\mathbb{R}^{n\times(N-1)q}$ such that the composed matrix*

$$\begin{bmatrix}\bar{\mathbf{C}}\\ \bar{\mathbf{R}}\end{bmatrix} := \begin{bmatrix}\mathbf{0} & \mathbf{I}\\ \mathbf{I} & \mathbf{R}\end{bmatrix} \in \mathbb{R}^{(n+(N-1)q)\times(n+(N-1)q)}$$

is full rank, if there are matrices $\mathbf{Q}_1>0$, $\mathbf{Q}_2>0$, and $\mathbf{Z}$ of proper dimensions such that

$$\tilde{\mathbf{Q}} := \begin{bmatrix}\mathbf{Q}_1 & -\mathbf{R}\mathbf{Q}_2\\ \star & \mathbf{Q}_2\end{bmatrix} > 0, \tag{7.8}$$

$$\begin{bmatrix}\bar{\mathbf{A}}\tilde{\mathbf{Q}}+\tilde{\mathbf{Q}}\bar{\mathbf{A}}'+\tilde{\mathbf{B}}\bar{\mathbf{Z}}+\bar{\mathbf{Z}}'\tilde{\mathbf{B}}' & \tilde{\mathbf{B}} & \tilde{\mathbf{Q}}\bar{\mathbf{C}}'\\ \star & -\gamma^2\mathbf{I} & \mathbf{0}\\ \star & \star & -\mathbf{I}\end{bmatrix} < 0, \tag{7.9}$$

with $\bar{\mathbf{Z}} := \mathbf{Z}\begin{bmatrix}-\mathbf{R}' & \mathbf{I}\end{bmatrix}$, then the feedback matrix $\mathbf{H} := \mathbf{Z}\mathbf{Q}_2^{-1}$ makes (7.7) stable and the H_∞-norm of the transfer function, H_{ext}, between $\mathbf{u}_{ext}$ and $\mathbf{e}$ is smaller than γ.

Proof. Note that

$$\mathbf{Q}_2\begin{bmatrix}-\mathbf{R}' & \mathbf{I}\end{bmatrix} = \bar{\mathbf{C}}\tilde{\mathbf{Q}}.$$

Furthermore, $\mathbf{H} = \mathbf{Z}\mathbf{Q}_2^{-1}$ if and only if $\mathbf{Z} = \mathbf{H}\mathbf{Q}_2$. Therefore,

$$\bar{\mathbf{Z}} = \mathbf{Z}\begin{bmatrix}-\mathbf{R}' & \mathbf{I}\end{bmatrix} = \mathbf{H}\mathbf{Q}_2\begin{bmatrix}-\mathbf{R}' & \mathbf{I}\end{bmatrix} = \mathbf{H}\bar{\mathbf{C}}\tilde{\mathbf{Q}}.$$

Replacing this last expression in (7.9) we obtain a sufficient condition for the BRL applied to system (7.7). □

Note that this theorem is only sufficient and not necessary because the choice of matrix $\bar{\mathbf{R}} = [\mathbf{I}\ \ \mathbf{R}]$ is arbitrary. If the inequalities do not hold for a particular matrix $\bar{\mathbf{R}}$, it does not mean that they do not hold for some other matrix. In particular, if $\mathbf{R} = \mathbf{0}$, we have

$$\begin{aligned}\bar{\mathbf{A}}\tilde{\mathbf{Q}} + \tilde{\mathbf{Q}}\bar{\mathbf{A}}' &= \begin{bmatrix}\mathbf{A}_{cl}\mathbf{Q}_1 + \mathbf{Q}_1\mathbf{A}_{cl}' & \mathbf{Q}_1\mathbf{C}'\mathbf{T}' - \mathbf{A}_{cl}\mathbf{R}\mathbf{Q}_2 \\ \star & -\mathbf{T}\mathbf{C}\mathbf{R}\mathbf{Q}_2 - \mathbf{Q}_2\mathbf{R}'\mathbf{C}'\mathbf{T}'\end{bmatrix} \\ &= \begin{bmatrix}\mathbf{A}_{cl}\mathbf{Q}_1 + \mathbf{Q}_1\mathbf{A}_{cl}' & \mathbf{Q}_1\mathbf{C}'\mathbf{T}' \\ \star & \mathbf{0}\end{bmatrix}.\end{aligned}$$

Which cannot be negative definite as the lower right block is identically zero and therefore (7.9) cannot hold. An interesting choice for this matrix is $\mathbf{R} = \frac{1}{2}\mathbf{C}^{+}\mathbf{T}^{+}$ so that $-\mathbf{T}\mathbf{C}\mathbf{R}\mathbf{Q}_2 - \mathbf{Q}_2\mathbf{R}'\mathbf{C}'\mathbf{T}' = -\mathbf{Q}_2$. Also note that because of Schur's complement, $\tilde{\mathbf{Q}} > 0 \iff \mathbf{Q}_1 > \mathbf{R}\mathbf{Q}_2\mathbf{R}'$.

Determining H and L.

In the case where $\mathbf{L}$ is unknown, a similar procedure can be defined by writing the system as

$$\dot{\tilde{\mathbf{x}}} = \left(\tilde{\mathbf{A}} + \tilde{\mathbf{B}}\mathbf{H}\bar{\mathbf{C}} + \tilde{\mathbf{B}}\mathbf{L}\hat{\mathbf{C}}\right)\tilde{\mathbf{x}},$$

with

$$\hat{\mathbf{C}} := \begin{bmatrix}\mathbf{C} & \mathbf{0}\end{bmatrix}.$$

We made $\mathbf{u}_{ext} = \mathbf{0}$ to illustrate only the stability condition.

Theorem 7.2. *For given fixed matrices* $\mathbf{R} \in \mathbb{R}^{n\times(N-1)q}$, $\mathbf{X} \in \mathbb{R}^{n-q\times n}$, *and* $\mathbf{Y} \in \mathbb{R}^{n-q\times(N-1)q}$ *such that the composed matrices*

$$\begin{bmatrix}\bar{\mathbf{C}} \\ \bar{\mathbf{R}}\end{bmatrix} := \begin{bmatrix}\mathbf{0} & \mathbf{I} \\ \mathbf{I} & \mathbf{R}\end{bmatrix} \in \mathbb{R}^{(n+(N-1)q)\times(n+(N-1)q)}$$

and

$$\hat{\mathbf{T}} := \begin{bmatrix}\hat{\mathbf{C}} \\ \hat{\mathbf{R}}\end{bmatrix} := \begin{bmatrix}\mathbf{C} & \mathbf{0} \\ \mathbf{X} & \mathbf{Y}\end{bmatrix} \in \mathbb{R}^{(n+(N-1)q)\times(n+(N-1)q)}$$

are full rank, if there exits matrices $\mathbf{Q}_1 > 0$, $\mathbf{Q}_2 > 0$, $\mathbf{Z}$ *and* $\hat{\mathbf{Z}}$ *of proper dimensions such that*

$$\tilde{\mathbf{Q}} := \begin{bmatrix} \mathbf{Q}_1 & -\mathbf{R}\mathbf{Q}_2 \\ \star & \mathbf{Q}_2 \end{bmatrix} > 0, \tag{7.10}$$

$$\tilde{\mathbf{A}}\tilde{\mathbf{Q}} + \tilde{\mathbf{Q}}\tilde{\mathbf{A}}' + \tilde{\mathbf{B}}\bar{\mathbf{Z}} + \bar{\mathbf{Z}}'\tilde{\mathbf{B}} + \tilde{\mathbf{B}}\hat{\mathbf{Z}} + \hat{\mathbf{Z}}'\tilde{\mathbf{B}}' < 0, \tag{7.11}$$

$$\hat{\mathbf{Z}}\hat{\mathbf{R}}' = \mathbf{0}, \tag{7.12}$$

$$\hat{\mathbf{C}}\tilde{\mathbf{Q}}\hat{\mathbf{R}}' = \mathbf{0}, \tag{7.13}$$

with $\bar{\mathbf{Z}} := \mathbf{Z}\begin{bmatrix} -\mathbf{R}' & \mathbf{I} \end{bmatrix}$*, then* $\tilde{\mathbf{A}} + \tilde{\mathbf{B}}\mathbf{H}\bar{\mathbf{C}} + \tilde{\mathbf{B}}\mathbf{L}\hat{\mathbf{C}}$ *is Hurwitz with* $\mathbf{H} := \mathbf{Z}\mathbf{Q}_2^{-1}$ *and* $\mathbf{L} := \hat{\mathbf{Z}}\hat{\mathbf{C}}'\left(\mathbf{C}\mathbf{Q}_1\mathbf{C}'\right)^{-1}$.

Proof. To determine $\mathbf{H}$ from the variable $\bar{\mathbf{Z}}$ the proof is the same as in Theorem 7.1. To determinate $\mathbf{L}$ from the variable $\hat{\mathbf{Z}}$, define $\hat{\mathbf{K}} := \hat{\mathbf{Z}}\tilde{\mathbf{Q}}^{-1} \iff \hat{\mathbf{Z}} = \hat{\mathbf{K}}\tilde{\mathbf{Q}}$. If (7.11) holds with $\tilde{\mathbf{Q}} > 0$ and $\mathbf{Z} = \mathbf{H}\mathbf{Q}_2$, we have

$$\left(\tilde{\mathbf{A}} + \tilde{\mathbf{B}}\mathbf{H}\bar{\mathbf{C}} + \tilde{\mathbf{B}}\hat{\mathbf{K}}\right)\tilde{\mathbf{Q}} + \tilde{\mathbf{Q}}\left(\tilde{\mathbf{A}} + \tilde{\mathbf{B}}\mathbf{H}\bar{\mathbf{C}} + \tilde{\mathbf{B}}\hat{\mathbf{K}}\right)' < 0.$$

If additionally (7.12) and (7.13) hold, we can write

$$\begin{aligned}
\hat{\mathbf{K}} &= \hat{\mathbf{Z}}\tilde{\mathbf{Q}}^{-1} \\
&= \hat{\mathbf{Z}}\hat{\mathbf{T}}'\left(\hat{\mathbf{T}}\tilde{\mathbf{Q}}\hat{\mathbf{T}}'\right)^{-1}\hat{\mathbf{T}} \\
&= \hat{\mathbf{Z}}\begin{bmatrix} \hat{\mathbf{C}}' & \hat{\mathbf{R}}' \end{bmatrix}\left(\begin{bmatrix} \hat{\mathbf{C}} \\ \hat{\mathbf{R}} \end{bmatrix}\tilde{\mathbf{Q}}\begin{bmatrix} \hat{\mathbf{C}}' & \hat{\mathbf{R}}' \end{bmatrix}\right)^{-1}\begin{bmatrix} \hat{\mathbf{C}} \\ \hat{\mathbf{R}} \end{bmatrix} \\
&= \begin{bmatrix} \hat{\mathbf{Z}}\hat{\mathbf{C}}' & \hat{\mathbf{Z}}\hat{\mathbf{R}}' \end{bmatrix}\begin{bmatrix} \hat{\mathbf{C}}\tilde{\mathbf{Q}}\hat{\mathbf{C}}' & \hat{\mathbf{C}}\tilde{\mathbf{Q}}\hat{\mathbf{R}}' \\ \hat{\mathbf{R}}\tilde{\mathbf{Q}}\hat{\mathbf{C}}' & \hat{\mathbf{R}}\tilde{\mathbf{Q}}\hat{\mathbf{R}} \end{bmatrix}^{-1}\begin{bmatrix} \hat{\mathbf{C}} \\ \hat{\mathbf{R}} \end{bmatrix} \\
&= \begin{bmatrix} \hat{\mathbf{Z}}\hat{\mathbf{C}}' & \mathbf{0} \end{bmatrix}\begin{bmatrix} \left(\hat{\mathbf{C}}\tilde{\mathbf{Q}}\hat{\mathbf{C}}'\right)^{-1} & \mathbf{0} \\ \mathbf{0} & \left(\hat{\mathbf{R}}\tilde{\mathbf{Q}}\hat{\mathbf{R}}\right)^{-1} \end{bmatrix}\begin{bmatrix} \hat{\mathbf{C}} \\ \hat{\mathbf{R}} \end{bmatrix} \\
&= \hat{\mathbf{Z}}\hat{\mathbf{C}}'\left(\hat{\mathbf{C}}\tilde{\mathbf{Q}}\hat{\mathbf{C}}'\right)^{-1}\hat{\mathbf{C}} \\
&= \hat{\mathbf{Z}}\hat{\mathbf{C}}'\left(\mathbf{C}\mathbf{Q}_1\mathbf{C}'\right)^{-1}\hat{\mathbf{C}} \\
&= \mathbf{L}\hat{\mathbf{C}}.
\end{aligned}$$

Therefore,

$$\left(\tilde{\mathbf{A}} + \tilde{\mathbf{B}}\mathbf{H}\bar{\mathbf{C}} + \tilde{\mathbf{B}}\mathbf{L}\hat{\mathbf{C}}\right)\tilde{\mathbf{Q}} + \tilde{\mathbf{Q}}\left(\tilde{\mathbf{A}} + \tilde{\mathbf{B}}\mathbf{H}\bar{\mathbf{C}} + \tilde{\mathbf{B}}\mathbf{L}\hat{\mathbf{C}}\right)' < 0,$$

what implies that $\tilde{\mathbf{A}} + \tilde{\mathbf{B}}\mathbf{H}\bar{\mathbf{C}} + \tilde{\mathbf{B}}\mathbf{L}\hat{\mathbf{C}}$ is Hurwitz. □

Note that, if $\mathbf{C}$ is block diagonal, forcing $\mathbf{Q}_1$ and the product $\hat{\mathbf{Z}}\hat{\mathbf{C}}'$ to also be block diagonal, leads to a block diagonal matrix $\mathbf{L}$. The problem with this approach is the arbitrary choice of suitable matrices $\mathbf{X}$ and $\mathbf{Y}$.

Numeric attempts (see Example 7.1 below) show that the norm of matrix $\mathbf{H}$ have some influence on the oscillations of signal $\mathbf{z}$. If for this or other reasons, it is of interest to limit the norm of $\mathbf{H}$, we additionally want to enforce that

$$\mathbf{H}'\mathbf{H} < M^2\mathbf{I} \iff \mathbf{Z}'\mathbf{Z} - M^2\mathbf{Q}_2^2 < 0, \tag{7.14}$$

with $M \in \mathbb{R}^+$ a positive design constant. This expression is not linear with respect to $\mathbf{Z}$ and $\mathbf{Q}_2$. To avoid this, the Crossed Products proposition can be used (see Proposition A.15) to enforce negativity of an upper bound for (7.14):

$$\frac{1}{\alpha}\left(\mathbf{Z}'\mathbf{Z} - M^2\mathbf{Q}_2^2\right) \leqslant \frac{1}{\alpha}\mathbf{Z}'\mathbf{Z} + M^2\left(\alpha\mathbf{I} - 2\mathbf{Q}_2\right) < 0,$$

for some scalar $\alpha > 0$. By Schur complement, this is equivalent to:

$$\begin{bmatrix} M^2\left(\alpha\mathbf{I} - 2\mathbf{Q}_2\right) & \mathbf{Z}' \\ \star & -\alpha\mathbf{I} \end{bmatrix} < 0 \tag{7.15}$$

This last inequality is linear with respect to its variables and can be used to complement Theorem 7.1.

Example 7.1. Consider $N = 10$ agents and the following matrices:

$$\bar{\mathbf{A}}_a = \begin{bmatrix} -1.60 & 0.20 \\ 0.10 & -3.30 \end{bmatrix}, \bar{\mathbf{A}}_b = \begin{bmatrix} -1.80 & 0.40 \\ 0.20 & -2.30 \end{bmatrix},$$

and $\forall i \in \{1,\ldots,10\}$, $\mathbf{B}_i = \mathbf{I}$ and $\mathbf{C}_i = \mathbf{I}$. The closed loops dynamics of each agents are so that $\mathbf{A}_{cl,i} = \mathbf{A}_i + \mathbf{B}_i\mathbf{L}_i\mathbf{C}_i = \bar{\mathbf{A}}_a$ when $i \in \{1,\ldots,5\}$ and $\mathbf{A}_{cl,i} = \bar{\mathbf{A}}_b$ when $i \in \{6,\ldots,10\}$. This represents systems whose velocity along the horizontal axis is given by the first output $[\mathbf{y}_i]_1$ and along the vertical axis by the second output $[\mathbf{y}_i]_2$. In this case, vector $\mathbf{z}$ represents the position of the agents on the plane. We consider the organization graph depicted in Figure 7.2 with the corresponding matrix

$$\mathbf{T} = D'(\mathscr{T}^o) = \begin{bmatrix} \mathbf{1}_{18\times 2} & -\mathbf{I}_{18\times 18} \end{bmatrix}.$$

First consider that the relative position feedback is defined as $\mathbf{H} = -\alpha\mathbf{T}'$, with $\alpha > 0$ a design scalar variable. In this case, $\gamma_{ext} = \|H_{ext}(s)\|_\infty$, $\gamma_w = \|H_w(s)\|_\infty$ and the eigenvalues $\lambda = \sigma + j\omega$ of matrix $\tilde{\mathbf{A}}_{cl}$ are functions of α only and can be easily computed. From a direct comparison with a second order transfer function, we can define for each eigenvalue a damping ration $\zeta = -\sigma/\sqrt{\sigma^2+\omega^2}$, and a natural frequency $\omega_n = \sqrt{\sigma^2+\omega^2}$. If $\zeta = 1$, then the

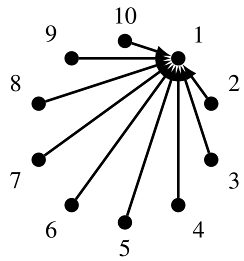


Figure 7.2.: Organization Tree $\mathscr{T}^o$ for Example 7.1.

respective eigenvalue is real and negative (critically damped). Smaller values for the damping constant imply oscillating behaviors which in the extreme case $\zeta = 0$ are undamped with frequency ω_n. For different values of α, the H_∞-norms are shown in Figure 7.3 a), while the real parts of the eigenvalues of the closed loop matrix are shown in b). The damping ratios and the natural frequencies are respectively shown in Figure 7.3 c) and d).

It can be seen that the H_∞-norms decrease with α making the closed loop system less sensitive to external inputs or reference changes. However, large values of α also make the damping ration of many eigenvalues to decrease sharply, making the response of the network more oscillatory. Furthermore, the natural frequency of these oscillations also increase with α. In practice, a compromise between sensitivity and oscillations must be achieved when designing a feedback matrix. Note that the stability of the closed loop system is always guaranteed as the real part of the eigenvalues are negative. However, with small values of α the speed of the response is also compromised as the eigenvalues are closer to the imaginary axis.

With $\mathbf{R} = \frac{1}{2}\mathbf{C}^+\mathbf{T}^+$ and $\gamma = 1.0$, Theorem 7.1 can be used to design a general feedback matrix $\mathbf{H}$. Simply applying the theorem leads to a matrix with $\|\mathbf{H}\| = 6.3960$ and $\|H_{ext}(s)\|_\infty = 0.4483$. In this case, the closed loop matrix $\tilde{\mathbf{A}}_{cl}$ is such that the largest real part of the eigenvalues is $\sigma_{max} = -0.7941$ (*i.e.* the matrix is Hurwitz), but the smallest damping ration of the eigenvalues is $\zeta_{min} = 0.1773$ what would imply important oscillations in the position of the agents. If the bound condition for the norm of $\mathbf{H}$, LMI (7.15), is included with $M = 1.0$, the problem with $\gamma = 1.0$ becomes unfeasible. That is, a feedback matrix such that the network attenuates the influence of external inputs over the formation objective and the oscillation of the agents is “mild”, cannot be achieved. By increasing the tolerated H_∞-norm bound to $\gamma = 2.0$, with $M = 1.0$, a solution for both LMI conditions can be obtained such that $\|H_{ext}(s)\|_\infty = 1.4615$, $\|\mathbf{H}\| = 0.8492$, $\sigma_{max} = -0.2799$ and $\zeta_{min} = 0.5064$. That is, a feedback that ensures that the oscillation of the network is around one third of the previous case, but where the influence of the external signal is larger. ■

a)

b)

c)

d)

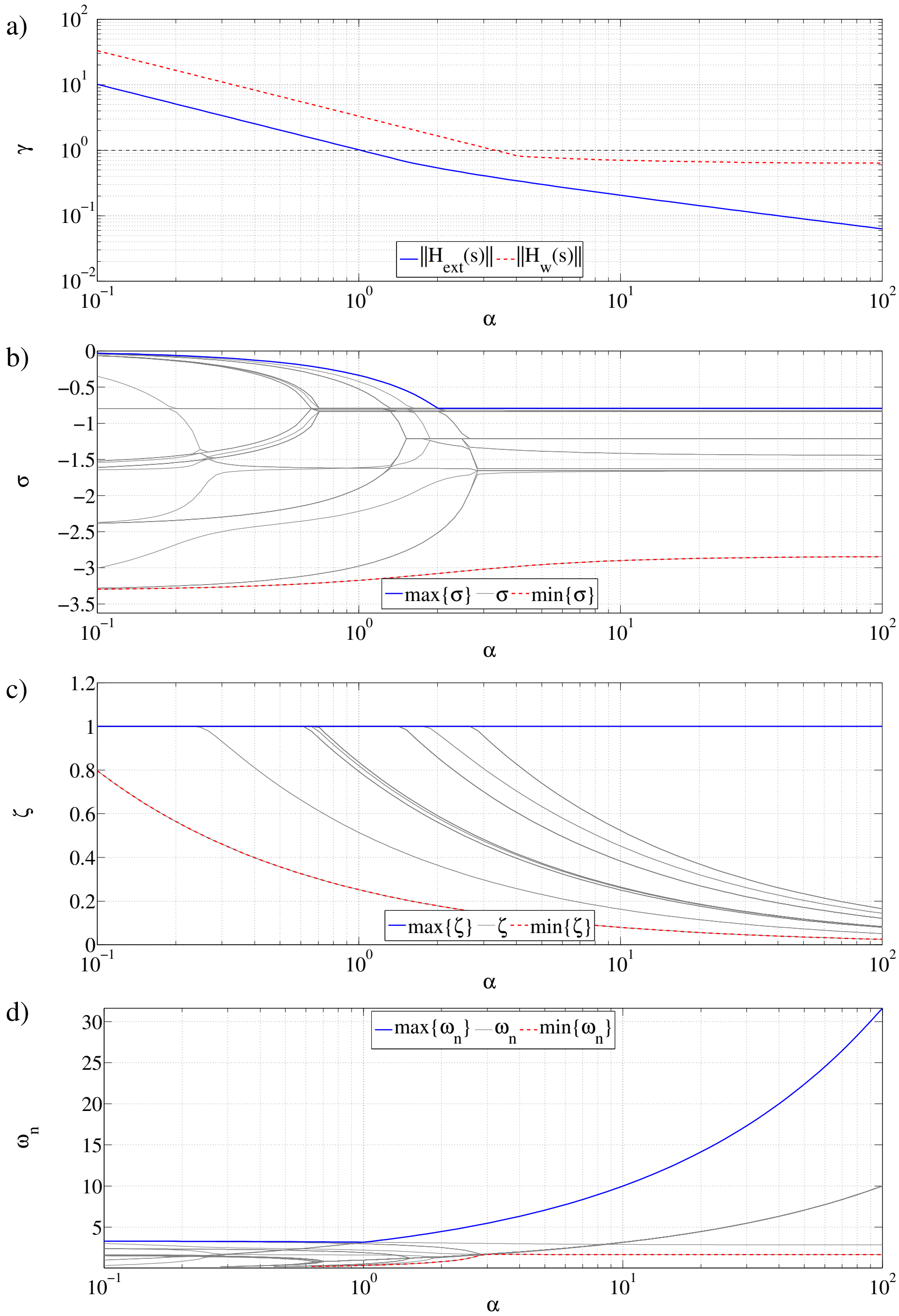


Figure 7.3.: Characteristic values of the network in Example 7.1 when $\mathbf{H} = -\alpha \mathbf{T}'$. a) $\gamma = \|H_{ext}(s)\|_\infty$ and $\gamma = \|H_w(s)\|_\infty$, b) $\sigma = \mathrm{Re}\left\{\mathrm{eig}\left\{\tilde{\mathbf{A}}_{cl}\right\}\right\}$, c) $\zeta = -\sigma/\sqrt{\sigma^2+\omega^2}$, and d) $\omega_n = \sqrt{\sigma^2+\omega^2}$.

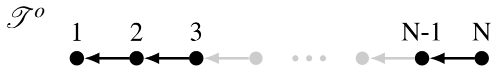


Figure 7.4.: Path organization graph $\mathscr{T}_P$.

7.2.3. Typical 2D reference vectors

Consider that $q = 2$, *i.e.*, the output space of all agents is $\mathbb{R}^2$. We assume implicitly that the plane is described in Cartesian coordinates. Also, consider the organization defined by the directed tree $\mathscr{T}^o$ in Figure 7.4, *i.e.*

$$\mathbf{T}_P = D'(\mathscr{T}_P) = \text{row}\left\{\mathbf{I}_{(N-1)q}, \mathbf{0}_{(N-1)q\times q}\right\} - \text{row}\left\{\mathbf{0}_{(N-1)q\times q}, \mathbf{I}_{(N-1)q}\right\}$$
$$= \begin{bmatrix} \mathbf{I} & -\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & -\mathbf{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} & -\mathbf{I} \end{bmatrix}.$$

In that case, the specified vectors $\mathbf{p}$ can be obtained by

$$\mathbf{p} = \text{col}\{\mathbf{q}_l\}_{l=1}^{N-1}$$

where $\mathbf{q}_l \in \mathbb{R}^2$ depends on the desired figure. Some examples are:

- Line with slope m and uniform distance d between agents:

  $$\mathbf{q}_l = \frac{d}{\sqrt{1+m^2}} \begin{bmatrix} 1 \\ m \end{bmatrix}$$

  $\forall l \in \{1, 2, ..., N-1\}$. Note that when $m = 0$, $\mathbf{q}_l = [d,\ 0]'$ and when $m \to +\infty$, $\mathbf{q}_l \to [0,\ d]'$.

- Equilateral triangle of side a:

  $$\mathbf{q}_l = \frac{a}{2v} \begin{bmatrix} 1 \\ \sqrt{3} \end{bmatrix}$$

  for $1 \leqslant l \leqslant v$, with $v = \text{floor}\{N/3\}$;

  $$\mathbf{q}_l = \frac{a}{2(w-v)} \begin{bmatrix} 1 \\ -\sqrt{3} \end{bmatrix}$$

for $v+1 \leqslant l \leqslant w$, with $w = \text{floor}\{2N/3\}$; and,

$$\mathbf{q}_l = \frac{a}{N-w}\begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

for $w+1 \leqslant l \leqslant N-1$. The integer part function floor is defined for $x \in \mathbb{R}_0^+$ as $\text{floor}\{x\} = \max\{m \in \mathbb{N}_0 | m \leqslant x\}$.

- Square of side a:

$$\mathbf{q}_l = \frac{a}{u}\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

for $1 \leqslant l \leqslant u$, with $u = \text{floor}\{N/4\}$;

$$\mathbf{q}_l = \frac{a}{v-u}\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

for $u+1 \leqslant l \leqslant v$, with $v = \text{floor}\{N/2\}$;

$$\mathbf{q}_l = \frac{a}{w-v}\begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

for $v+1 \leqslant l \leqslant w$, with $u = \text{floor}\{3N/4\}$;

$$\mathbf{q}_l = \frac{a}{N-w}\begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

for $w+1 \leqslant l \leqslant N-1$.

- Regular polygon with N sides of length a:

$$\mathbf{q}_l = a\begin{bmatrix} \sin\left(\frac{2\pi}{N}l\right) \\ \cos\left(\frac{2\pi}{N}l\right) \end{bmatrix}$$

$\forall l \in \{1,2,...,N-1\}$. Note that this is equivalent to locating the agents in a circumference of radius $r = \frac{a}{2}\csc\left(\frac{\pi}{N}\right)$.

Other figures and motion transformations such as rotation, escalation or reflexions of the previous figures can be defined according to the needs of the controller.

7.2.4. Formation Decision

We have shown that with centralized measurements, formation around a reference vector $\mathbf{p}$ can be successfully reached. It remains to be answered how vector $\mathbf{p}$ should be chosen by the central controller in a systematic way. In this section a method based on the linear programming problem known as the *Assignment Problem* is proposed. This problem can be described as how to assign a given number of resources (or workers) to an equal number of tasks in such a way that an overall linear cost is minimized. See Appendix A.5 for details.

The proposed methodology can be informally explained as follows. At one moment, the central controller decides to command the N agents distributed in $\mathbb{R}^q$ to form a given figure (say, "a circle"). However, this specification can be satisfied by any of the $N!$ permutations that result from ordering the agents in a different way (given a circle, interchanging one agent with other results in a "different" circle, and so there are $N!$ circles). Furthermore, the centroid of the figure can be anywhere in $\mathbb{R}^q$. Therefore the central controller additionally requires that the agents form a figure that is centered at the mean value of the current position of the agents and that the overall movement of the agents is minimized. That is, at the moment when the order to form is given, the controller needs to *assign* each of the N agents to each of the N positions that define the figure in order to minimize a functional.

To do this, consider a permutation matrix $\mathbf{E} \in \mathbb{R}^{Nq \times Nq}$ described by

$$\mathbf{E} = \operatorname{col}\left\{\mathbf{s}_k'\right\}_{k \in P},$$

where

$$P \in \left\{\{k_1, k_2, \ldots, k_N\} | k_i \in W \ \wedge \ k_i \neq k_j\right\},$$

is an ordered permutation of the set $W = \{1, 2, \ldots, N\}$. Note that $\det\{\mathbf{E}\} = 1$. Pre-multiplying a vector by this matrix results in a vector that is a permutation of the N subvectors in $\mathbb{R}^q$ that compose the original aggregated vector. This permutation is completely described by the set P.

Example 7.2. If $N = 3$, there are $3! = 6$ possible permutations, and so

$$P \in \{\{1,2,3\}, \{1,3,2\}, \{2,1,3\}, \{2,3,1\}, \{3,1,2\}, \{3,2,1\}\},$$

respectively,

$$\mathbf{E} \in \left\{ \begin{array}{@{}c@{}c@{}} \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} & \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} \end{array}, \begin{array}{@{}c@{}c@{}} \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix} & \begin{matrix} 1 \\ 3 \\ 2 \end{matrix} \end{array}, \begin{array}{@{}c@{}c@{}} \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} & \begin{matrix} 2 \\ 1 \\ 3 \end{matrix} \end{array}, \begin{array}{@{}c@{}c@{}} \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix} & \begin{matrix} 2 \\ 3 \\ 1 \end{matrix} \end{array}, \begin{array}{@{}c@{}c@{}} \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix} & \begin{matrix} 3 \\ 1 \\ 2 \end{matrix} \end{array}, \begin{array}{@{}c@{}c@{}} \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix} & \begin{matrix} 3 \\ 2 \\ 1 \end{matrix} \end{array} \right\}.$$

The set P should be understood as the worker assigned to each task (and not the task assigned to each worker). That is, for example, $P = \{3,2,1\}$ means that worker 3 is assigned to the first task, worker 2 to the second task, and worker 1 to the third task. ■

Assume that the desired formation of the agents is described through a known organization matrix $\mathbf{T}_P \in \mathbb{R}^{(N-1)q\times Nq}$ and a fixed vector $\mathbf{p}_P \in \mathbb{R}^{(N-1)q}$. Then, one of the possible formation vectors $\bar{\mathbf{z}}$ of the network around its current mean value $E\{\mathbf{z}\}$ is

$$\bar{\mathbf{z}} = \mathbf{T}_P^{+}\mathbf{p}_P + \mathbf{1}E\{\mathbf{z}\}.$$

Any of the $N!$ permutations, $\mathbf{E}\bar{\mathbf{z}}$, of $\bar{\mathbf{z}}$ is a candidate formation for the network. Note that

$$\mathbf{z} = \mathbf{T}^{+}\mathbf{e}_0 + \mathbf{1}E\{\mathbf{z}\}.$$

From here, the goal of the central controller is to choose which permutation matrix $\mathbf{E}^*$ is the one that makes $\bar{\mathbf{z}}^* = \mathbf{E}^*\bar{\mathbf{z}}$ closer to $\mathbf{z}$. That is, to solve the minimization problem

$$\begin{aligned} \min_{\mathbf{E}} \quad & J = \|\mathbf{z} - \mathbf{E}\bar{\mathbf{z}}\|^2 \\ \text{s.t.} \quad & \mathbf{E} = \operatorname{col}\left\{\mathbf{s}_k'\right\}_{k\in P} \end{aligned}$$

As $\mathbf{E}\mathbf{1} = \mathbf{1}$, the objective functional can be rewritten as

$$J = \|\mathbf{T}^{+}\mathbf{e}_0 - \mathbf{E}\mathbf{T}_P^{+}\mathbf{p}_P\|^2 = \sum_{l=1}^{N} \|\mathbf{s}_l'\mathbf{T}^{+}\mathbf{e}_0 - \mathbf{s}_l'\mathbf{E}\mathbf{T}_P^{+}\mathbf{p}_P\|^2.$$

Note that $\mathbf{s}_l'\mathbf{E} = \mathbf{s}_k'$ with $k \in P$. From here, the problem can be written into an assignment problem by defining resources (the agents) as $k \in W = \{1,2,\ldots,N\}$ and tasks (their position in the formation) $l \in T = \{1,2,\ldots,N\}$. The assignment costs of each vector $\mathbf{s}_k$ relative to each task $l \in T$ are defined as:

$$c(k,l) = \left\|\mathbf{s}_l'\mathbf{T}^{+}\mathbf{e}_0 - \mathbf{s}_k'\mathbf{T}_P^{+}\mathbf{p}_P\right\|^2.$$

And so, considering the notation of the assignment problem in Appendix A.5, $J = \sum_{l=1}^{N}\sum_{k=1}^{N} c(k,l)x_{kl}$. For known vectors $\mathbf{e}_0$ and $\mathbf{p}_P$, solving this problem at a fixed instant with any algorithm in the central controller, leads to a suitable vector $\mathbf{p}$ relative to the measurement organization $\mathbf{T}$ that can be chosen as:

$$\mathbf{p} = \mathbf{T}\bar{\mathbf{z}}^* = \mathbf{T}\left(\mathbf{E}^*\mathbf{T}_P^{+}\mathbf{p}_P + \mathbf{E}^*\mathbf{1}E\{\mathbf{z}\}\right) = \mathbf{T}\mathbf{E}^*\mathbf{T}_P^{+}\mathbf{p}_P.$$

Example 7.3. Consider $N = 10$ agents with $q = 2$ outputs. The agents are initially distributed in the plane so that, considering the organization matrix used in Example 7.1, the initial relative

Table 7.1.: Assignment of Agents within a Formation in Example 7.3.

Case	P	J
Optimal	$\{7,1,2,4,3,8,5,6,9,10\}$	1029.452897
Arbitrary	$\{1,7,3,5,2,4,8,10,9,6\}$	1223.402266

differences are given randomly by

$$\mathbf{e}_0 = \mathrm{col}\left\{ \begin{bmatrix} 122.1499 \\ -129.6115 \end{bmatrix}, \begin{bmatrix} 222.3776 \\ -40.5040 \end{bmatrix}, \begin{bmatrix} 58.4222 \\ 53.5299 \end{bmatrix}, \begin{bmatrix} 120.3640 \\ 145.7429 \end{bmatrix}, \begin{bmatrix} 256.7263 \\ 18.6710 \end{bmatrix}, \begin{bmatrix} 69.7056 \\ -229.3907 \end{bmatrix}, \begin{bmatrix} 208.6200 \\ 10.4253 \end{bmatrix}, \begin{bmatrix} 150.1805 \\ 222.4607 \end{bmatrix}, \begin{bmatrix} 185.8276 \\ 208.6263 \end{bmatrix} \right\}.$$

The controller wishes to reach a lineal formation with slope $m = 1$ and a relative distance between adjacent agents of $d = 30$. That is, using the description and organization matrix of Section 7.2.3, defined by the vector

$$\mathbf{p}_P = \mathrm{col}\left\{ \begin{bmatrix} 21.2132 \\ 21.2132 \end{bmatrix} \right\}_{l=1}^{N-1}.$$

The costs matrix that describes this situation is given by:

$$[c(k,l)] = \begin{bmatrix}
82.2154 & 98.6333 & 180.7350 & 123.8396 & 228.3615 & 230.2818 & 161.9825 & 183.0011 & 310.6433 & 312.1791 \\
81.1053 & 99.3166 & 157.3772 & 102.0060 & 201.6872 & 202.8899 & 181.1957 & 154.9670 & 283.7396 & 283.7943 \\
90.5360 & 108.6235 & 136.6384 & 85.2889 & 176.0866 & 176.3487 & 203.0405 & 127.8287 & 257.5209 & 255.7785 \\
107.7751 & 124.6368 & 119.8854 & 77.0920 & 152.1027 & 151.1074 & 226.7576 & 102.3014 & 232.2195 & 228.2676 \\
129.7466 & 145.1537 & 108.9723 & 80.0759 & 130.6291 & 127.9377 & 251.8186 & 79.9435 & 208.1700 & 201.4685 \\
154.4437 & 168.5374 & 105.7234 & 93.1725 & 113.1049 & 108.1787 & 277.8600 & 64.1588 & 185.8592 & 175.7071 \\
180.7528 & 193.7528 & 110.8147 & 112.9163 & 101.5946 & 94.0064 & 304.6305 & 60.3469 & 165.9895 & 151.5137 \\
208.0631 & 220.1713 & 123.2167 & 136.4517 & 98.2354 & 88.1578 & 331.9538 & 70.4779 & 149.5372 & 129.7685 \\
236.0275 & 247.4079 & 141.0135 & 162.1359 & 103.8217 & 92.2300 & 359.7039 & 89.9585 & 137.7326 & 111.9075 \\
264.4384 & 275.2198 & 162.4416 & 189.0953 & 117.0798 & 105.0760 & 387.7892 & 114.0961 & 131.8299 & 100.0336
\end{bmatrix}$$

The Hungarian Algorithm can be used to solve the associated assignment problem. The optimal result is shown in Table 7.1 along with an arbitrary permutation of the agents in the formation. It is clear that the optimal case has a lower cost. In Figure 7.5, both assignments can be graphically seen. The lines between the points show the distance between the original point and its assignment within the formation, which sum is minimized to obtain the optimal assignment. ■

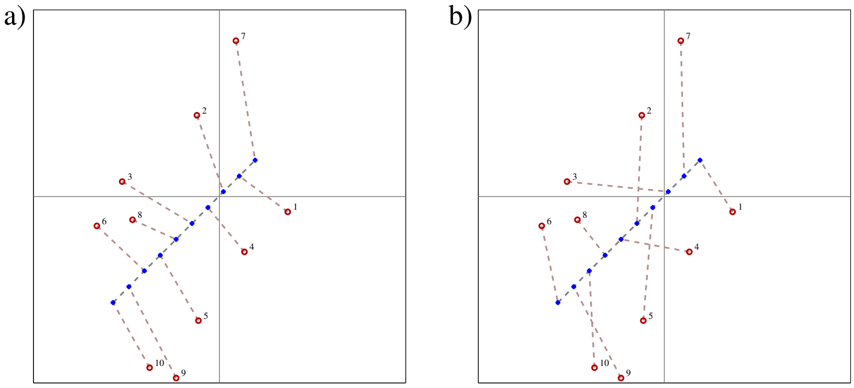


Figure 7.5.: Assignment of N agents within a formation in Example 7.3. a) Optimal assignment, b) Arbitrary assignment.

7.2.5. Fully Distributed Implementation of a Central Controller

The control law proposed in the previous section can be implemented either in the central controller or distributively in the agents. In the first case the output signals of the agents $\mathbf{y}_i$, the external inputs $\mathbf{u}_{ext,i}$, and the control signals $\mathbf{u}_i$ need to be communicated between the agents and the controller, making formation control dependent on the quality of the communication links between the systems. On the other hand, if the agents have the ability of measuring the relative distances between the vectors $\mathbf{z}_i$, and $\mathbf{L} = \operatorname{diag}\{\mathbf{L}_i\}_{i=1}^{N}$, with $\mathbf{L}_i \in \mathbb{R}^{q\times q}$, the controller can be implemented distributively in the agents without the need of active exchange of signals.

Let us consider that each agent has local access to a vector $\mathbf{e}_i = \mathbf{T}_i\mathbf{z}$, that can be used as part of the control strategy, where $\mathbf{T}_i = D'(\mathscr{T}_i^{o}) \in \mathbb{R}^{(N-1)q\times Nq}$ and $\mathscr{T}_i$ is an arbitrary spanning tree over all agents. Note that the organizations are potentially different for every agent. Observe further that $\mathbf{z} = \mathbf{T}_i^{+}\mathbf{e}_i + \mathbf{J}\mathbf{z}$ and therefore,

$$\mathbf{e}_0 = \mathbf{T}\left(\mathbf{T}_i^{+}\mathbf{e}_i + \mathbf{J}\mathbf{z}\right) = \mathbf{T}\mathbf{T}_i^{+}\mathbf{e}_i.$$

That is, any relative measurement performed by the agents can be “translated” into the relative measurement done by the centralized controller. Similarly, $\mathbf{e}_i = \mathbf{T}_i\mathbf{T}^{+}\mathbf{e}_0$. Also note that $\mathbf{T}^{+}\mathbf{T}\mathbf{T}_i^{+} = (\mathbf{I}-\mathbf{J})\,\mathbf{T}_i^{+} = \mathbf{T}_i^{+}$.

With this, if vector $\mathbf{p}_P$ and matrix $\mathbf{T}_P$ are known by each agent, the calculation of the reference vector can be done locally by independently solving the associated assignment problem with the cost functions:

$$c(k,l) = \left\|\mathbf{s}_l'\mathbf{T}^{+}\mathbf{e}_0 - \mathbf{s}_k'\mathbf{T}_P^{+}\mathbf{p}_P\right\|^2 = \left\|\mathbf{s}_l'\mathbf{T}_i^{+}\mathbf{e}_i - \mathbf{s}_k'\mathbf{T}_P^{+}\mathbf{p}_P\right\|^2.$$

A local reference vector can then be defined as:

$$\mathbf{p}_i = \mathbf{T}_i\mathbf{T}^{+}\mathbf{p} = \mathbf{T}_i\mathbf{T}^{+}\mathbf{T}\mathbf{E}^{*}\mathbf{T}_P^{+}\mathbf{p}_P = \mathbf{T}_i\mathbf{E}^{*}\mathbf{T}_P^{+}\mathbf{p}_P.$$

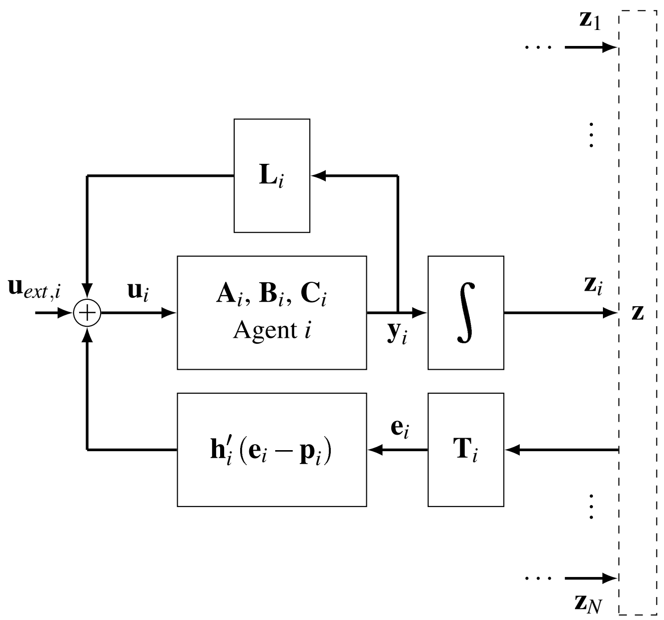


Figure 7.6.: Distributed Formation Control Strategy without Communication.

From here, each input signal $\mathbf{u}_i = \mathbf{s}_i'\mathbf{u}$ can be calculated as:

$$\mathbf{u}_i = \mathbf{L}_i\mathbf{y}_i + \mathbf{h}_i'\mathbf{e}_i - \mathbf{h}_i'\mathbf{p}_i + \mathbf{u}_{ext,i}, \tag{7.16}$$

with $\mathbf{u}_{ext,i} = \mathbf{s}_i'\mathbf{u}_{ext}$ and $\mathbf{h}_i' = \mathbf{s}_i'\mathbf{H}\mathbf{T}\mathbf{T}_i^+$ a block row of gains used to weight the *locally available* vectors $\mathbf{e}_i$ and $\mathbf{p}_i$. Note that $\mathbf{h}_i'\mathbf{p}_i = \mathbf{s}_i'\mathbf{H}\mathbf{p}$. Through the introduction of these vectors, the centralized measurement $\mathbf{e}_0$ and reference $\mathbf{p}$ are not used for control anymore and, therefore, the transmission of these variables to the agents is not necessary. This control strategy is to be seen in Figure 7.6.

The disadvantage of this fully distributed strategy is that each of the N agents has to perform $N-1$ relative distance measurements, while in the centralized implementation only $N-1$ measurements are required. The use of one strategy instead of the other is then justified by practical considerations regarding the relative cost of signals transmission versus measurement. Naturally, partially distributed strategies can also be studied as particular cases.

Example 7.4. Consider again the network in Example 7.1 and five different position feedback matrices $\mathbf{H}_h = -\alpha_h\mathbf{T}'$, with $h \in \{1,2,3,4,5\}$, defined by $\alpha_1 = 0.1$, $\alpha_2 = 0.5$, $\alpha_3 = 1.0$, $\alpha_4 = 5.0$ and, $\alpha_5 = 10.0$. Each agent has the ability of independently solve the associated assignment problem with an implementation of the Hungarian method and with local relative position measurements, given through a star-graph centered on the agent. The relative position feedback gains $\mathbf{h}_{i,h}$, distributively implemented on each agent $i \in \mathscr{V}$, are calculated for each $h \in \{1,2,3,4,5\}$ so that $\mathbf{h}_{i,h}' = \mathbf{s}_i'\mathbf{H}_h\mathbf{T}\mathbf{T}_i^+$.

Table 7.2.: Numeric Results for Example 7.4.

h	α_h	$\|H_{ext}(s)\|_\infty$	$ISE(60s)$	$IAE(60s)$
1	0.1	10.1588	3147.45	455.37
2	0.5	2.0318	908.47	158.79
3	1.0	1.0159	512.22	85.46
4	5.0	0.3014	198.65	30.63
5	10.0	0.2058	159.71	26.45

The simulations start with the agents not moving and randomly distributed on the plane. They are immediately instructed to form a line with slope $m = 1$ and a distance between the agents $d = 30$. At $t = 30s$, the agents are instructed to change their formation into a circle of radius $r = 100$. Additionally, a constant input $u_{ext,1} = [15,\ 15]'$ is added to the first agent all through the simulation time. Which results on the whole group of agents moving across the plane following the first agent in order to maintain the required formation regardless of the external input.

Due to the assumption that vector $\mathbf{z}$ cannot be measured, the ISD and IAD indicators used up to here cease to make sense for evaluating the performance of the network. That is why, similar accumulated error indicators are defined to obtain a fair comparison of the simulation results:

$$ISE(t) = 10^{-3} \int_0^t \mathbf{e}'\mathbf{e}\, dt,$$

$$IAE(t) = 10^{-2} \int_0^t \sum_{i=1}^{N-1} |[\mathbf{e}]_i|\ dt$$

Table 7.2 shows the numeric results of the simulations in terms of the ISE and IAE indicators and the H$_\infty$-norm of the transfer function between the external input and the formation error. It is clear that a smaller H$_\infty$-norm implies less accumulated error. However, from the discussion in Example 7.1, a higher oscillation should also be expected. This can be seen by comparing the trajectories of the agents as a function of time like in Figure 7.7, with $\alpha_1 = 0.1$, and in Figure 7.9, with $\alpha_5 = 10.0$. Clearly, a higher value of α implies more oscillations. Note that due to the effect of the constant external input in the first agent, the whole network changes constantly its position on the plane. Also, note that for $\alpha_1 = 0.1$, many of the eigenvalues are close to the imaginary axis, making the network response considerably slow.

Figure 7.10 shows the spatial trajectories that the agents follow with $\alpha_3 = 1.0$ in order to achieve a line formation at $t = 30$ and a circle formation at $t = 60$. Figure 7.8 shows the same trajectories as a function of time. It can be seen that the agents follow relatively smooth trajectories with only some oscillations in the behavior of the first agent. ■

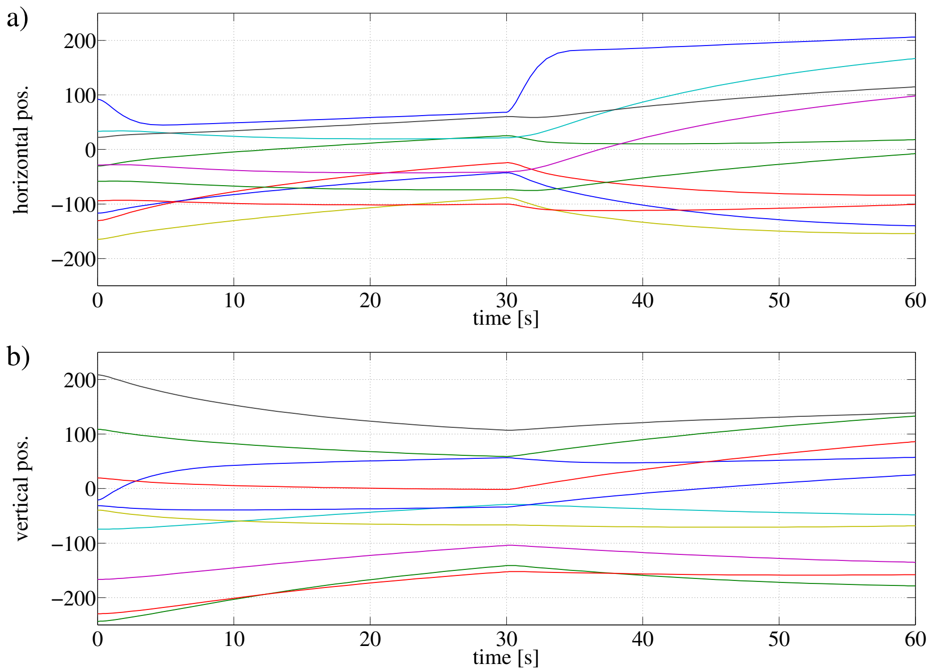


Figure 7.7.: Temporal trajectories for $\alpha = 0.1$ in Example 7.4. a) Horizontal position and b) Vertical position.

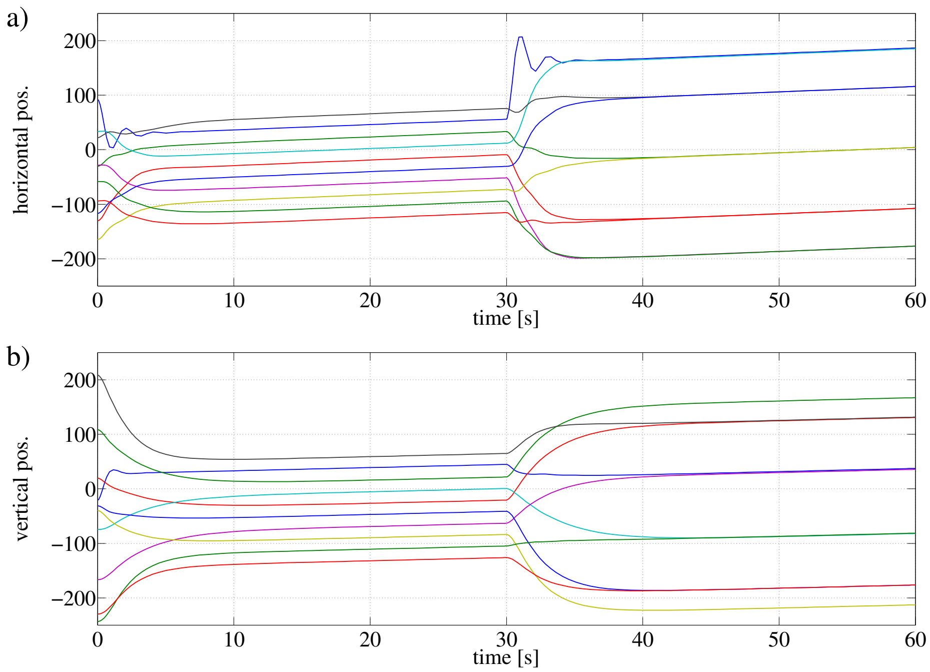


Figure 7.8.: Temporal trajectories for $\alpha = 1.0$ in Example 7.4. a) Horizontal position and b) Vertical position.

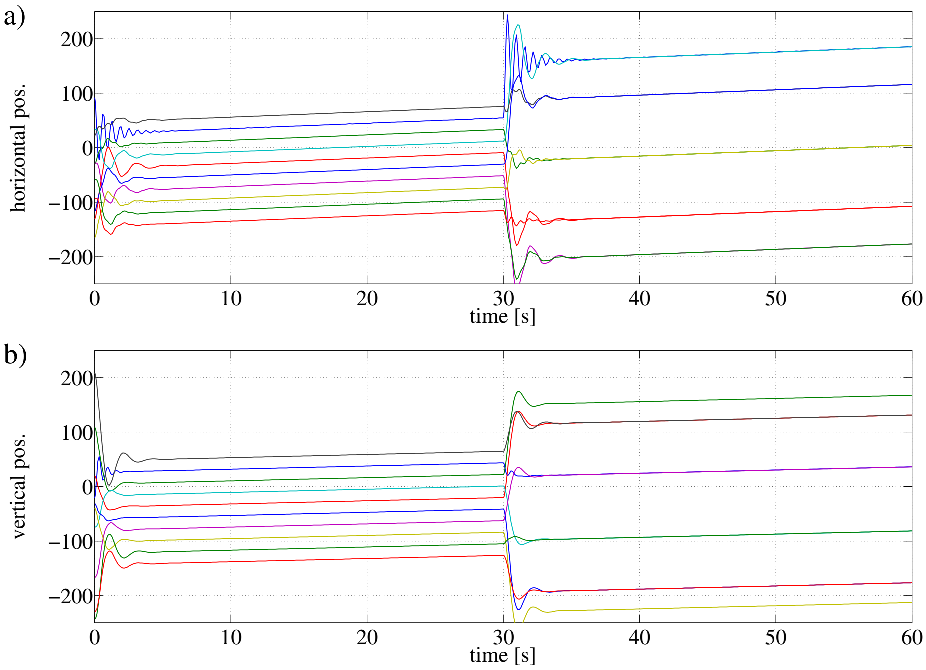


Figure 7.9.: Temporal trajectories for $\alpha = 10.0$ in Example 7.4. a) Horizontal position and b) Vertical position.

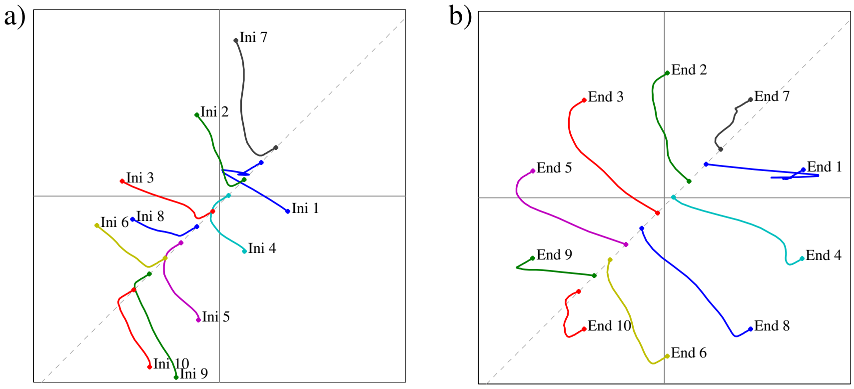


Figure 7.10.: Spatial trajectories for $\alpha = 1.0$ in Example 7.4. a) $t \leqslant 30$, b) $30 < t \leqslant 60$.

7.3. Centralized Analysis of a Network with a Distributively Implemented Controller

Before proceeding, a centralized description of the network with a distributively implemented controller needs to be developed. For this, consider that each agent $i \in \mathscr{V}$ performs relative measurements $\mathbf{e}_i \in \mathbb{R}^{(N-1)q}$ described through an organization matrix $\mathbf{T}_i = D'(\mathscr{T}_i^o)$ in such a way that $\mathbf{e}_i = \mathbf{T}_i\mathbf{z}$, where $\mathbf{z} \in \mathbb{R}^{Nq}$ is measured in the unknown absolute coordinates. This can also be measured from an external position through a nominal organization matrix $\mathbf{T} = D'(\mathscr{T}^o)$, so that $\mathbf{e}_0 = \mathbf{T}\mathbf{z} = \mathbf{T}\mathbf{T}_i^+\mathbf{e}_i \iff \mathbf{e}_i = \mathbf{T}_i\mathbf{T}^+\mathbf{e}_0$. We assume that a general optimal reference vector $\mathbf{p} \in \mathbb{R}^{(N-1)q}$, relative to organization $\mathbf{T}$, is known, constant, and can be "translated" by each agent as $\mathbf{p} = \mathbf{T}\mathbf{T}_i^+\mathbf{p}_i \iff \mathbf{p}_i = \mathbf{T}_i\mathbf{T}^+\mathbf{p}$. Furthermore, the agents implement the control law in equation (7.16). That is, $\forall i \in \mathscr{V}$,

$$\mathbf{u}_i = \mathbf{L}_i\mathbf{y}_i + \mathbf{h}_i'\mathbf{e}_i - \mathbf{h}_i'\mathbf{p}_i + \mathbf{u}_{ext,i},$$

where the matrices $\mathbf{h}_i \in \mathbb{R}^{(N-1)q\times q}$ are known and fixed. From here, we can write that

$$\mathbf{u} = \operatorname{col}\{\mathbf{u}_i\}_{i\in\mathscr{V}} = \mathbf{L}\mathbf{y} + \left(\sum_{i\in\mathscr{V}} \mathbf{s}_i\mathbf{h}_i'\mathbf{T}_i\right)\mathbf{T}^+(\mathbf{e}_0 - \mathbf{p}) + \mathbf{u}_{ext}. \tag{7.17}$$

With $\mathbf{L} = \operatorname{diag}\{\mathbf{L}_i\}_{i=1}^N$. Note that this expression is equivalent to equation (7.5) when $\mathbf{h}_i' = \mathbf{s}_i'\mathbf{H}\mathbf{T}\mathbf{T}_i^+$:

$$\left(\sum_{i\in\mathscr{V}} \mathbf{s}_i\mathbf{h}_i'\mathbf{T}_i\right)\mathbf{T}^+ = \left(\sum_{i\in\mathscr{V}} \mathbf{s}_i\mathbf{s}_i'\mathbf{H}\mathbf{T}\mathbf{T}_i^+\mathbf{T}_i\right)\mathbf{T}^+ = \sum_{i\in\mathscr{V}} \mathbf{s}_i\mathbf{s}_i'\mathbf{H}\mathbf{T}(\mathbf{I}-\mathbf{J})\mathbf{T}^+ = \mathbf{I}\mathbf{H}\mathbf{I} = \mathbf{H}.$$

However, (7.17) captures in a better way the *individual* behavior of the agents as it expressly contains the organization matrices $\mathbf{T}_i$ and the locally implemented control gains $\mathbf{h}_i$. In other words, (7.5) can be used for a centralized analysis of a centrally planed consensus and formation strategy, while (7.17) is better suit for a centralized analysis of a distributively implemented control strategy.

Defining an error signal $\mathbf{e} = \mathbf{e}_0 - \mathbf{p}$, and considering a zero external input ($\mathbf{u}_{ext} = \mathbf{0}$) and a constant reference vector ($\frac{d}{dt}\mathbf{p} = \mathbf{0}$), the overall dynamics of the error and the states of the agents can be written as

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{e}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} + \mathbf{B}\mathbf{L}\mathbf{C} & \mathbf{B}\left(\sum_{i\in\mathscr{V}} \mathbf{s}_i\mathbf{h}_i'\mathbf{T}_i\right)\mathbf{T}^+ \\ \mathbf{T}\mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{e} \end{bmatrix}. \tag{7.18}$$

Naturally, if this autonomous system is stable, then formation can be achieved and the states of the agent approach the origin.

7.3.1. Distributed Switching Modes

An interesting problem associated with formation control with distributed relative measurements, is the possibility that, at least temporarily, the measurements associated with each agent change. This can be due to a strategic decision of the agents that change their formation algorithm or to a failure scenario similar to those studied in the unintended switching case in Section 5.2.3. However, contrary to that case and to the general communication dynamics studied in Section 5.3.3, when a failure in relative measurement occurs, the network cannot switch into an "unsafe" mode where a residual matrix interferes with the dynamics of the error. Nevertheless, like in other switching situations, given a nominal mode of the network, the number of possible switching modes that can be defined is considerably large.

This can be seen quite easily by considering only the measurement failure case. Each of the N agents uses $N-1$ relative position measurements. Therefore, there are $N(N-1)$ measurements that can fail. If only one failure occurs there are ${}^{N(N-1)}C_1 = N(N-1)$ possible discrete modes. If two failures occur simultaneously, there are ${}^{N(N-1)}C_2 = N(N-1)(N(N-1)-1)$ modes. *etc*. In total then, there are

$$M_{\sup} = \sum_{i=0}^{N(N-1)} \left({}^{N(N-1)}C_i\right) = \sum_{i=0}^{N(N-1)} \frac{(N(N-1))!}{i!(N(N-1)-i)!} = 2^{N(N-1)}$$

possible discrete modes if up to $N(N-1)$ failures are allowed to occur simultaneously.

At each switching mode, several parameters have to be defined. In particular, N organization matrices and N feedback vectors need to be associated with each mode to completely describe the distributed parameters of the network. Furthermore, additional information on the residence time at each mode and the switching probabilities associated is also needed.

Definition 7.3.1 (Distributed Switching Mode). A *distributed switching mode*, $q_{\hat{\imath}}$, is the union of a set of N matrices $\{\mathbf{T}_i(\hat{\imath})\}_{i\in\mathscr{V}}$, and of N feedback vectors $\{\mathbf{h}_i(\hat{\imath})\}_{i\in\mathscr{V}}$. Each pair $\{\mathbf{T}_i(\hat{\imath}), \mathbf{h}_i(\hat{\imath})\}$ is associated with the agent $i \in \mathscr{V}$. Additionally, an uncertain residence time $T_{(\hat{\imath})} = T_{(\hat{\imath})}^{min} + \Delta T_{(\hat{\imath})}$, with $\Delta T_{(\hat{\imath})} \in [0, \Delta T_{(\hat{\imath})}^{max}]$, is given together with a switching probability vector $\boldsymbol{\pi}_{\hat{\imath}}^{+} \in \Lambda_M$, where M is the total number of distributed switching modes, so $q_{\hat{\imath}} \in Q = \{q_1, q_2, ..., q_M\}$.

With this definition, the control signal of the network can be calculated as

$$\mathbf{u} = \mathbf{L}\mathbf{y} + \left(\sum_{i\in\mathscr{V}} \mathbf{s}_i \mathbf{h}_i'(\hat{\imath}) \mathbf{T}_i(\hat{\imath})\right) \mathbf{T}^{+} (\mathbf{e}_0 - \mathbf{p}) + \mathbf{u}_{ext}, \tag{7.19}$$

Note that it is assumed that the switching matrices $\mathbf{h}_i'(\hat{\imath})\mathbf{T}_i(\hat{\imath})$ multiply the error $\mathbf{e} = \mathbf{e}_0 - \mathbf{p}$ and not only $\mathbf{e}_0$. This can be regarded as a security mechanism. If that where not the case, when $\mathbf{h}_i'(\hat{\imath})\mathbf{T}_i(\hat{\imath}) = \mathbf{0}$, the reference position vector would behave like an input to the network and would modify its transient behavior. With this control law, however, if $\mathbf{h}_i'(\hat{\imath})\mathbf{T}_i(\hat{\imath}) = \mathbf{0}$ then the system remains static until the system comes to a mode where the product is no longer zero.

From here, under no external inputs and constant formation reference, formation can be studied through the stability of the following switching system,

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{e}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}+\mathbf{BLC} & \mathbf{B}\left(\sum_{i\in\mathscr{V}} \mathbf{s}_i \mathbf{h}_i'(\hat{\imath})\mathbf{T}_i(\hat{\imath})\right)\mathbf{T}^+ \\ \mathbf{TC} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{e} \end{bmatrix}, \tag{7.20}$$

with $q_{\hat{\imath}} \in Q$. This can be written in compact way as

$$\dot{\tilde{\mathbf{x}}} = \tilde{\mathbf{A}}(\hat{\imath})\tilde{\mathbf{x}},$$

with

$$\tilde{\mathbf{x}} = \mathrm{col}\left\{\mathbf{x},\mathbf{e}\right\}, \text{ and } \tilde{\mathbf{A}}(\hat{\imath}) = \begin{bmatrix} \mathbf{A}+\mathbf{BLC} & \mathbf{B}\left(\sum_{i\in\mathscr{V}} \mathbf{s}_i \mathbf{h}_i'(\hat{\imath})\mathbf{T}_i(\hat{\imath})\right)\mathbf{T}^+ \\ \mathbf{TC} & \mathbf{0} \end{bmatrix}.$$

Again, the results of Chapter 3, particularly Theorem 3.7, can be directly applied to study stability of the previous expression and therefore determinate if the network can reach formation under a specific switching process. We distinguish three cases where switching may occur:

Definition 7.3.2 (Switching Types).

- *Failure switching*: If for all $i \in \mathscr{V}$ and for all $q_{\hat{\imath}} \in Q$, $\mathbf{h}_i(\hat{\imath}) = \mathbf{h}_i \in \mathbb{R}^{(N-1)q}$ and $\mathbf{T}_i(\hat{\imath}) = \mathbf{E}_{\hat{\imath}}\mathbf{T}_i$, where $\mathbf{T}_i = D'(\mathscr{T}_i^o) \in \mathbb{R}^{(N-1)q\times Nq}$ and $\mathbf{E}_i(\hat{\imath}) \in \mathbb{R}^{(N-1)q\times(N-1)q}$ is a block diagonal matrix where the block element $[\mathbf{E}_i(\hat{\imath})]_{jj} = \mathbf{0}$ if a failure occurs in the j-th measurement of agent $i \in \mathscr{V}$ and $[\mathbf{E}_i(\hat{\imath})]_{jj} = \mathbf{I}$ otherwise.
- *Algorithmic switching*. If for all $i \in \mathscr{V}$ and for all $q_{\hat{\imath}} \in Q$, $\mathbf{T}_i(\hat{\imath}) = \mathbf{T}_i = D'(\mathscr{T}_i^o)$ and $\mathbf{h}_i(\hat{\imath}) \neq \mathbf{h}_i(\hat{\jmath})$ for at least some $i \in \mathscr{V}$ and some $q_{\hat{\jmath}} \in Q\backslash\{q_{\hat{\imath}}\}$.
- *Apparent switching*. If for all $i \in \mathscr{V}$ and all $\{q_{\hat{\imath}}, q_{\hat{\jmath}}\} \in Q \times Q$, $\mathbf{h}_i'(\hat{\imath})\mathbf{T}_i(\hat{\imath}) = \mathbf{h}_i'(\hat{\jmath})\mathbf{T}_i(\hat{\jmath})$.

The failure switching case refers to an unintended switching process that occurs when successive failures occur in the distributed measurements. Each agent has a nominal organization matrix that describes the measurements that it performs and a nominal feedback matrix. The failures at agent $i \in \mathscr{V}$ are described by the diagonal entries of matrix $\mathbf{E}_i(\hat{\imath})$

The algorithmic and apparent switching cases are intended switching processes. The first one of these deals with intended changes of the feedback matrix maintaining the local measurement of the error. The later case considers switching in the measurements and the feedback weights in such a way that the feedback signal $\mathbf{u}_i$ does actually not change because of the switching. That is, the agents change their measurements but adapt their feedback matrices so that the switching effect is not reveled to the network. This might be the result of a failure avoidance mechanism or technological restrictions in the measurements.

Several different cases can be defined by combining the previous ones. For example, an apparent switching strategy that considers also measurement failures. Because its unintended nature, failure switching demands a more critical stability analysis. The apparent switching case can be trivially analyzed directly from equation (7.18).

Note that the dimension of the switched matrix $\tilde{\mathbf{A}} \in \mathbb{R}^{(Nn+(N-1)q)\times(Nn+(N-1)q)}$ might be very large in networks with too many agents. Furthermore, in the defined switching processes, particularly in the failure case, there are typically a large number of discrete modes. Also, Lyapunov functions have usually a large number of matrix variables. These three facts combined can make the LMI problems associated with stability difficult to solve numerically, leading to scenarios where ir cannot be practically tested.

Example 7.5. Consider the same $N = 10$ agents in the network of Example 7.4 with a relative position feedback $\mathbf{H} = -1.0\mathbf{T}'$ distributively implemented in the agents. Each agent can be under a measurement failure that inhibits all relative position measurements. That is, each agent can either measure all the relative positions of the other agents with respect to itself, or it cannot measure anything.

As in Example 5.10, we will neglect the probability that two (or more) different failures can simultaneously be repaired, simultaneously occur, or simultaneously occur and be repaired. In principle all different failures can occur with the same probability. If a failure occurs, the probability of it being repaired after an uncertain residence time, $T_{(\hat{\imath})} = T_{(\hat{\imath})}^{min} + \Delta T_{(\hat{\imath})}$, is 50%. The sum of the probabilities that other different failure occurs is then also 50%. The probabilities of jumping between modes can be stated in a matrix $\mathbf{\Pi} = \mathrm{col}\left\{(\boldsymbol{\pi}_{\hat{\imath}}^{+})'\right\}_{\hat{\imath}\in Q} \in \mathbb{R}^{M\times M}$. We will assume that the residence time at each discrete mode $\hat{\imath}$ depends on the number of failures, in such a way that $T_{(\hat{\imath})}^{min} = 3.0(0.1)^{f_{(\hat{\imath})}}$ and $\Delta T_{(\hat{\imath})}^{max} = T_{(\hat{\imath})}^{min}/4$, where $f_{(\hat{\imath})} \in \mathbb{N}_0$ is the number of failures at mode $\hat{\imath}$. The only exception is when there are no failures, the nominal mode $\hat{\imath} = 1$, where we will consider that $\Delta T_{(\hat{\imath})}^{max} = 0$, *i.e.* the nominal mode has an known and certain residence time of $T_{(\hat{\imath})} = 3.0$.

If we consider that more than one agent can be in a failure mode at the same time, we need to define

$$M = \sum_{i=0}^{N} \left({}^{N}C_{i}\right) = 2^{N} = 1,024$$

different discrete operation modes $\hat{\imath} \in Q$. Considering Theorem 3.7, this implies that there are 1,024 LMI restrictions. Furthermore, given the dimension of the matrices involved ($\tilde{\mathbf{A}} \in \mathbb{R}^{38\times38} \implies \mathbf{P}(\boldsymbol{\alpha}) \in \mathbb{R}^{38\times38}$) each monomial matrix has $38(38+1)/2 = 741$ scalar variables! The number of variables involved can be very difficult to handle by any solver. One way of lighten the burden of the solver is to impose that all variable matrices are diagonal (in this case, each monomial will be described only by 38 variables). Additionally, the other variables defined in Theorem 3.7 can also be forced to have a diagonal structure. This is however, in many cases not enough to obtain a solution in practical time.

To show the numeric complexity of the problem, consider the different situations summarized in Table 7.3 for networks with less agents but under equivalent failure switching

Table 7.3.: Feasibility Results for Example 7.5.

N	$\dim\{\tilde{\mathbf{A}}\}$	Max. Failures	M	Scalar variables	App. Solver Time	Feasibility
4	14	4	16	659	$3[s]$	Yes
5	18	5	32	$1,723$	$6[s]$	Yes
6	22	6	64	$4,243$	$23[s]$	Yes
7	26	2	29	$2,238$	$13[s]$	Yes
7	26	7	128	$10,059$	$122[s]$	Yes
8	30	2	37	$3,306$	$30[s]$	Yes
9	34	2	46	$4,669$	$60[s]$	Yes
10	38	1	11	$1,188$	$8[s]$	Yes
10	38	2	56	$6,363$	$121[s]$	Yes
10	38	10	$1,024$	$117,683$	?	?

schemes. For networks with less than eight agents, we consider that all agents can fail simultaneously. With eight or more agents, we consider only that a limited number of agents can fail at the same time. Note that the time needed to solve the feasibility problem proposed by Theorem 3.7 depends mainly on the number of scalar variables. This depends at the same time on the number of discrete modes M but also on the dimension of the variables ($\dim\{\tilde{\mathbf{A}}\} = Nn + (N-1)q = 4N-2$). Furthermore, the time also depends on the efficiency of the implementation of the problem and the characteristics of the underlaying software and hardware. This makes it very difficult to predict the time needed to solve the problem when all ten agents can fail. Note that then, there are more than 117 thousand scalar variables while in the worst case successfully solved, the maximum number of variables is around 10 thousand.

However, note that in all the studied cases, the problems are feasible. This gives space to speculate that the problem with ten agents and ten simultaneous failures is also feasible and that the network can achieve formation through the specified failure switching law. This is corroborated by Figures 7.11 and 7.12 where the behavior of the network is simulated under exactly the same conditions as in the previous example but considering a random switching sequence with the defined characteristics. It can be seen that the specified formations can still be reached under this failure configuration but with slightly different trajectories as in the ideal case. ■

7.3.2. Measurement delays

Another interesting topic is delayed measurements. That is when the relative measurements of an agent presents some time delay when used for feedback. For example, this can happen if the relative position measurements are taken with some kind of optical device and processing

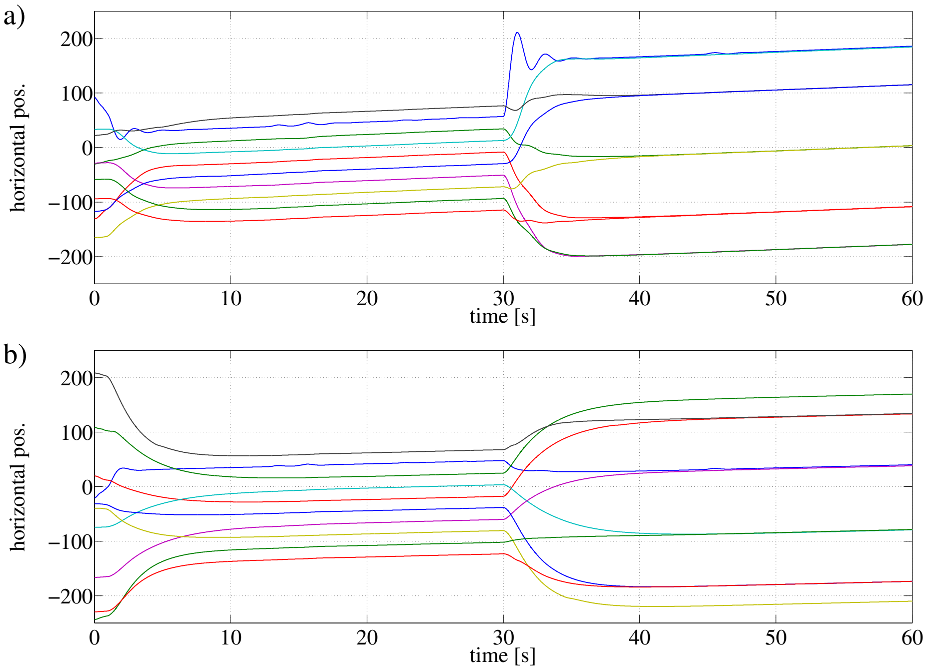


Figure 7.11.: Temporal trajectories under $M = 1{,}024$ possible failure states in Example 7.5.

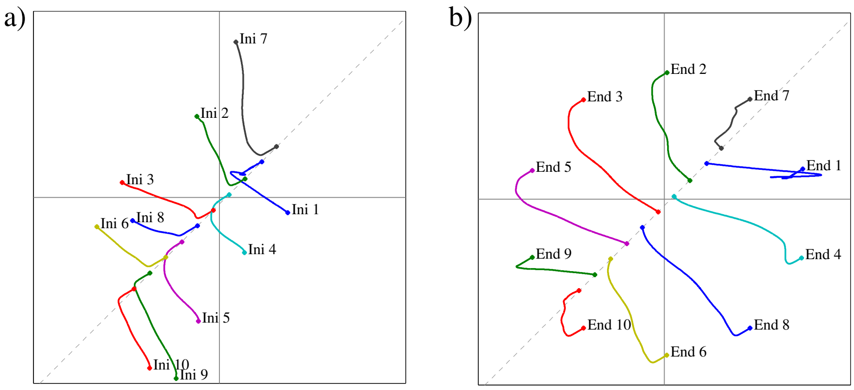


Figure 7.12.: Spatial trajectories under $M = 1{,}024$ possible failure states in Example 7.5. a) $t \leqslant 30$, b) $30 < t \leqslant 60$.

is needed to estimate the relative distances between the agents. Consider that, for every agent $i \in \mathscr{V}$, a delay $\theta_i \in \mathbb{R}^+$, with $\theta_i \neq \theta_j$, $i, j \in \mathscr{V}$, is associated so the feedback signal can be locally computed as

$$\mathbf{u}_i(t) = \mathbf{L}_i \mathbf{y}_i(t) + \mathbf{h}_i' \mathbf{e}_i(t-\theta_i) - \mathbf{h}_i' \mathbf{p}_i,$$

implying that

$$\begin{aligned}\mathbf{u}(t) &= \mathbf{L}\mathbf{y}(t) + \left(\sum_{i \in \mathscr{V}} \mathbf{s}_i \mathbf{h}_i' \mathbf{T}_i \mathbf{T}^+ \left(\mathbf{e}_0(t-\theta_i) - \mathbf{p} \right) \right) \\ &= \mathbf{L}\mathbf{y}(t) + \left(\sum_{i \in \mathscr{V}} \mathbf{s}_i \mathbf{h}_i' \mathbf{T}_i \mathbf{T}^+ \mathbf{e}(t-\theta_i) \right).\end{aligned}$$

From here, the dynamics of the error and the states of the agents can be written as:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{e}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}+\mathbf{BLC} & \mathbf{0} \\ \mathbf{TC} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{e} \end{bmatrix} + \sum_{i \in \mathscr{V}} \begin{bmatrix} \mathbf{0} & \mathbf{B}\left(\mathbf{s}_i \mathbf{h}_i' \mathbf{T}_i\right) \mathbf{T}^+ \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}(t-\theta_i) \\ \mathbf{e}(t-\theta_i) \end{bmatrix}.$$

Note that the reference vectors $\mathbf{p}_i$ and $\mathbf{p}$ are considered to be constant and the external input is omitted. If some agents have exactly the same delay, or some have no delay, similar simplified expressions can also be obtained. This can be compactly written as,

$$\dot{\tilde{\mathbf{x}}} = \tilde{\mathbf{A}}_0 \tilde{\mathbf{x}} + \sum_{i \in \mathscr{V}} \tilde{\mathbf{A}}_i \tilde{\mathbf{x}}_{\theta_i}, \tag{7.21}$$

with,

$$\begin{aligned}&\tilde{\mathbf{x}} = \operatorname{col}\{\mathbf{x}, \mathbf{e}\}, \ \tilde{\mathbf{x}}_{\theta_i}(t) = \tilde{\mathbf{x}}(t-\theta_i), \\ &\tilde{\mathbf{A}}_0 = \begin{bmatrix} \mathbf{A}+\mathbf{BLC} & \mathbf{0} \\ \mathbf{TC} & \mathbf{0} \end{bmatrix}, \ \tilde{\mathbf{A}}_i = \begin{bmatrix} \mathbf{0} & \mathbf{B}\left(\mathbf{s}_i \mathbf{h}_i' \mathbf{T}_i\right) \mathbf{T}^+ \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.\end{aligned}$$

Note that these matrices are extremely sparse. This property can be used to lighten numeric operations between them.

A way to deal with delayed systems in an LMI framework is through Lyapunov-Krasovskii functions, *e.g.* [85, 92–94, 97, 110, 114]. Consider the following positive definite function:

$$v = \tilde{\mathbf{x}}' \mathbf{P}_0 \tilde{\mathbf{x}} + \sum_{i \in \mathscr{V}} \left(\int_{t-\theta_i}^{t} \tilde{\mathbf{x}}'(\alpha) \mathbf{P}_i^s \tilde{\mathbf{x}}(\alpha) d\alpha + \int_{-\theta_i}^{0} \int_{t+\beta}^{t} \dot{\tilde{\mathbf{x}}}'(\alpha) \mathbf{P}_i^d \dot{\tilde{\mathbf{x}}}(\alpha) d\alpha d\beta \right), \tag{7.22}$$

with $\mathbf{P}_0 = \mathbf{P}_0' > 0$; $\forall i \in \mathscr{V}$, $\mathbf{P}_i^s = (\mathbf{P}_i^s)' \geqslant 0$ and $\mathbf{P}_i^d = (\mathbf{P}_i^d)' \geqslant 0$ of proper dimension; and time derivative

$$\dot{v} = \tilde{\mathbf{x}}' \mathbf{P}_0 \dot{\tilde{\mathbf{x}}} + \dot{\tilde{\mathbf{x}}}' \mathbf{P}_0 \tilde{\mathbf{x}} + \sum_{i \in \mathscr{V}} \left(\tilde{\mathbf{x}}' \mathbf{P}_i^s \tilde{\mathbf{x}} - \tilde{\mathbf{x}}_{\theta_i}' \mathbf{P}_i^s \tilde{\mathbf{x}}_{\theta_i} + \theta_i \dot{\tilde{\mathbf{x}}}' \mathbf{P}_i^d \dot{\tilde{\mathbf{x}}} - \int_{t-\theta_i}^{t} \dot{\tilde{\mathbf{x}}}'(\alpha) \mathbf{P}_i^d \dot{\tilde{\mathbf{x}}}(\alpha) d\alpha \right).$$

From equation (A.11) in the Appendix we can write that

$$-\theta_i \int_{t-\theta_i}^{t} \dot{\tilde{\mathbf{x}}}'(\alpha)\mathbf{P}_i^d\dot{\tilde{\mathbf{x}}}(\alpha)d\alpha \leqslant -\left(\int_{t-\theta_i}^{t} \dot{\tilde{\mathbf{x}}}(\alpha)d\alpha\right)'\mathbf{P}_i^d\left(\int_{t-\theta_i}^{t} \dot{\tilde{\mathbf{x}}}(\alpha)d\alpha\right) = -\left(\tilde{\mathbf{x}}-\tilde{\mathbf{x}}_{\theta_i}\right)'\mathbf{P}_i^d\left(\tilde{\mathbf{x}}-\tilde{\mathbf{x}}_{\theta_i}\right),$$

implying that,

$$\dot{v} \leqslant \dot{v}_b = \tilde{\mathbf{x}}'\mathbf{P}_0\dot{\tilde{\mathbf{x}}} + \dot{\tilde{\mathbf{x}}}'\mathbf{P}_0\tilde{\mathbf{x}} + \sum_{i\in\mathscr{V}}\left(\tilde{\mathbf{x}}'\mathbf{P}_i^s\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_{\theta_i}'\mathbf{P}_i^s\tilde{\mathbf{x}}_{\theta_i} + \theta_i\dot{\tilde{\mathbf{x}}}'\mathbf{P}_i^d\dot{\tilde{\mathbf{x}}} - \frac{1}{\theta_i}\left(\tilde{\mathbf{x}}-\tilde{\mathbf{x}}_{\theta_i}\right)'\mathbf{P}_i^d\left(\tilde{\mathbf{x}}-\tilde{\mathbf{x}}_{\theta_i}\right)\right).$$

Imposing $\dot{v}_b < 0$, leads to the sufficient condition for $\dot{v} < 0$. Replacing the expression for $\dot{\tilde{\mathbf{x}}}$ and rewriting the previous inequality into the form $\dot{v}_b = \mathfrak{v}'\mathfrak{D}\mathfrak{v}$, with $\mathfrak{v} = \mathrm{col}\left\{\tilde{\mathbf{x}}, \mathrm{col}\left\{\tilde{\mathbf{x}}_{\theta_i}\right\}_{i\in\mathscr{V}}\right\}$, stability of the states of the agents and consensus can be studied through LMI (7.23),

$$\mathfrak{D} = \left[\begin{array}{c|cccccc} \mathfrak{D}_{00} & \mathfrak{D}_{01} & \cdots & \mathfrak{D}_{0i} & \cdots & \mathfrak{D}_{0N} \\ \hline \star & \mathfrak{D}_{11} & \cdots & \mathfrak{D}_{ij} & \cdots & \mathfrak{D}_{1N} \\ \vdots & \vdots & \ddots & & & \vdots \\ \star & \star & & \mathfrak{D}_{ii} & & \mathfrak{D}_{iN} \\ \vdots & \vdots & & & \ddots & \vdots \\ \star & \star & \cdots & \star & \cdots & \mathfrak{D}_{NN} \end{array}\right] < 0, \tag{7.23}$$

where $\forall i \in \mathscr{V}$ and $\forall j \in \{i+1, i+2, \ldots, N\}$ each block is defined by:

$$\begin{aligned}
\mathfrak{D}_{00} &= \mathbf{P}_0\tilde{\mathbf{A}}_0 + \tilde{\mathbf{A}}_0'\mathbf{P}_0 + \sum_{i\in\mathscr{V}}\left(\mathbf{P}_i^s - \frac{1}{\theta_i}\mathbf{P}_i^d\right) + \tilde{\mathbf{A}}_0'\mathbf{P}^d\tilde{\mathbf{A}}_0, \\
\mathfrak{D}_{0i} &= \mathbf{P}_0\tilde{\mathbf{A}}_i + \tilde{\mathbf{A}}_0'\mathbf{P}^d\tilde{\mathbf{A}}_i + \frac{1}{\theta_i}\mathbf{P}_i^d, \\
\mathfrak{D}_{ii} &= -\mathbf{P}_i^s - \frac{1}{\theta_i}\mathbf{P}_i^d + \tilde{\mathbf{A}}_i'\mathbf{P}^d\tilde{\mathbf{A}}_i, \\
\mathfrak{D}_{ij} &= \tilde{\mathbf{A}}_i'\mathbf{P}^d\tilde{\mathbf{A}}_j,
\end{aligned}$$

and $\mathbf{P}^d = \sum_{i\in\mathscr{V}}\theta_i\mathbf{P}_i^d$.

Note that if for any $i \in \mathscr{V}$ in the Lyapunov function (7.22), $\mathbf{P}_i^d = \mathbf{0}$, then the derived conditions do not depend on the value θ_i. That is, when this value is not known, LMI (7.23) can be used to check stability of $\tilde{\mathbf{x}}$ by forcing the respective matrices to be zero. If the delays are known to be within certain interval but are not precisely known, LMI (7.23) can be modified to also study these cases. This of course implies more restrictive numeric conditions.

Also observe that the matrix in LMI (7.23) is composed of $(N+1)^2$ blocks, each of them of dimension $(Nn+(N-1)q) \times (Nn+(N-1)q)$. Therefore,

$$\mathfrak{D} \in \mathbb{R}^{((N+1)(Nn+(N-1)q))\times((N+1)(Nn+(N-1)q))}$$

can be difficult to handle numerically due to the high dimension calculations involved. Furthermore, the Lyapunov matrices $\mathbf{P}_0$, $\mathbf{P}_i^s$ and $\mathbf{P}_i^d$, with $i \in \mathscr{V}$, are also of high dimension (they are defined in $\mathbb{R}^{(Nn+(N-1)q)\times(Nn+(N-1)q)}$) what makes the LMI problem also difficult to handle for most solvers. These aspects need to be considered for an efficient implementation of the LMI based test for stability of the delayed vector $\tilde{\mathbf{x}}$.

Example 7.6. Consider the same $N = 10$ agents that in the previous examples. Each agent presents a delay in their relative position measurement given by

$$\theta_1 = 0.0990,\ \ \theta_2 = 0.0992,\ \ \theta_3 = 0.0994,\ \ \theta_4 = 0.0997,\ \ \theta_5 = 0.0999,$$
$$\theta_6 = 0.1001,\ \ \theta_7 = 0.1003,\ \ \theta_8 = 0.1006,\ \ \theta_9 = 0.1008,\ \ \theta_{10} = 0.1010.$$

Note that the mean value of these values is $E(\theta) = \frac{1}{N}\sum_{i\in\mathscr{V}} \theta_i = 0.1$. In this case, LMI (7.23) is such that $\mathfrak{D} \in \mathbb{R}^{418\times418}$ and the Lyapunov function is defined by $1+10\times2 = 21$ matrices in $\mathbb{R}^{38\times38}$. As in the previous example, this involves a considerable large number of scalar variables ($21\times38(38+1)/2 = 15,561$) what makes it difficult to handle the problem. To diminish this number, diagonal structures of the Lyapunov matrices cannot be assumed because the structure of matrices $\tilde{\mathbf{A}}_0$ and $\tilde{\mathbf{A}}_i$, $\forall i \in \mathscr{V}$, with a zero block in position $(2,2)$, makes the diagonal structure too restrictive. However, symmetric Toeplitz[1] matrices can be used to have a total of $21\times38 = 798$ scalar variables.

With this, LMI (7.23) can be verified feasible with the stated delays. This is confirmed in Figure 7.13 where the network response is simulated under the same conditions of the previous examples. Note that the delays in the measurements make the agents positions to oscillate heavily. However, this oscillation is not strong enough to break the formation and to destabilize the internal dynamic of the agents.

Consider now a more demanding scenario with the following values:

$$\theta_1 = 0.2970,\ \ \theta_2 = 0.2977,\ \ \theta_3 = 0.2983,\ \ \theta_4 = 0.2990,\ \ \theta_5 = 0.2997,$$
$$\theta_6 = 0.3003,\ \ \theta_7 = 0.3010,\ \ \theta_8 = 0.3017,\ \ \theta_9 = 0.3023,\ \ \theta_{10} = 0.3030,$$

with $E(\theta) = 0.3$. In this case, LMI (7.23) cannot be verified feasible. Because this is only a sufficient condition for stability and because of the special structure chosen for the Lyapunov matrices, this does not imply that the network under these conditions cannot reach formation successfully. Nevertheless, it does give space to suspect instability. Indeed, considering the same simulation situation, Figure 7.14 shows that the oscillation of the agents makes it impossible to achieve formation. ■

[1] A Toeplitz matrix is a matrix in which each descending diagonal from left to right is constant. For example, the following matrix is Toeplitz and symmetric:

$$\begin{bmatrix} a & b & c & d & e \\ b & a & b & c & d \\ c & b & a & b & c \\ d & c & b & a & b \\ e & d & c & b & a \end{bmatrix}.$$

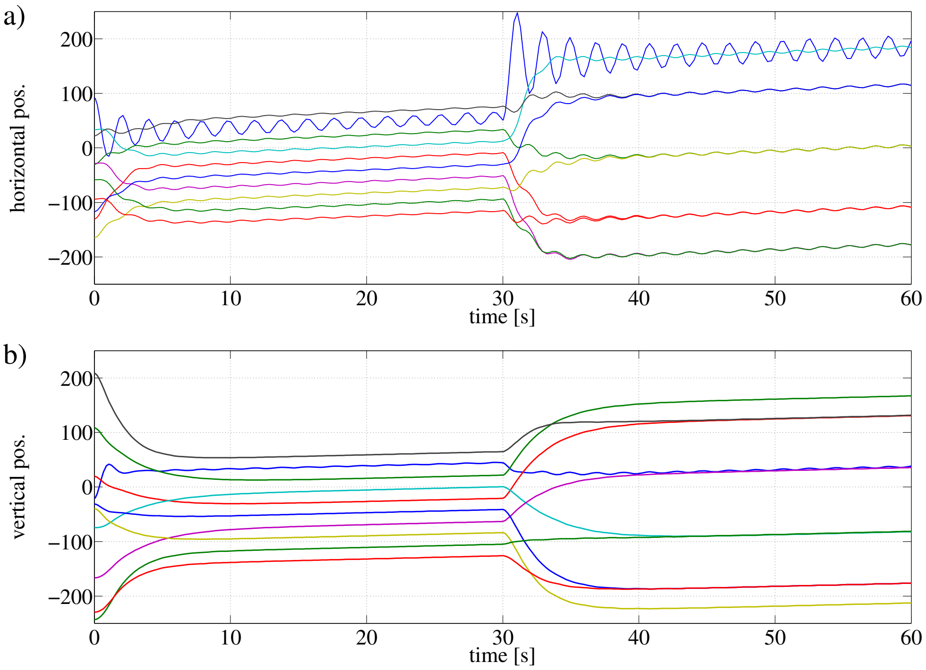


Figure 7.13.: Temporal trajectories with $E(\theta) = 0.1$ in Example 7.6.

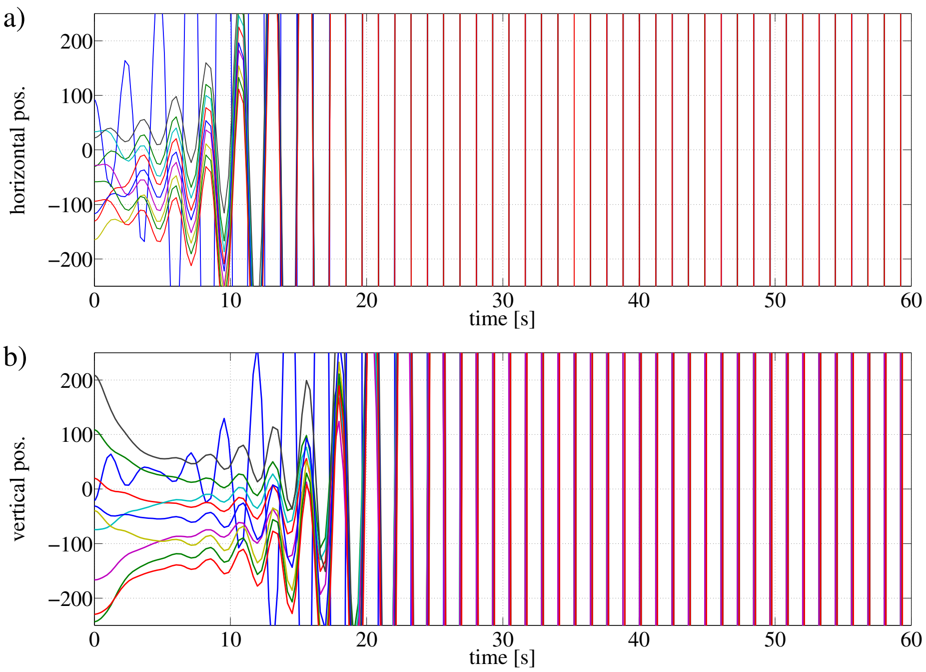


Figure 7.14.: Temporal trajectories with $E(\theta) = 0.3$ in Example 7.6.

7.3.3. Discussion

Most of the cases studied in this thesis and, in particular, formation control, give rise to a number of questions about the nature of the studied problems. Up to this point, we have treated the consensus and formation problems from a global perspective. That is, all analyses have been done considering that the whole network is known to the analyzer, even though the control actions can be distributed in the local controllers of the agents. This, of course, can only be done in small networks where each agent is seen as an element of a larger system. However, in more complex cases, the analysis of the network can only be done locally, either by one of the agents itself or by a local controller for a small section of a larger network. This gives space to talk about an *individual perspective* of the consensus problem.

In this case, the agent that performs the analysis can only have partial information of the network, typically limited to the dynamical behavior of its “neighbors”. In general, the agent can also not assume that it has the power to influence other agents. It can only do its best to stay “near” them.

When an agent looks at its neighbors, how does it evaluate if consensus is reached or not? He can certainly not evaluate the error vector for the whole network but only for a small section, probably described by a star tree from wich the agent is the center. In this sense, the idea of organization is no longer valid as before for the whole network, but only as sub organizations for each agent who evaluates different target errors. However, if a global approach cannot be followed, then how can we evaluate whether the network reached consensus or not?

Another aspect to take into consideration is the number of neighbors that each agent has. This might be a fixed number or a time variant quantity. Probably, the number of neighbors that an agent can “see” will tend to increase as the agents are near to consensus or formation. Can each agent manage information from as many neighbors as it might have? Or is it limited to a maximal number of neighbors? Then, how does an agent choose its observed neighbors? Which other agents are actually neighbor candidates?

How to deal with such questions in large networks is neither clear nor immediate. Probably, at each instant, the agent will be tempted to solve an optimization problem in order to minimize the error with respect to the reported information about the neighbors. How to do this? What functional should be optimized? This could also be seen as a game theoretical problem. However, most game theoretical approaches deal with a restricted number of strategies. On the contrary, in the control case, an infinite number of strategies is usually considered: continuous values for the input vectors of each agent.

This kind of questions leads to a richer understanding of consensus, distributed control, and its associated implementation problems. To study this scenario, the fundamental assumption that consensus is a design objective needs to be dropped, as there is no way to compute how close is the network to consensus if the whole network is not known.

8 Conclusion

8.1. Summary

In this thesis, a comprehensive analysis of the consensus problem is formulated from a control theoretical perspective. First, a general framework on graph theory and stability of systems is summarized in Chapter 2. This is complemented with some mathematical concepts treated in Appendix A. This mathematical background is closed in Chapter 3 with results on the stability of time triggered switched systems.

After this background is stated, Chapter 4 defines the consensus problem as a control objective, independent of the characteristics of the agents, through the key concept of hierarchical organization. From here, Chapter 5 studies different choices for feedback laws so that the plant achieves this control objective. The first section of this chapter is dedicated to Laplacian algorithms, which are the most popular in consensus references. Here we distinguish between integrator networks and autonomous agents networks to study consensus accordingly. The second section of this chapter deals with different kind of algorithms and describes their main characteristics when applied to different networks. Finally, a third section is focused on different assumptions on the dynamics of the agents in an attempt to generalize the analysis to a larger class of networks.

Chapters 6 and 7 apply the previously discussed concepts to two related problems. The first one is that of power sharing in electric grids with inverters as generation units. This chapter is in strict relationship with Appendix B. In the following chapter, the formation problem is described as a higher order consensus problem and the distributed implementation of a controller to achieve an optimal formation is studied from a centralized perspective.

8.2. Contributions

The main contribution of this thesis is to bring the consensus problem closer to classical Control Theory. That is, to describe the problem in three different and independent entities: the network of agents, the consensus objective defined through the idea of organization, and the consensus algorithm as an output feedback. This description allows to systematically analyze consensus with well known tools from control theory. In this way, consensus can be studied by an equivalent stability problem under several assumptions, generalizing the class of systems and controllers that can be considered, but maintaining a global framework that can be used to further investigate different cases and applications.

Nevertheless, there are also a number of specific contributions related to the different chapters of the thesis that also need to be mentioned. Concerning graph theory, it is important to mention the generalization of weighted graphs to include higher dimension weights. This is a small contribution to the graph theoretical field, but it is a key aspect for the consensus analysis, as it allows to study higher order coupled agents without losing the convenient graphical representation of the algorithms.

The results on switched systems in Chapter 3 are also worth mentioning. The main contribution in this aspect is the extension of known results on stability of stochastic switched systems to include uncertainties on the implicit discrete time system matrices. These results can be directly applied to the study of consensus.

As already mentioned, the consensus formulation stated in Chapter 4 can be regarded as the main contribution of this work. From here, several results can be deducted in the following chapter. For the case of integrator networks, sufficient and necessary conditions for consensus can be achieved even in the case where the dynamics of the agents are coupled. These conditions can be further modified to study consensus under the influence of perturbations and uncertainties, and to define design procedures that consider performance criteria as robustness or convergence rate.

But it is in the case of autonomous agents where the most contributions in the consensus area are achieved. Particularly, we found necessary and sufficient conditions for reaching consensus in a wide range of networks. Namely, when the residual signal vanishes. These conditions can be again easily extended to study the roll of perturbations and uncertainties and to define design procedures subject to some performance criteria. As far as the author knows, output consensus between agents with arbitrary linear dynamics has not been characterized before independently of the states dynamics.

To avoid the problem of residual signals, several alternative algorithms were studied. First, we propose the use of selflooped algorithms to diminish the residual effect on the network. As in general this is not possible, we consider further a complementary state feedback that may not be necessarily described in a graph theoretical way. Further in this direction, non graphically restricted algorithms are studied. This helps to stress the point that consensus does not necessarily depend on the graphical description of the algorithms but on the relationship between the different matrices that describes the network and the algorithm.

Switched algorithms are also addressed through the previously discussed results on switched systems. The use of such algorithms is highlighted to describe the operation of the network under communication failure. It is shown that, again because of the residual signals, it is difficult to guarantee consensus in all cases. However, under some operation conditions, consensus can still be reached with this kind of strategies.

The final section of Chapter 5, further expands the discussion to networks with different dynamical assumptions. The emphasis here is to enlarge the class of dynamic phenomena that can be considered rather than to propose new control strategies. In particular, it is shown that higher order dynamics can be studied by imposing stability of not only the consensus error, but also of other auxiliary variables. The discussion on the role of communications, as an independent control entity of the consensus algorithm, opens space for interesting future developments.

Chapter 6 on microgrids is an application of the previously discussed topics and well studied power electric concepts. From a consensus perspective, its main contribution is to exemplify the role of hardware interconnections between the agents and how they can influence the control strategy to be used. In particular, it is shown that selflooped algorithms can be successfully applied to such networks. From an electrical point of view, the definitions of the grid components and of their control strategies are borrowed from the available references. However, the use of the explicit model of the grid, deducted in Appendix B, for design and evaluation of control strategies, is usually not to be found in related works.

The formation problem chapter exemplifies the consensus analysis of higher order networks. It considers a coherent description of the problem to define a feedback strategy that guarantees that any formation can be reached successfully without the need of exchanging speed information between the agents. It only requires access to relative position measurements. It also describes a novel methodology to calculate an optimal formation given the current relative position of the agents. Based on local measurements, both elements can be distributively implemented in the agents. This leaves space for a centralized analysis of the network subject to a distributed implementation of the control strategy, which includes switching faults and delayed measurements.

8.3. Future Work

Further development in several parts of this work can be considered. Principally in three different areas: switched systems, general consensus, and the application to electric systems.

Firstly, regarding to stochastic switched systems, the possibility of verifying stability by observing the systems behavior at switching instants only, opens a wide spectrum of opportunities to study the behavior of time triggered switched systems under different assumptions. Nevertheless, only sufficient conditions to test stability are shown based on a particular choice of Lyapunov functions. To extend the analysis to find less conservative restrictions, favoring other kind of Lyapunov functions derived, for example, from homogeneous polynomials, is a pending matter. Furthermore, only time triggered switching schemes are considered. It seems natural to proceed to more sophisticated scenarios where switching can occur as a function of the continuous states of the system. This is not a trivial task as it is difficult to, in general, describe switching regions to analyze the systems. A possible way to proceed is to extend the studied approach to continuous time Markov chains and define the probability vectors as a function of the continuous states of the system.

From the consensus point of view, the main difficulty for convergence of the error signal is the influence of the states of the agents on the dynamics of the consensus error through what have been denominated the residual signal. An interesting idea is to characterize this influence through an scalar indicator in order to be able to compare different strategies and networks.

The underlaying hypothesis is that this indicator exists as a function of the information of the network and the controller only, and not of the time response of the network. This approach can be used to further analyze other topics, as the influence of communication dynamics in the consensus objective. A further situation to be studied is that of distributed analysis discussed at the end of Chapter 7.

In the electric grid case, an interesting topic is to include different dynamics for the generation units. Particularly, to generalize the study to the case where some agents model VSIs but others represent classical rotatory machine generators. This situation is nowadays an important issue as the current state of distributed electric generation considers a hybrid scheme where both technologies coexist. Furthermore, a unified analysis of both kinds of power sharing (active and reactive) is needed. Both quantities are highly interconnected and an independent analysis of each one can be restrictive. A third topic in this aspect is to change the view of the problem from a mainly consensus theoretical one to an applied one. For this, the model and control of a real grid seems as an interesting further step.

Appendices

A
Other General Results

A.1. Pseudo Inverses

Generalized inverses, more specifically the Moore-Penrose pseudoinverse, are an important and well studied subject in linear algebra. Some of the definitions and properties associated with this topic are however some times overpassed. Therefore we include here a brief introduction on the subject based on the book [115].

Definition A.1.1. Given a matrix $\mathbf{A} \in \mathbb{C}^{m\times n}$, its *Moore-Penrose pseudoinverse* (or just *pseudoinverse* for short) $\mathbf{A}^+ \in \mathbb{C}^{n\times m}$ satisfies the *Penrose equations*:

$$\mathbf{A}\mathbf{A}^+\mathbf{A} = \mathbf{A}, \tag{A.1}$$
$$\mathbf{A}^+\mathbf{A}\mathbf{A}^+ = \mathbf{A}^+, \tag{A.2}$$
$$(\mathbf{A}\mathbf{A}^+)^* = \mathbf{A}\mathbf{A}^+, \tag{A.3}$$
$$(\mathbf{A}^+\mathbf{A})^* = \mathbf{A}^+\mathbf{A}. \tag{A.4}$$

where $\mathbf{A}^*$ is the conjugate transpose of $\mathbf{A}$.

The name of this generalized inverse comes from the work of R. Penrose who showed in 1955 that this matrix is unique and of E. H. Moore who previously studied the operation though defined in a different way.

Lemma A.1 (Penrose)**.** *For every* $\mathbf{A} \in \mathbb{C}^{m\times n}$*, if* $\mathbf{A}^+$ *exits, then it is unique.*

Proof. Let $\mathbf{X}$ and $\mathbf{Y}$ be different matrices that satisfy the four Penrose equations. Then,

$$\begin{aligned}
\mathbf{X} &= \mathbf{X}(\mathbf{A}\mathbf{X}) = \mathbf{X}(\mathbf{A}\mathbf{X})^* = \mathbf{X}\mathbf{X}^*\mathbf{A}^* = \mathbf{X}\mathbf{X}^*(\mathbf{A}\mathbf{Y}\mathbf{A})^* = \mathbf{X}\mathbf{X}^*\mathbf{A}^*\mathbf{Y}^*\mathbf{A}^* \\
&= \mathbf{X}(\mathbf{A}\mathbf{X})^*(\mathbf{A}\mathbf{Y})^* = \mathbf{X}(\mathbf{A}\mathbf{Y})^* = (\mathbf{X}\mathbf{A})\mathbf{Y} \\
&= (\mathbf{X}\mathbf{A})^*(\mathbf{Y}\mathbf{A})^*\mathbf{Y} = (\mathbf{A}^*\mathbf{X}^*\mathbf{A}^*)\mathbf{Y}^*\mathbf{Y} = \mathbf{A}^*\mathbf{Y}^*\mathbf{Y} \\
&= (\mathbf{Y}\mathbf{A})^*\mathbf{Y} = (\mathbf{Y}\mathbf{A})\mathbf{Y} \\
&= \mathbf{Y}
\end{aligned}$$

What contradicts the statement that $\mathbf{X}$ and $\mathbf{Y}$ are different. □

It can also be shown that for every matrix $\mathbf{A} \in \mathbb{C}^{m\times n}$ the pseudoinverse always exists. Indeed, if $r = \operatorname{rank}\{\mathbf{A}\} = 0 \iff \mathbf{A} = \mathbf{0}_{m\times n}$, then $\mathbf{A}^+ = \mathbf{0}_{n\times m}$ satisfies the Penrose equations. For $0 < r = \operatorname{rank}\{\mathbf{A}\} \leqslant \min\{n,m\}$ the following lemma gives an explicit formula for the pseudoinverse.

Lemma A.2 (MacDuffee)**.** *For* $\mathbf{A} \in \mathbb{C}^{m\times n}$ *with* $r = \text{rank}\{\mathbf{A}\} > 0$*, and a full rank factorization*

$$\mathbf{A} = \mathbf{F}\mathbf{G}, \tag{A.5}$$

where $\mathbf{F} \in \mathbb{C}^{m\times r}$ *and* $\mathbf{G} \in \mathbb{C}^{r\times n}$ *are full rank matrices, then*

$$\mathbf{A}^+ = \mathbf{G}^*(\mathbf{F}^*\mathbf{A}\mathbf{G}^*)^{-1}\mathbf{F}^*. \tag{A.6}$$

Proof. First we need to prove that $\mathbf{F}^*\mathbf{A}\mathbf{G}^*$ is nonsingular. By (A.5),

$$\mathbf{F}^*\mathbf{A}\mathbf{G}^* = (\mathbf{F}^*\mathbf{F})(\mathbf{G}\mathbf{G}^*),$$

and both factors on the right hand term are $r \times r$ matrices with rank r. Then $\mathbf{F}^*\mathbf{A}\mathbf{G}^*$ can be expressed as the multiplication of two full rank matrices and is then also full rank and nonsingular. Its inverse is then:

$$(\mathbf{F}^*\mathbf{A}\mathbf{G}^*)^{-1} = (\mathbf{G}\mathbf{G}^*)^{-1}(\mathbf{F}^*\mathbf{F})^{-1}.$$

By inspection, it is easy to show that $\mathbf{A}^+ = \mathbf{G}^*(\mathbf{G}\mathbf{G}^*)^{-1}(\mathbf{F}^*\mathbf{F})^{-1}\mathbf{F}^*$ satisfies the Penrose equations and therefore is the unique pseudoinverse of matrix $\mathbf{A}$. □

As for every matrix of rank $r > 0$, a full rank factorization can be found, then the previous lemma also shows that every matrix has a pseudoinverse. From here, the following corollary is immediate by making one of the full rank factors of the matrices equal to the identity.

Corollary A.3. *If* $\mathbf{B} \in \mathbb{C}^{m\times r}$ *is full column rank (i.e.* $\text{rank}\{\mathbf{B}\} = r$*) and* $\mathbf{C} \in \mathbb{C}^{r\times n}$ *is full row rank (i.e.* $\text{rank}\{\mathbf{C}\} = r$*), their respective pseudo inverses are:*

$$\mathbf{B}^+ = (\mathbf{B}^*\mathbf{B})^{-1}\mathbf{B}^* \textit{ and}$$
$$\mathbf{C}^+ = \mathbf{C}^*(\mathbf{C}\mathbf{C}^*)^{-1}.$$

A pseudo inverse in the shape of $\mathbf{B}^+ = (\mathbf{B}^*\mathbf{B})^{-1}\mathbf{B}^*$ will be addressed as *left* pseudoinverse, as $\mathbf{B}^+\mathbf{B} = \mathbf{I}$, and in the shape of $\mathbf{C}^+ = \mathbf{C}^*(\mathbf{C}\mathbf{C}^*)^{-1}$, as *right* pseudoinverse as $\mathbf{C}\mathbf{C}^+ = \mathbf{I}$. Moreover, the following facts are also easy to prove:

Proposition A.4. *If* $\mathbf{A} \in \mathbb{C}^{n\times n}$ *is nonsingular (i.e.* $\text{rank}\{\mathbf{A}\} = n$*), then its pseudo inverse is* $\mathbf{A}^+ = \mathbf{A}^{-1}$.

Proof. The right pseudoinverse of $\mathbf{A}$ is $\mathbf{A}^+ = (\mathbf{A}^*\mathbf{A})^{-1}\mathbf{A}^* = (\mathbf{A})^{-1}(\mathbf{A}^*)^{-1}\mathbf{A}^* = \mathbf{A}^{-1}$. Alternatively, it can be shown that the inverse matrix satisfies the Penrose equations. □

Proposition A.5. *For any matrix* $\mathbf{X} \in \mathbb{C}^{m\times n}$*, if* $\mathbf{U} \in \mathbb{C}^{m\times m}$ *and* $\mathbf{V} \in \mathbb{C}^{n\times n}$ *are unitary matrices (i.e.,* $\mathbf{U}\mathbf{U}^* = \mathbf{U}^*\mathbf{U} = \mathbf{I}$ *and* $\mathbf{V}\mathbf{V}^* = \mathbf{V}^*\mathbf{V} = \mathbf{I}$*) then,*

$$(\mathbf{U}\mathbf{X}\mathbf{V})^+ = \mathbf{V}^*\mathbf{X}^+\mathbf{U}^*.$$

Proof. By inspection, $\mathbf{A} = \mathbf{UXV}$ and $\mathbf{A}^+ = \mathbf{V}^*\mathbf{X}^+\mathbf{U}^*$ satisfy the Penrose equations:

1. $(\mathbf{UXV})\left(\mathbf{V}^*\mathbf{X}^+\mathbf{U}^*\right)(\mathbf{UXV}) = \mathbf{U}(\mathbf{XX}^+\mathbf{X})\mathbf{V} = \mathbf{UXV}$,
2. $\left(\mathbf{V}^*\mathbf{X}^+\mathbf{U}^*\right)(\mathbf{UXV})\left(\mathbf{V}^*\mathbf{X}^+\mathbf{U}^*\right) = \mathbf{V}^*(\mathbf{X}^+\mathbf{XX}^+)\mathbf{U}^* = \mathbf{V}^*\mathbf{X}^+\mathbf{U}^*$,
3. $$\begin{aligned}\left((\mathbf{UXV})(\mathbf{V}^*\mathbf{X}^+\mathbf{U}^*)\right)^* &= \left(\mathbf{UXX}^+\mathbf{U}^*\right)^* = \mathbf{U}\left(\mathbf{XX}^+\right)^*\mathbf{U}^* \\ &= \mathbf{UXX}^+\mathbf{U}^* = (\mathbf{UXV})(\mathbf{V}^*\mathbf{X}^+\mathbf{U}^*),\end{aligned}$$
4. $$\begin{aligned}\left((\mathbf{V}^*\mathbf{X}^+\mathbf{U}^*)(\mathbf{UXV})\right)^* &= \left(\mathbf{V}^*\mathbf{XX}^+\mathbf{V}\right)^* = \mathbf{V}^*\left(\mathbf{XX}^+\right)^*\mathbf{V} \\ &= \mathbf{V}^*\mathbf{XX}^+\mathbf{V} = (\mathbf{V}^*\mathbf{X}^+\mathbf{U}^*)(\mathbf{UXV}).\end{aligned}$$

□

The previous proposition becomes important because of the Singular Value Decomposition (SVD).

Definition A.1.2. The singular values of matrix $\mathbf{A} \in \mathbb{C}^{m\times n}$, with $m \leqslant n$, are the square root of the eigenvalues of matrix $\mathbf{AA}^*$: $\operatorname{svd}\{\mathbf{A}\} = \left\{\sigma \geqslant 0, m \leqslant n | \sigma^2 \in \operatorname{eig}\{\mathbf{AA}^*\}\right\}$. Analogously, if $m \geqslant n$, then the singular values are the square root of the eigenvalues of $\mathbf{A}^*\mathbf{A}$: $\operatorname{svd}\{\mathbf{A}\} = \left\{\sigma \geqslant 0, m \geqslant n | \sigma^2 \in \operatorname{eig}\{\mathbf{A}^*\mathbf{A}\}\right\}$.

Note that the nonzero eigenvalues of $\mathbf{AA}^*$ are the same as the nonzero eigenvalues of $\mathbf{A}^*\mathbf{A}$. The two cases of the definition are then equivalent for nonzero singular values and only introduced so that $|\operatorname{svd}\{\mathbf{A}\}| = \min\{m,n\}$. Also note that if $r = \operatorname{rank}\{\mathbf{A}\} \leqslant \min\{m,n\}$ there are r nonzero singular values that are usually labeled in descending order:

$$\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_r > \sigma_{r+1} = \sigma_{r+2} = \cdots = \sigma_{\min\{m,n\}} = 0.$$

Furthermore, from the definition of positive semi-definite matrices (Section 2.2.2), the LMI $\mathbf{A}'\mathbf{A} \leqslant \sigma_1^2\mathbf{I}$ follows immediately.

Theorem A.6 (Singular Values Decomposition (SVD))**.** *Given* $\mathbf{A} \in \mathbb{C}^{m\times n}$ *with* $r = \operatorname{rank}\{\mathbf{A}\} > 0$ *non zero singular values* $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_r > 0$*, then there are unitary matrices* $\mathbf{U} \in \mathbb{C}^{m\times m}$ *and* $\mathbf{V} \in \mathbb{C}^{n\times n}$ *such that:*

$$\Sigma = \mathbf{U}^*\mathbf{AV} = \begin{bmatrix} \operatorname{diag}\{\sigma_i\}_{i=1}^r & \mathbf{0}_{r\times(n-r)} \\ \mathbf{0}_{(m-r)\times r} & \mathbf{0}_{(m-r)\times(n-r)} \end{bmatrix} \in \mathbb{R}^{m\times n}.$$

Moreover, $\mathbf{U} = \operatorname{row}\{\mathbf{u}_i\}_{i=1}^m$ *and* $\mathbf{V} = \operatorname{row}\{\mathbf{v}_i\}_{i=1}^n$*, where* $\mathbf{u}_i \in \mathbb{C}^m$ *are the normalized eigenvectors of* $\mathbf{AA}^*$ *and* $\mathbf{v}_i \in \mathbb{C}^n$ *the normalized eigenvectors of* $\mathbf{A}^*\mathbf{A}$*.*

Proof. See, for example, [75, Ch. 2.6, pp. 19], [115, Ch. 6.2, pp. 206], [123, Ch. 2.4, pp. 76], [126, Ch. 5.12, pp. 411], etc. □

From Proposition A.5 and Theorem A.6 it is immediate that:

Corollary A.7. *Given* $\mathbf{A} \in \mathbb{C}^{m\times n}$ *with* $0 < r = \operatorname{rank}\{\mathbf{A}\} \leqslant \min\{m,n\}$ *and its SVD,* $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^*$*, then*

$$\mathbf{A}^+ = \mathbf{V}\Sigma^+\mathbf{U}^*.$$

Where,

$$\Sigma^+ = \begin{bmatrix} \operatorname{diag}\{1/\sigma_i\}_{i=1}^r & \mathbf{0}_{r\times(m-r)} \\ \mathbf{0}_{(n-r)\times r} & \mathbf{0}_{(n-r)\times(m-r)} \end{bmatrix} \in \mathbb{R}^{n\times m}.$$

This becomes particularly important to numerically compute the pseudoinverse of a matrix in an efficient way and to calculate the products of the third and fourth Penrose equations:

$$\mathbf{A}\mathbf{A}^+ = \mathbf{U}\Sigma\mathbf{V}^*\mathbf{V}\Sigma^+\mathbf{U}^* = \mathbf{U}\Sigma\Sigma^+\mathbf{U}^* = \mathbf{U}\begin{bmatrix} \mathbf{I}_{r\times r} & \mathbf{0}_{r\times(m-r)} \\ \mathbf{0}_{(m-r)\times r} & \mathbf{0}_{(m-r)\times(m-r)} \end{bmatrix}\mathbf{U}^*,$$
$$\mathbf{A}^+\mathbf{A} = \mathbf{V}\Sigma^+\mathbf{U}^*\mathbf{U}\Sigma\mathbf{V}^* = \mathbf{V}\Sigma\Sigma^+\mathbf{V}^* = \mathbf{V}\begin{bmatrix} \mathbf{I}_{r\times r} & \mathbf{0}_{r\times(n-r)} \\ \mathbf{0}_{(n-r)\times r} & \mathbf{0}_{(n-r)\times(n-r)} \end{bmatrix}\mathbf{V}^*.$$

A.2. Norms

Norms are an important topic in this thesis and therefore a (very) short summary of their main characteristics is presented here. The statements presented here are of common knowledge and can be found in many linear algebra and control books. In particular we quote [64, 69, 75] for being classical control specialized books.

Definition A.2.1. A *norm* $\nu(\cdot)$ is any operator over $\mathbf{x} \in \mathbb{C}^n$ such that:

- $\forall \mathbf{x} \in \mathbb{R}^n$, $\nu(\mathbf{x}) \geqslant 0$, with $\nu(\mathbf{x}) = 0 \iff \mathbf{x} = \mathbf{0}$.
- $\forall \mathbf{x} \wedge \mathbf{y} \in \mathbb{C}^n$, $\nu(\mathbf{x}+\mathbf{y}) \leqslant \nu(\mathbf{x}) + \nu(\mathbf{y})$.
- $\forall \mathbf{x} \in \mathbb{C}^n \wedge \alpha \in \mathbb{R}$, $\nu(\alpha\mathbf{x}) = |\alpha|\nu(\mathbf{x})$.

We reserve the notation $\|\mathbf{x}\|$ for the Euclidean norm of $\mathbf{x} \in \mathbb{R}^n$:

$$\|\mathbf{x}\| = \sqrt{\mathbf{x}'\mathbf{x}}.$$

Note that this norm is dependent on time if vector $\mathbf{x}$ is so. A generalization of this norm to make it time independent is the signal norm L_2 that is defined as:

$$\|\mathbf{x}\|_2 = \left(\int_{-\infty}^{+\infty} \mathbf{x}'\mathbf{x}dt\right)^{1/2}.$$

Because of Parseval's equation,

$$\int_{-\infty}^{+\infty} \|f(t)\|^2 dt = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \|F(j\omega)\|^2 d\omega,$$

where $F(j\omega) = \mathscr{L}\{f(t)\}|_{s=j\omega}$, the Fourier transformation of the signal $f(t)$, it holds that

$$\|\mathbf{x}\|_2 = \left(\frac{1}{2\pi}\int_{-\infty}^{+\infty} \hat{\mathbf{x}}'(-j\omega)\hat{\mathbf{x}}(j\omega)d\omega\right)^{1/2}.$$

With $\hat{\mathbf{x}}(s) = \mathscr{L}\{\mathbf{x}(t)\}$, the Laplace transformation of $\mathbf{x}$. From a general engineering perspective, this norm can be interpreted as the energy of a signal $\mathbf{x} \in \mathbb{R}^n$. However, other norms can also be defined. For example, the L_p signal norm for any $p \in \mathbb{N}$ is defined as:

$$\|\mathbf{x}\|_p = \left(\int_{-\infty}^{+\infty} \|\mathbf{x}\|^p dt\right)^{1/p}.$$

As $p \to \infty$, the L_∞-norm is obtained as:

$$\|\mathbf{x}\|_\infty = \max_t \|\mathbf{x}\|.$$

Matrix norms inducted from a general vector norm $\nu(\cdot)$, are defined as:

$$\nu(\mathbf{A}) = \max_{\mathbf{x}\neq\mathbf{0}} \frac{\nu(\mathbf{Ax})}{\nu(\mathbf{x})}.$$

Therefore $\nu(\mathbf{Ax}) \leqslant \nu(\mathbf{A})\nu(\mathbf{x})$. In the case of the Euclidean norm, the matrix norm is equivalent to the largest singular value of the matrix. That is,

$$\|\mathbf{A}\| = \max\{\mathrm{svd}\{\mathbf{A}\}\} = \sqrt{\max\left\{\mathrm{eig}\left\{\mathbf{A}'\mathbf{A}\right\}\right\}}. \tag{A.7}$$

System norms over a transfer function matrix $G(s)$ are defined in a similar way. In particular, the H_∞-norm can be defined as:

$$\|G(s)\|_\infty = \sup_\omega\{\|G(j\omega)\|\} = \sup_\omega\{\max\{\mathrm{svd}\{G(j\omega)\}\}\}.$$

An important property of the H_∞-norm is that $\|G(s)\hat{\mathbf{u}}\|_2 \leqslant \|G(s)\|_\infty\|\hat{\mathbf{u}}\|_2$, with $\hat{\mathbf{u}}$ the Laplace transformation of $\mathbf{u}$. This leads to the following tight bound for the H_∞-norm:

$$\|G(s)\|_\infty \geqslant \frac{\|G(s)\hat{\mathbf{u}}\|_2}{\|\hat{\mathbf{u}}\|_2} = \frac{\|\mathbf{z}(t)\|_2}{\|\mathbf{u}(t)\|_2},$$

with $\hat{\mathbf{z}}(s) = G(s)\hat{\mathbf{u}}(s) = \mathscr{L}\{\mathbf{z}(t)\}$, the Laplace transformation of the signal $\mathbf{z}$. Therefore, the H_∞-norm can be interpreted as the maximal energy gain between an input signal $\mathbf{u}$ and an output signal $\mathbf{z}$. This norm is then a natural choice to characterize the sensitivity of a system to perturbations. Indeed, if the H_∞-norm of the transfer function matrix between a perturbation and the outputs of a system is small, then the energy transfered to the output by the perturbation is also small, making the system itself less sensitive to this signal.

The scalar function

$$\mu(\mathbf{A}) = \max\left\{\text{eig}\left\{\frac{\mathbf{A}+\mathbf{A}'}{2}\right\}\right\},$$

is the so called *logarithm 2-norm* (or simply *log norm*) of matrix $\mathbf{A}$. Note that this operator is *not* a norm as it can be negative. A specialized summary on the topic with connections to stability theory can be found in [130]. A more general reference on the exponential matrix where this function becomes important is [128]. This value is related to the stability of a linear dynamic system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$ as if $\mu(\mathbf{A}) < 0$, then, the Lyapunov function $v(\mathbf{x}) = \mathbf{x}'\mathbf{x} = \|\mathbf{x}\|^2$ proves stability of the system. However, it can be the case that a Hurwitz matrix has a positive log norm.

An important result, proven in [130], is that $\forall t \geqslant 0$,

$$\|e^{\mathbf{A}t}\| \leqslant e^{\mu(\mathbf{A})t} \tag{A.8}$$

Where $\|\cdot\|$ is the Euclidean norm generalized to matrices. Observe that, as for Hurwitz matrices the log norm can be positive, this bound might be very conservative in those cases. Also note that if $\alpha(\mathbf{A}) = \max\{\text{real}\{\text{eig}\{\mathbf{A}\}\}\}$, $\alpha(\mathbf{A}) \leqslant \mu(\mathbf{A}) \implies e^{\alpha(\mathbf{A})t} \leqslant e^{\mu(\mathbf{A})t}$ but there is no guarantee that $\|e^{\mathbf{A}t}\| \leqslant e^{\alpha(\mathbf{A})t}$. From the definition of the Euclidean norm (A.7) and from (A.8), if $\mathbf{A} \in \mathbb{R}^{n\times n}$, the following matrix inequality follows,

$$e^{\mathbf{A}'t}e^{\mathbf{A}t} \leqslant \|e^{\mathbf{A}t}\|^2\mathbf{I} \leqslant e^{2\mu(\mathbf{A})t}\mathbf{I}. \tag{A.9}$$

A.3. General Algebraic Results

The results presented here are general algebraic properties of matrices that can be found in different general texts as, for example, [121, 123, 124, 126] and control classical texts as [64, 66, 75]. The results presented here are sometimes modified for the notation and objectives of this thesis.

Proposition A.8 (Congruence Transformation)**.** *For a given symmetric matrix $\mathbf{R} = \mathbf{R}' \in \mathbb{R}^{n\times n}$ and a linear full rank transformation $\mathbf{T} \in \mathbb{R}^{n\times n}$ it holds true that $\mathbf{R} > 0 \iff \mathbf{T}'\mathbf{R}\mathbf{T} > 0$.*

Proof. $\mathbf{R} > 0 \iff \forall \mathbf{x} \neq \mathbf{0}$, $\mathbf{x}'\mathbf{R}\mathbf{x} > 0$. Defining $\mathbf{y} = \mathbf{T}^{-1}\mathbf{x}$, then $\mathbf{x}'\mathbf{R}\mathbf{x} = \mathbf{y}'\mathbf{T}'\mathbf{R}\mathbf{T}\mathbf{y} > 0$. As $\mathbf{y} \neq \mathbf{0} \iff \mathbf{x} \neq \mathbf{0}$, $\mathbf{T}'\mathbf{R}\mathbf{T} > 0$. □

Proposition A.9 (Augmented Eigenvalues)**.** *For matrices* $\mathbf{X} \in \mathbb{R}^{q\times n}$ *and* $\mathbf{Y} \in \mathbb{R}^{n\times q}$*, with* $q \geqslant n$*, the non-zero eigenvalues of* $\mathbf{YX}$ *are the same as the non-zero eigenvalues of* $\mathbf{XY}$*. That is:*

$$\text{eig}\{\mathbf{YX}\} = \text{eig}\{\mathbf{XY}\} \cup Z,$$

where Z is a set of zeros ($Z = \{0, 0...0\}$*) with cardinality* $|Z| = q - n$*.*

Proof. This proof is based on the Sylvester's Determinant Theorem. For any scalar λ, define the following matrix:

$$\mathbf{M} = \begin{bmatrix} \lambda\mathbf{I}_q & \mathbf{X} \\ \lambda\mathbf{Y} & \lambda\mathbf{I}_n \end{bmatrix}.$$

The LU composition of this matrix is:

$$\mathbf{M} = \begin{bmatrix} \mathbf{I}_q & \mathbf{0} \\ \mathbf{Y} & \mathbf{I}_n \end{bmatrix} \begin{bmatrix} \lambda\mathbf{I}_q & \mathbf{X} \\ \mathbf{0} & \lambda\mathbf{I}_n - \mathbf{YX} \end{bmatrix}.$$

Therefore, the determinant of $\mathbf{M}$ is $|\mathbf{M}| = \lambda|\lambda\mathbf{I}_n - \mathbf{YX}|$. Similar, the UL composition of $\mathbf{M}$ is

$$\mathbf{M} = \begin{bmatrix} \lambda\mathbf{I}_q - \mathbf{XY} & \mathbf{X} \\ \mathbf{0} & \lambda\mathbf{I}_n \end{bmatrix} \begin{bmatrix} \mathbf{I}_q & \mathbf{0} \\ \mathbf{Y} & \mathbf{I}_n \end{bmatrix},$$

and the determinant $|\mathbf{M}| = |\lambda\mathbf{I}_q - \mathbf{XY}|\lambda$. So finally, for $\lambda \neq 0$,

$$|\lambda\mathbf{I}_n - \mathbf{YX}| = |\lambda\mathbf{I}_q - \mathbf{XY}|.$$

It is then clear that whenever $\lambda \neq 0$, the characteristic polynomial of matrix $\mathbf{XY}$ has the same roots as the characteristic polynomial of matrix $\mathbf{YX}$. As there are $m \leqslant n$ non-zero solutions for the characteristic equation of $\mathbf{YX}$, matrix $\mathbf{XY}$ should have $q - m$ zero eigenvalues. This situation can be written as the union of sets stated in the lemma. □

Proposition A.10 (Matrix Inversion)**.** *For non singular matrices* $\mathbf{A}$*,* $\mathbf{C}$ *and* $\mathbf{C}^{-1} + \mathbf{D}\mathbf{A}^{-1}\mathbf{B}$*, then*

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}\left(\mathbf{C}^{-1} + \mathbf{D}\mathbf{A}^{-1}\mathbf{B}\right)^{-1}\mathbf{D}\mathbf{A}^{-1}.$$

Proof. Multiplication of $(\mathbf{A} + \mathbf{BCD})$ by the stated inverse leads directly to the identity matrix:

$$\begin{aligned}
(\mathbf{A} + \mathbf{BCD})(\mathbf{A} + \mathbf{BCD})^{-1} &= (\mathbf{A} + \mathbf{BCD})(\mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}\left(\mathbf{C}^{-1} + \mathbf{D}\mathbf{A}^{-1}\mathbf{B}\right)^{-1}\mathbf{D}\mathbf{A}^{-1}) \\
&= \left(\mathbf{I} + \mathbf{BCDA}^{-1}\right) - (\mathbf{B}(\mathbf{C}^{-1} + \mathbf{D}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{D}\mathbf{A}^{-1} - \\
&\quad ... - \mathbf{BCDA}^{-1}\mathbf{B}(\mathbf{C}^{-1} + \mathbf{D}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{D}\mathbf{A}^{-1}) \\
&= \left(\mathbf{I} + \mathbf{BCDA}^{-1}\right) - \\
&\quad ... - \mathbf{BC}(\mathbf{C}^{-1} + \mathbf{D}\mathbf{A}^{-1}\mathbf{B})(\mathbf{C}^{-1} + \mathbf{D}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{D}\mathbf{A}^{-1} \\
&= \mathbf{I} + \mathbf{BCDA}^{-1} - \mathbf{BCDA}^{-1} = \mathbf{I}.
\end{aligned}$$

□

Proposition A.11 (Linear System Solution)**.** *The solution of the excited linear differential equation,*

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t),$$

with $\mathbf{x}(t) \in \mathbb{R}^n$*,* $\mathbf{u}(t) \in \mathbb{R}^p$*,* $\mathbf{A} \in \mathbb{R}^{n\times n}$*,* $\mathbf{B} \in \mathbb{R}^{n\times p}$*, and initial condition* $\mathbf{x}(t_0) = \mathbf{x}_0$ *is*

$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{x}_0 + \int_{t_0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau.$$

Proof. First note that,

$$\begin{aligned}
\mathbf{x}(t) &= e^{\mathbf{A}(t-t_0)}\mathbf{x}_0 + \int_{t_0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau \\
&= e^{\mathbf{A}(t-t_0)}\mathbf{x}_0 + e^{\mathbf{A}t}\int_{t_0}^{t} e^{-\mathbf{A}\tau}\mathbf{B}\mathbf{u}(\tau)d\tau.
\end{aligned}$$

Then it follows that,

$$\begin{aligned}
\dot{\mathbf{x}}(t) &= \frac{d}{dt}\left(e^{\mathbf{A}(t-t_0)}\mathbf{x}_0\right) + \frac{d}{dt}\left(e^{\mathbf{A}t}\int_{t_0}^{t} e^{-\mathbf{A}\tau}\mathbf{B}\mathbf{u}(\tau)d\tau\right) \\
&= \mathbf{A}e^{\mathbf{A}(t-t_0)}\mathbf{x}_0 + \left(\mathbf{A}e^{\mathbf{A}t}\int_{t_0}^{t} e^{-\mathbf{A}\tau}\mathbf{B}\mathbf{u}(\tau)d\tau + e^{\mathbf{A}t}e^{-\mathbf{A}t}\mathbf{B}\mathbf{u}(t)\right) \\
&= \mathbf{A}\left(e^{\mathbf{A}(t-t_0)}\mathbf{x}_0 + e^{\mathbf{A}t}\int_{t_0}^{t} e^{-\mathbf{A}\tau}\mathbf{B}\mathbf{u}(\tau)d\tau\right) + \mathbf{B}\mathbf{u}(t) \\
&= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t).
\end{aligned}$$

□

A.4. General LMI Results

The following results can be found in many LMI publications as they are useful to solve particular problems.

Proposition A.12 (Schur's Complement [80])**.** *For constant matrices of proper dimensions,* $\mathbf{R} < 0$ *and* $\mathbf{P} - \mathbf{S}'\mathbf{R}^{-1}\mathbf{S} < 0$ *is true if and only if*

$$\begin{bmatrix} \mathbf{P} & \mathbf{S}' \\ \mathbf{S} & \mathbf{R} \end{bmatrix} < 0$$

Proof. Consider the decomposition:

$$\begin{bmatrix} \mathbf{P} & \mathbf{S}' \\ \mathbf{S} & \mathbf{R} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{S}'\mathbf{R}^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{P}-\mathbf{S}'\mathbf{R}^{-1}\mathbf{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{S}'\mathbf{R}^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}'$$

Which, because of the Congruence Transformation property and the diagonal structure of the in-between matrix, is negative definite if and only if $\mathbf{P}-\mathbf{S}'\mathbf{R}^{-1}\mathbf{S} < 0$ and $\mathbf{R} < 0$. □

Lemma A.13 (Bounded Real Lemma (BRL) [80])**.** *For a real linear system described by*

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{w}$$
$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{w}.$$

Matrix $\mathbf{A}$ *is Hurwitz (i.e. all its eigenvalues have negative real parts) and the* H_∞*-norm of the transfer function* $H_{wy}(s)$*, between the signal* $\mathbf{w}$ *and the output* $\mathbf{y}$ *of the system, is smaller than* $\gamma > 0$*, if and only if there exists* $\mathbf{P} = \mathbf{P}' > 0$ *of proper dimensions such that the following inequality holds*

$$\begin{bmatrix} \mathbf{PA}+\mathbf{A}'\mathbf{P}+\mathbf{C}'\mathbf{C} & \mathbf{PB}+\mathbf{C}'\mathbf{D} \\ \star & \mathbf{D}'\mathbf{D}-\gamma^2\mathbf{I} \end{bmatrix} < 0. \tag{A.10}$$

A complete proof of the BRL is lengthy and difficult to summarize without omitting important information. Most of the seminal works on LMI and H_∞-norm, quote the result without proof as a basic milestone. See for example [80, 86, 87, 101, 108, 109]. A complete proof of the lemma can be found in [75, Ch. 12.1, pp. 238] (Corollary 12.3). We state here only some key ideas to properly understand the lemma.

The H_∞-norm of the system can be characterized as:

$$\|H_{wy}(s)\|_\infty < \gamma \iff \sup \frac{\sqrt{\int_0^T \mathbf{y}'\mathbf{y}dt}}{\sqrt{\int_0^T \mathbf{w}'\mathbf{w}dt}} < \gamma$$
$$\iff \sup \left\{ l(\mathbf{x},\mathbf{w}) := \int_0^T \mathbf{y}'\mathbf{y}dt - \gamma^2 \int_0^T \mathbf{w}'\mathbf{w}dt \right\} < 0$$

That is, the required bound on the H_∞-norm can be verified by solving a maximization problem. From standard LQR theory (see for example [63, 64]), it follows that a suboptimal solution for the maximization problem can be found if and only if the Hamiltonian function of the system, $h(\mathbf{x},\mathbf{w}) := \dot{v} + l(\mathbf{x},\mathbf{w})$, is negative, where $v(\mathbf{x}) > 0$ is a Lyapunov function. When $v(\mathbf{x}) = \mathbf{x}'\mathbf{P}\mathbf{x} \implies \dot{v} = \mathbf{x}'\mathbf{P}\dot{\mathbf{x}} + \dot{\mathbf{x}}'\mathbf{P}\mathbf{x}$, with $\mathbf{P} = \mathbf{P}' > 0$, then it follows that the Hamiltonian is negative when

$$\dot{v} + \mathbf{y}'\mathbf{y} - \gamma^2\mathbf{w}'\mathbf{w} < 0 \iff \begin{bmatrix} \mathbf{x}' \\ \mathbf{w}' \end{bmatrix} \begin{bmatrix} \mathbf{PA}+\mathbf{A}'\mathbf{P}+\mathbf{C}'\mathbf{C} & \mathbf{PB}+\mathbf{C}'\mathbf{D} \\ \star & \mathbf{D}'\mathbf{D}-\gamma^2\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x} & \mathbf{w} \end{bmatrix} < 0 \iff \text{(A.10)}.$$

Note that there is no further restriction regarding the use of the BRL but the existence of a real system described by $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$. In particular, signal $\mathbf{w}$ can be arbitrarily chosen, without restrictions on its origin or nature as long as $\|\mathbf{w}\|_2 < +\infty$.

If the inequality in the lemma holds, the system matrix $\mathbf{A}$ is Hurwitz because of Sylvester's criterion on positive (negative) definite Hermitian matrices: (A.10) $\implies \mathbf{PA} + \mathbf{A}'\mathbf{P} + \mathbf{C}'\mathbf{C} < 0 \iff \mathbf{PA} + \mathbf{A}'\mathbf{P} < -\mathbf{C}'\mathbf{C} \leqslant 0 \implies \mathbf{PA} + \mathbf{A}'\mathbf{P} < 0 \iff \mathbf{A}$ is Hurwitz.

An immediate corollary of the BRL is that the H_∞-norm of the system can be calculated through a convex minimization problem:

$$\|H_{wy}\|_\infty^2 = \inf\left\{\mu = \gamma^2 \in \mathbb{R}^+ | \text{LMI (A.10)} \wedge \mathbf{P} = \mathbf{P}' > 0\right\}.$$

Several equivalent representations of the Lemma can be found in the references. For example, by Schur's complement:

$$\begin{aligned}
\text{(A.10)} &\iff \begin{bmatrix} \mathbf{PA} + \mathbf{A}'\mathbf{P} & \mathbf{PB} \\ \star & -\gamma^2\mathbf{I} \end{bmatrix} + \begin{bmatrix} \mathbf{C}' \\ \mathbf{D}' \end{bmatrix} \mathbf{I} \begin{bmatrix} \mathbf{C} & \mathbf{D} \end{bmatrix} < 0 \\
&\iff \begin{bmatrix} \mathbf{PA} + \mathbf{A}'\mathbf{P} & \mathbf{PB} & \mathbf{C}' \\ \star & -\gamma^2\mathbf{I} & \mathbf{D}' \\ \star & \star & -\mathbf{I} \end{bmatrix} < 0
\end{aligned}$$

Applying a symmetric full rank congruence transformation and defining $\tilde{\mathbf{P}} = \frac{1}{\gamma}\mathbf{P} > 0$,

$$\begin{aligned}
\text{(A.10)} &\iff \begin{bmatrix} \frac{1}{\sqrt{\gamma}}\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\sqrt{\gamma}}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sqrt{\gamma}\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{PA} + \mathbf{A}'\mathbf{P} & \mathbf{PB} & \mathbf{C}' \\ \star & -\gamma^2\mathbf{I} & \mathbf{D}' \\ \star & \star & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{\gamma}}\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\sqrt{\gamma}}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \sqrt{\gamma}\mathbf{I} \end{bmatrix} < 0 \\
&\iff \begin{bmatrix} \tilde{\mathbf{P}}\mathbf{A} + \mathbf{A}'\tilde{\mathbf{P}} & \tilde{\mathbf{P}}\mathbf{B} & \mathbf{C}' \\ \star & -\gamma\mathbf{I} & \mathbf{D}' \\ \star & \star & -\gamma\mathbf{I} \end{bmatrix} < 0
\end{aligned}$$

A dual representation of the lemma can be obtained by defining $\mathbf{Q} = \mathbf{P}^{-1} > 0$:

$$\begin{aligned}
\text{(A.10)} &\iff \begin{bmatrix} \mathbf{Q} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{PA} + \mathbf{A}'\mathbf{P} & \mathbf{PB} & \mathbf{C}' \\ \star & -\gamma^2\mathbf{I} & \mathbf{D}' \\ \star & \star & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{Q} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} < 0 \\
&\iff \begin{bmatrix} \mathbf{AQ} + \mathbf{QA}' & \mathbf{B} & \mathbf{QC}' \\ \star & -\gamma^2\mathbf{I} & \mathbf{D}' \\ \star & \star & -\mathbf{I} \end{bmatrix} < 0
\end{aligned}$$

This last expression becomes important to design stabilizing state feedback controllers $\mathbf{u} = \mathbf{K}\mathbf{x}$ that guarantee a bound for the H_∞-norm of the transfer function between perturbation $\mathbf{w}$ and the output $\mathbf{y}$. Indeed, consider that $\mathbf{A} = \bar{\mathbf{A}} + \bar{\mathbf{B}}\mathbf{K}$, where $\mathbf{K}$ is unknown, then $\mathbf{A}\mathbf{Q} = \bar{\mathbf{A}}\mathbf{Q} + \bar{\mathbf{B}}\mathbf{K}\mathbf{Q} = \bar{\mathbf{A}}\mathbf{Q} + \bar{\mathbf{B}}\mathbf{Z}$ where $\mathbf{Z} = \mathbf{K}\mathbf{Q} \iff \mathbf{K} = \mathbf{Z}\mathbf{Q}^{-1}$ is a new variable. In that case, the inequality becomes linear with respect to its variables and its feasibility can be verified numerically using any available software.

The Finsler Lemma is another important result that is usually used to introduce additional variables that help relaxing the numeric verification of a Lyapunov stability condition. The Lemma was introduced by Paul Finsler in 1937, [120], it can also be found in a survey of 1979, [131], and is related to the so called *Projection Lemma* in [76]. The formulation stated here can be found in [98]. Other applications of the Lemma to LMIs are, *e.g.*, [86, 87, 102, 104, 112].

Lemma A.14 (Finsler [98]). *Consider $\mathbf{w} \in \mathbb{R}^n$, $\mathbf{\Xi} \in \mathbb{R}^{n\times n}$ and $\mathbf{\Upsilon} \in \mathbb{R}^{m\times n}$ with rank$\{\mathbf{\Upsilon}\} < n$ and $\mathbf{\Upsilon}^\perp$ a basis for the null space of $\mathbf{\Upsilon}$ (that is, $\mathbf{\Upsilon}\mathbf{\Upsilon}^\perp = \mathbf{0}$). Then, the following statements are equivalent.*

① $\mathbf{w}'\mathbf{\Xi}\mathbf{w} < 0$, $\forall \mathbf{w} \neq \mathbf{0}$: $\mathbf{\Upsilon}\mathbf{w} = \mathbf{0}$.

② $\left(\mathbf{\Upsilon}^\perp\right)'\mathbf{\Xi}\mathbf{\Upsilon}^\perp < 0$.

③ $\exists \mu \in \mathbb{R}$: $\mathbf{\Xi} - \mu\mathbf{\Upsilon}'\mathbf{\Upsilon} < 0$.

④ $\exists \mathbf{\Gamma} \in \mathbb{R}^{n\times m}$: $\mathbf{\Xi} + \mathbf{\Gamma}\mathbf{\Upsilon} + \mathbf{\Upsilon}'\mathbf{\Gamma}' < 0$.

Proof. We will prove that ① $\iff$ ② $\implies$ ③ $\implies$ ④ $\implies$ ② .

- ① $\iff$ ② : every $\mathbf{w}$ so that $\mathbf{\Upsilon}\mathbf{w} = \mathbf{0}$ can be written as $\mathbf{w} = \mathbf{\Upsilon}^\perp\mathbf{y}$. Therefore,
$$\begin{aligned} \mathbf{w}'\mathbf{\Xi}\mathbf{w} < 0, \forall \mathbf{w} \neq \mathbf{0} : \mathbf{\Upsilon}\mathbf{w} = \mathbf{0} \quad &\iff \mathbf{y}'\left(\mathbf{\Upsilon}^\perp\right)'\mathbf{\Xi}\mathbf{\Upsilon}^\perp\mathbf{y} < 0, \forall \mathbf{y} \neq \mathbf{0} \\ &\iff \left(\mathbf{\Upsilon}^\perp\right)'\mathbf{\Xi}\mathbf{\Upsilon}^\perp < 0 \end{aligned}$$

- ② $\implies$ ③ : Matrix $\mathbf{\Upsilon}$ can be written as $\mathbf{\Upsilon} = \mathbf{\Upsilon}_L\mathbf{\Upsilon}_R$ with $\mathbf{\Upsilon}_L$ and $\mathbf{\Upsilon}_R$ full rank. Then, defining $\mathbf{\Omega} = \mathbf{\Upsilon}_R'\left(\mathbf{\Upsilon}_R\mathbf{\Upsilon}_R'\right)^{-1}\left(\mathbf{\Upsilon}_L\mathbf{\Upsilon}_L'\right)^{-1/2}$, if ② is fulfilled then $(\mathbf{\Upsilon}^\perp)'\mathbf{\Xi}\mathbf{\Upsilon}^\perp < 0$ and there exists a $\mu \in \mathbb{R}$ such that:
$$\begin{bmatrix} \mathbf{\Omega}' \\ (\mathbf{\Upsilon}^\perp)' \end{bmatrix}\left(\mathbf{\Xi} - \mu\mathbf{\Upsilon}'\mathbf{\Upsilon}\right)\begin{bmatrix} \mathbf{\Omega} & \mathbf{\Upsilon}^\perp \end{bmatrix} = \begin{bmatrix} \mathbf{\Omega}'\mathbf{\Xi}\mathbf{\Omega} - \mu\mathbf{I} & \mathbf{\Omega}'\mathbf{\Xi}\mathbf{\Upsilon}^\perp \\ \star & (\mathbf{\Upsilon}^\perp)'\mathbf{\Xi}\mathbf{\Upsilon}^\perp \end{bmatrix} < 0$$
Consequently, $\mathbf{\Xi} - \mu\mathbf{\Upsilon}'\mathbf{\Upsilon} < 0$.

- ③ $\implies$ ④ : if ③ is verified, then $\mathbf{\Gamma} = -\frac{\mu}{2}\mathbf{\Upsilon}'$ satisfy ④ .

- ④ $\implies$ ② : Multiplying ④ to the left by $\left(\mathbf{\Upsilon}^\perp\right)'$ and to the right by $\mathbf{\Upsilon}^\perp$ leads to $\left(\mathbf{\Upsilon}^\perp\right)'\mathbf{\Xi}\mathbf{\Upsilon}^\perp < 0$.

□

The Finsler Lemma can be applied to find equivalent Lyapunov stability conditions. Indeed, for discrete time systems the following may be defined:

$$\mathbf{w} = \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}(k+1) \end{bmatrix}, \boldsymbol{\Upsilon} = \begin{bmatrix} \mathbf{A} & -\mathbf{I} \end{bmatrix}, \boldsymbol{\Upsilon}^{\perp} = \begin{bmatrix} \mathbf{I} \\ \mathbf{A} \end{bmatrix}, \boldsymbol{\Xi} = \begin{bmatrix} -\mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{P} \end{bmatrix}$$

What leads to the following conditions:

① : $\exists \mathbf{P} = \mathbf{P}' > 0$ so that

$$\begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}(k+1) \end{bmatrix}' \begin{bmatrix} -\mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{P} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}(k+1) \end{bmatrix} < 0, \forall \mathbf{x} \neq \mathbf{0} : \begin{bmatrix} \mathbf{A} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}(k+1) \end{bmatrix} = \mathbf{0}$$

② : $\exists \mathbf{P} = \mathbf{P}' > 0$ so that

$$\begin{bmatrix} \mathbf{I} \\ \mathbf{A} \end{bmatrix}' \begin{bmatrix} -\mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{P} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{A} \end{bmatrix} = \mathbf{A}'\mathbf{P}\mathbf{A} - \mathbf{P} < 0$$

③ : $\exists \mathbf{P} = \mathbf{P}' > 0$ and $\mu \in \mathbb{R}$ so that

$$\begin{bmatrix} -\mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{P} \end{bmatrix} - \mu \begin{bmatrix} \mathbf{A}' \\ -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A} & -\mathbf{I} \end{bmatrix} < 0$$

④ : $\exists \mathbf{P} = \mathbf{P}' > 0$ and $\boldsymbol{\Gamma} \in \mathbb{R}^{2n \times n}$ so that

$$\begin{bmatrix} -\mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{P} \end{bmatrix} + \boldsymbol{\Gamma} \begin{bmatrix} \mathbf{A} & -\mathbf{I} \end{bmatrix} + \begin{bmatrix} \mathbf{A}' \\ -\mathbf{I} \end{bmatrix} \boldsymbol{\Gamma}' < 0$$

Proposition A.15 (Crossed Products)**.** *For given vectors* $\mathbf{v} \in \mathbb{R}^n$*,* $\mathbf{w} \in \mathbb{R}^m$*, matrices* $\mathbf{M}$ *and* $\mathbf{N}$ *of appropriate dimensions. and for any* $\alpha > 0$*, the following statement is true*

$$\mathbf{v}'\mathbf{N}'\mathbf{M}\mathbf{w} + \mathbf{w}'\mathbf{M}'\mathbf{N}\mathbf{v} \leqslant \frac{1}{\alpha}\mathbf{v}'\mathbf{N}'\mathbf{N}\mathbf{v} + \alpha \mathbf{w}'\mathbf{M}'\mathbf{M}\mathbf{w}.$$

When $\mathbf{v} = \mathbf{w}$*, is immediate that*

$$\mathbf{N}'\mathbf{M} + \mathbf{M}'\mathbf{N} \leqslant \frac{1}{\alpha}\mathbf{N}'\mathbf{N} + \alpha \mathbf{M}'\mathbf{M}.$$

Proof. Because of the definition of interior product between vectors, the following is always true.

$$\left(\frac{1}{\sqrt{\alpha}}\mathbf{N}\mathbf{v} - \sqrt{\alpha}\mathbf{M}\mathbf{w}\right)' \left(\frac{1}{\sqrt{\alpha}}\mathbf{N}\mathbf{v} - \sqrt{\alpha}\mathbf{M}\mathbf{w}\right) \geqslant 0.$$

By factorization we obtain the expression of the Proposition. □

The following lemma and its proof can be found in [92].

Lemma A.16 (Gu's Integral Inequality [92])**.** *For a scalar $\tau > 0$, a vector function $\mathbf{v} : [0, \tau] \to \mathbb{R}^m$ and any positive definite symmetric matrix $\mathbf{M} \in \mathbb{R}^{m\times m}$ such that the integrations in the following are well defined, then*

$$\tau \int_0^\tau \mathbf{v}'(\alpha)\mathbf{M}\mathbf{v}(\alpha)d\alpha \geqslant \left(\int_0^\tau \mathbf{v}(\alpha)d\alpha\right)' \mathbf{M} \left(\int_0^\tau \mathbf{v}(\alpha)d\alpha\right). \tag{A.11}$$

Proof. Using Schur's complement, the following statement can be verified as always true for any $0 \leqslant \alpha \leqslant \tau$:

$$\begin{bmatrix} \mathbf{v}'(\alpha)\mathbf{M}\mathbf{v}(\alpha) & \mathbf{v}'(\alpha) \\ \star & \mathbf{M}^{-1} \end{bmatrix} \geqslant 0$$

Integration over $0 \leqslant \alpha \leqslant \tau$ leads to

$$\begin{bmatrix} \int_0^\tau \mathbf{v}'(\alpha)\mathbf{M}\mathbf{v}(\alpha)d\alpha & \int_0^\tau \mathbf{v}'(\alpha)d\alpha \\ \star & \tau\mathbf{M}^{-1} \end{bmatrix} \geqslant 0.$$

By applying Schur's complement one gets the expression of the lemma. For the scalar case ($m = 1$) this result is well known in calculus. □

The following result can be found in [83] and deals with the problem of output feedback design.

Theorem A.17 ([83])**.** *If $\mathbf{C} \in \mathbb{R}^{q\times n}$ is full row rank and there exist matrices $\mathbf{Q} > 0$, $\mathbf{N} \in \mathbb{R}^{p\times q}$, and $\mathbf{M} \in \mathbb{R}^{q\times q}$ such that*

$$\mathbf{AQ} + \mathbf{QA}' + \mathbf{BNC} + \mathbf{C}'\mathbf{N}'\mathbf{B}' < 0, \tag{A.12}$$

and

$$\mathbf{MC} = \mathbf{CQ}, \tag{A.13}$$

then the feedback matrix $\mathbf{L} = \mathbf{NM}^{-1} \in \mathbb{R}^{p\times q}$ makes $\mathbf{A} + \mathbf{BLC} \in \mathbb{R}^{n\times n}$ to be Hurwitz.

Proof. From (A.13), if $\mathbf{C}$ is full row rank, it follows that $\mathbf{M}$ is also full rank and therefore is invertible. From the Lyapunov condition, $\mathbf{A}+\mathbf{BLC}$ is Hurwitz if and only if it exists $\mathbf{Q}>0$ such that

$$\mathbf{AQ}+\mathbf{QA}'+\mathbf{BLCQ}+\mathbf{QC}'\mathbf{L}'\mathbf{B}'<0. \tag{A.14}$$

If (A.12) and (A.13) hold, then $\mathbf{L}=\mathbf{NM}^{-1} \iff \mathbf{N}=\mathbf{LM} \implies \mathbf{NC}=\mathbf{LMC} \iff \mathbf{NC}=\mathbf{LCQ} \implies$ (A.14). □

Several extensions of this result can be considered including, for example, conditions for output feedback design subject to H_∞-norm restrictions.

A.5. Other Results

The *Assignment Problem* is popular in linear optimization and usually described as how to assign a number of resources to an equal number of tasks in such a way that an overall linear cost is minimized.

Problem A.1 (Assignment Problem). Given sets W and T both with cardinality n; given a cost function $c: W \times T \to \mathbb{R}_0^+$; and n^2 binary variables x_{kl} such that $x_{kl}=1$ if $k \in W$ is assigned to $l \in T$ and $x_{kl}=0$ otherwise;

$$\begin{aligned}
&\min_{x_{kl}\geqslant 0} \quad J=\sum_{k\in W}\sum_{l\in T} c(k,l)x_{kl}\\
&\quad \text{s.t.}\\
&\qquad \sum_{l\in T} x_{kl}=1,\ \forall k\in W\\
&\qquad \sum_{k\in W} x_{kl}=1,\ \forall l\in T
\end{aligned}$$

This problem can be found in most optimization text books, but we quote the seminal papers [125, 129] because of the proposed method known as *The Hungarian Method* or *Munkres Algorithm*, which is the most popular algorithmic solution. This algorithm finds an optimal solution in polynomial time and can be efficiently implemented. The obvious brute-force algorithm of testing all possible $n!$ permutations becomes prohibitive for large n. Besides this, other optimal and sub-optimal algorithms can be proposed to solve the problem. Implementation details of the Hungarian method can be found on line at `http://csclab.murraystate.edu/bob.pilgrim/445/munkres.html`. A Matlab® implementation of the algorithm can be found at `http://www.mathworks.com/matlabcentral/fileexchange/20328-munkres-assignment-algorithm`.

B
Electrical Grid Modeling

This Appendix explains the model of an electrical grid and the main components used in this Thesis. Most of the statements here are standard knowledge in electrical engineering and can be found in many text books on the subject as [173, 175].

B.1. Electric quantities

Balanced electrical three phase signals (voltage or current) are represented as three elements vector in the so call *abc* representation:

$$\mathbf{x}^{abc} = \begin{bmatrix} x^a \\ x^b \\ x^c \end{bmatrix} = \begin{bmatrix} A\sin(\psi) \\ A\sin(\psi - 2\pi/3) \\ A\sin(\psi - 4\pi/3) \end{bmatrix}.$$

These signals are balanced in the sense that they add up to zero: $x^a + x^b + x^c = 0$. Unbalanced signals will not be discussed. The electrical angle $\psi : \mathbb{R}_0^+ \mapsto \mathbb{R}_0^+$ is a function of time whose derivative is the frequency of the electrical signal. We assume that this function can be described by a nominal constant angular frequency $\omega = 2\pi f$ and an angular phase $\delta : \mathbb{R}_0^+ \mapsto \mathbb{R}_0^+$ so that $\psi(t) = \omega t + \delta(t) = 2\pi f t + \delta(t)$, where f is measured in Hertz and usually has the value $50[Hz]$ or $60[Hz]$ depending on national and regional regulations. Note that the real frequency of the signal is then $\frac{d}{dt}\psi = \omega + \frac{d}{dt}\delta$.

Because of their trigonometric characteristics, electrical signals in the *abc* representation are difficult to be dealt with. Therefore the Park's transformation (or $dq0$ transformation) is proposed to map the *abc* signal to a rotatory reference frame of frequency ω where the quantities can be managed easily. This transformation is defined as:

$$\mathbf{x}^{dq0} = \mathbf{T}_{abc-dq0}\mathbf{x}^{abc},$$

with the time dependent matrix

$$\mathbf{T}_{abc-dq0} = \sqrt{\frac{2}{3}} \begin{bmatrix} \sin(\omega t) & \sin(\omega t - 2\pi/3) & \sin(\omega t - 4\pi/3) \\ \cos(\omega t) & \cos(\omega t - 2\pi/3) & \cos(\omega t - 4\pi/3) \\ 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}.$$

From here, the $dq0$ representation of an electric signal can be calculated as:

$$\mathbf{x}^{dq0} = \mathbf{T}_{abc-dq0} \begin{bmatrix} A\sin(\psi) \\ A\sin(\psi - 2\pi/3) \\ A\sin(\psi - 4\pi/3) \end{bmatrix} = A\sqrt{\frac{3}{2}} \begin{bmatrix} \cos(\delta) \\ \sin(\delta) \\ 0 \end{bmatrix} = \begin{bmatrix} x^q \\ x^d \\ 0 \end{bmatrix}.$$

Note that the third element of the vector is always zero for balanced signals and therefore only the dq components are relevant for analysis. This is coherent with the fact that electrical signals are usually described by their amplitude A and the phase angle δ defined around the nominal frequency ω. Therefore, the following simplification holds true when it is known that the signals are balanced:

$$\mathbf{x}^{dq} = \begin{bmatrix} x^q \\ x^d \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{x}^{dq0} \iff \mathbf{x}^{dq0} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}^{dq}.$$

An inverse transformation can also be defined:

$$\mathbf{x}^{abc} = \mathbf{T}_{dq0-abc}\mathbf{x}^{dq0},$$

with

$$\mathbf{T}_{dq0-abc} := \mathbf{T}^{-1}_{abc-dq0} = \mathbf{T}'_{abc-dq0}.$$

Some relevant properties of the $dq0$ transformation are:

$$\begin{aligned}
(\mathbf{x}^{abc})'(\mathbf{y}^{abc}) &= (\mathbf{x}^{dq0})'(\mathbf{y}^{dq0}), \\
\frac{d}{dt}\mathbf{T}_{abc-dq0} &= \mathbf{W}'_{dq0}\mathbf{T}_{abc-dq0}, \\
\frac{d}{dt}\mathbf{T}_{dq0-abc} &= \mathbf{T}_{dq0-abc}\mathbf{W}_{dq0}, \\
\frac{d}{dt}\mathbf{x}^{abc} &= \frac{d}{dt}\left(\mathbf{T}_{dq0-abc}\mathbf{x}^{dq0}\right) \\
&= \left(\frac{d}{dt}\mathbf{T}_{dq0-abc}\right)\mathbf{x}^{dq0} + \mathbf{T}_{dq0-abc}\frac{d}{dt}\mathbf{x}^{dq0} \\
&= \mathbf{T}_{dq0-abc}\mathbf{W}_{dq0}\mathbf{x}^{dq0} + \mathbf{T}_{dq0-abc}\frac{d}{dt}\mathbf{x}^{dq0},
\end{aligned}$$

where

$$\mathbf{W}_{dq0} = \begin{bmatrix} 0 & -\omega & 0 \\ \omega & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Consider current and voltage signals given in the *abc* representation respectively by:

$$\mathbf{i}^{abc} = \begin{bmatrix} I\sin(\omega t + \delta_i) \\ I\sin(\omega t - 2\pi/3 + \delta_i) \\ I\sin(\omega t - 4\pi/3 + \delta_i) \end{bmatrix}, \ \mathbf{v}^{abc} = \begin{bmatrix} V\sin(\omega t + \delta_v) \\ V\sin(\omega t - 2\pi/3 + \delta_v) \\ V\sin(\omega t - 4\pi/3 + \delta_v) \end{bmatrix}.$$

Or equivalently in the $dq0$ reference frame:

$$\mathbf{i}^{dq0} = \sqrt{3}I_{RMS}\begin{bmatrix} \cos(\delta_i) \\ \sin(\delta_i) \\ 0 \end{bmatrix}, \ \mathbf{v}^{dq0} = \sqrt{3}V_{RMS}\begin{bmatrix} \cos(\delta_v) \\ \sin(\delta_v) \\ 0 \end{bmatrix}.$$

With the amplitude and the RMS (*Root Mean Square*) values related by $I = \sqrt{2}I_{RMS}$ and $V = \sqrt{2}V_{RMS}$. The RMS value of a periodic quantity $x(t)$ is defined as

$$X_{RMS} := \sqrt{\frac{1}{T}\int_0^T x^2(t)dt},$$

where $T = 1/f$ is the period of the signal. Applying this definition to pure sinusoidal signals, the stated relationships between the amplitude and the RMS value are obtained.

The *active power* of the signals is defined as:

$$\begin{aligned} P = \mathbf{v}^{dq0}\cdot\mathbf{i}^{dq0} &= v^d i^d + v^q i^q \\ &= 3V_{RMS}I_{RMS}\left[\cos(\delta_i)\cos(\delta_v) + \sin(\delta_i)\sin(\delta_v)\right] \\ &= 3V_{RMS}I_{RMS}\cos(\delta_v - \delta_i). \end{aligned}$$

From a physical perspective, this quantity is the fraction of the electric power that can be successfully transformed into mechanical work. The *reactive power* in counterpart is the power that cannot be transformed into mechanical work:

$$\begin{aligned} Q = |\mathbf{v}^{dq0}\times\mathbf{i}^{dq0}| &= v^d i^q - v^q i^d \\ &= 3V_{RMS}I_{RMS}\left[\sin(\delta_v)\cos(\delta_i) - \cos(\delta_v)\sin(\delta_i)\right] \\ &= 3V_{RMS}I_{RMS}\sin(\delta_v - \delta_i). \end{aligned}$$

The *apparent power* is the vectorial addition of the previous quantities. That is:

$$\begin{aligned} |S| = \sqrt{P^2 + Q^2} &= \sqrt{(v^d i^d)^2 + (v^d i^q)^2 + (v^q i^d)^2 + (v^q i^q)^2} \\ &= 3V_{RMS}I_{RMS}. \end{aligned}$$

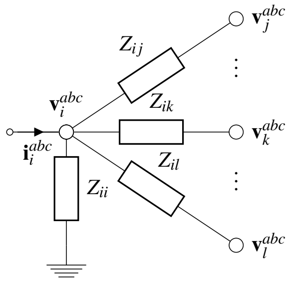


Figure B.1.: Voltage and current at node $i \in \mathcal{V}$ of a microgrid with $\mathcal{N}_i = \{j, ..., k, ..., l\}$.

To have a relative indication of the power fraction that can be converted into work, the *power factor* is defined as:

$$\begin{aligned} f_p &= P/|S| = \cos(\delta_v - \delta_i) \\ &= \cos(\arctan(v^q/v^d) - \arctan(i^q/i^d)) \\ &= \frac{v^d i^d + v^q i^q}{\sqrt{(v^d i^d)^2 + (v^d i^q)^2 + (v^q i^d)^2 + (v^q i^q)^2}}. \end{aligned}$$

B.2. Grid circuital relationships

A schematic circuital representation of a node connected to the rest of a grid with the description of Section 6.1 can be seen in Figure B.1. At each node $i \in \mathcal{V}$ a three phase balanced voltage $\mathbf{v}_i^{abc}$ is inducted by a generation unit injecting to the grid a current $\mathbf{i}_i^{abc}$.

From Kirchhoff's current law applied at node $i \in \mathcal{V}$ we obtain that

$$\mathbf{i}_i^{abc} = \mathbf{i}_{ii}^{abc} + \sum_{j \in \mathcal{N}_i} \mathbf{i}_{ij}^{abc} \iff \mathbf{i}_i^{dq} = \mathbf{i}_{ii}^{dq} + \sum_{j \in \mathcal{N}_i} \mathbf{i}_{ij}^{dq}. \tag{B.1}$$

From Kirchhoff's voltage law applied over the load at node $i \in \mathcal{V}$:

$$\mathbf{v}_i^{abc} = R_{ii}\mathbf{i}_{ii}^{abc} + L_{ii}\frac{d}{dt}\mathbf{i}_{ii}^{abc} \iff \mathbf{v}_i^{dq} = \left(R_{ii}\mathbf{I} + L_{ii}\mathbf{W}_{dq}\right)\mathbf{i}_{ii}^{dq} + L_{ii}\frac{d}{dt}\mathbf{i}_{ii}^{dq}. \tag{B.2}$$

With $\mathbf{W}_{dq}$ the 2×2 upper left submatrix of $\mathbf{W}_{dq0}$. Similarly, over each line between $i \in \mathcal{V}$ and $j \in \mathcal{N}_i$,

$$\mathbf{v}_i^{abc} = \mathbf{v}_j^{abc} + R_{ij}\mathbf{i}_{ij}^{abc} + L_{ij}\frac{d}{dt}\mathbf{i}_{ij}^{abc} \iff \mathbf{v}_i^{dq} = \mathbf{v}_j^{dq} + \left(R_{ij}\mathbf{I} + L_{ij}\mathbf{W}_{dq}\right)\mathbf{i}_{ij}^{dq} + L_{ij}\frac{d}{dt}\mathbf{i}_{ij}^{dq}. \tag{B.3}$$

Taking to Laplace domine and combining equations (B.1), (B.2) and (B.3):

$$\mathbf{i}_i^{dq}(s) = \left(R_{ii}\mathbf{I} + L_{ii}\mathbf{W}_{dq} + L_{ii}s\mathbf{I}\right)^{-1}\mathbf{v}_i^{dq}(s) + \sum_{j \in \mathcal{N}_i}\left(R_{ij}\mathbf{I} + L_{ij}\mathbf{W}_{dq} + L_{ij}s\mathbf{I}\right)^{-1}(\mathbf{v}_i^{dq}(s) - \mathbf{v}_j^{dq}(s)).$$

Note that the inverse matrices in the previous equation can be rewritten using

$$\left(R\mathbf{I}+L\mathbf{W}_{dq}+Ls\mathbf{I}\right)^{-1}=\begin{bmatrix}R+Ls & -\omega L\\ \omega L & R+Ls\end{bmatrix}^{-1}=\frac{1}{(R+Ls)^2+\omega^2L^2}\begin{bmatrix}R+Ls & \omega L\\ -\omega L & R+Ls\end{bmatrix},$$

and therefore,

$$\begin{aligned}\begin{bmatrix}i_i^q\\ i_i^d\end{bmatrix}&=\frac{1}{(R_{ii}+L_{ii}s)^2+\omega^2L_{ii}^2}\begin{bmatrix}R_{ii}+L_{ii}s & \omega L_{ii}\\ -\omega L_{ii} & R_{ii}+L_{ii}s\end{bmatrix}\begin{bmatrix}v_i^q\\ v_i^d\end{bmatrix}+\ldots\\ &\ldots+\sum_{j\in\mathscr{N}_i}\frac{1}{(R_{ij}+L_{ij}s)^2+\omega^2L_{ij}^2}\begin{bmatrix}R_{ij}+L_{ij}s & \omega L_{ij}\\ -\omega L_{ij} & R_{ij}+L_{ij}s\end{bmatrix}\begin{bmatrix}v_i^q-v_j^q\\ v_i^v-v_j^v\end{bmatrix}.\end{aligned}$$

Identifying the d and q components,

$$\begin{aligned}i_i^q(s)=&\left[\frac{R_{ii}+L_{ii}s}{(R_{ii}+L_{ii}s)^2+\omega^2L_{ii}^2}+\sum_{j\in\mathscr{N}_i}\frac{R_{ij}+L_{ij}s}{(R_{ij}+L_{ij}s)^2+\omega^2L_{ij}^2}\right]v_i^q(s)+\ldots\\ &\ldots\left[\frac{\omega L_{ii}}{(R_{ii}+L_{ii}s)^2+\omega^2L_{ii}^2}+\sum_{j\in\mathscr{N}_i}\frac{\omega L_{ij}}{(R_{ij}+L_{ij}s)^2+\omega^2L_{ij}^2}\right]v_i^d(s)+\ldots\\ &\ldots-\sum_{j\in\mathscr{N}_i}\frac{R_{ij}+L_{ij}s}{(R_{ij}+L_{ij}s)^2+\omega^2L_{ij}^2}v_j^q(s)-\sum_{j\in\mathscr{N}_i}\frac{\omega L_{ij}}{(R_{ij}+L_{ij}s)^2+\omega^2L_{ij}^2}v_j^d(s),\end{aligned}$$

$$\begin{aligned}i_i^d(s)=&\left[\frac{-\omega L_{ii}}{(R_{ii}+L_{ii}s)^2+\omega^2L_{ii}^2}+\sum_{j\in\mathscr{N}_i}\frac{-\omega L_{ij}}{(R_{ij}+L_{ij}s)^2+\omega^2L_{ij}^2}\right]v_i^q(s)+\ldots\\ &\ldots\left[\frac{R_{ii}+L_{ii}s}{(R_{ii}+L_{ii}s)^2+\omega^2L_{ii}^2}+\sum_{j\in\mathscr{N}_i}\frac{R_{ij}+L_{ij}s}{(R_{ij}+L_{ij}s)^2+\omega^2L_{ij}^2}\right]v_i^d(s)+\ldots\\ &\ldots-\sum_{j\in\mathscr{N}_i}\frac{-\omega L_{ij}}{(R_{ij}+L_{ij}s)^2+\omega^2L_{ij}^2}v_j^q(s)-\sum_{j\in\mathscr{N}_i}\frac{R_{ij}+L_{ij}s}{(R_{ij}+L_{ij}s)^2+\omega^2L_{ij}^2}v_j^d(s).\end{aligned}$$

A common assumption on electrical systems is that the dynamics of the phase angle are very fast with respect to the rest of the components of the grid. Therefore, it can be considered that currents are in stationary state, *i.e.* $s = 0$. Then,

$$
\begin{aligned}
i_i^q = & \left[\frac{R_{ii}}{(R_{ii})^2 + \omega^2 L_{ii}^2} + \sum_{j \in \mathcal{N}_i} \frac{R_{ij}}{(R_{ij})^2 + \omega^2 L_{ij}^2} \right] v_i^q + \ldots \\
& \ldots \left[\frac{\omega L_{ii}}{(R_{ii})^2 + \omega^2 L_{ii}^2} + \sum_{j \in \mathcal{N}_i} \frac{\omega L_{ij}}{(R_{ij})^2 + \omega^2 L_{ij}^2} \right] v_i^d + \ldots \\
& \ldots - \sum_{j \in \mathcal{N}_i} \frac{R_{ij}}{(R_{ij})^2 + \omega^2 L_{ij}^2} v_j^q - \sum_{j \in \mathcal{N}_i} \frac{\omega L_{ij}}{(R_{ij})^2 + \omega^2 L_{ij}^2} v_j^d,
\end{aligned}
$$

$$
\begin{aligned}
i_i^d = & \left[\frac{-\omega L_{ii}}{(R_{ii})^2 + \omega^2 L_{ii}^2} + \sum_{j \in \mathcal{N}_i} \frac{-\omega L_{ij}}{(R_{ij})^2 + \omega^2 L_{ij}^2} \right] v_i^q + \ldots \\
& \ldots \left[\frac{R_{ii}}{(R_{ii})^2 + \omega^2 L_{ii}^2} + \sum_{j \in \mathcal{N}_i} \frac{R_{ij}}{(R_{ij})^2 + \omega^2 L_{ij}^2} \right] v_i^d + \ldots \\
& \ldots - \sum_{j \in \mathcal{N}_i} \frac{-\omega L_{ij}}{(R_{ij})^2 + \omega^2 L_{ij}^2} v_j^q - \sum_{j \in \mathcal{N}_i} \frac{R_{ij}}{(R_{ij})^2 + \omega^2 L_{ij}^2} v_j^d.
\end{aligned}
$$

Using these expressions for the injected current as a function of the induced voltages and from the definition of active and reactive power one gets:

$$
\begin{aligned}
P_i = & \left[\frac{R_{ii}}{R_{ii}^2 + \omega^2 L_{ii}^2} + \sum_{j \in \mathcal{N}_i} \frac{R_{ij}}{R_{ij}^2 + \omega^2 L_{ij}^2} \right] (v_i^q)^2 + \ldots \\
& \ldots - \sum_{j \in \mathcal{N}_i} \frac{R_{ij}}{R_{ij}^2 + \omega^2 L_{ij}^2} v_j^q v_i^q - \sum_{j \in \mathcal{N}_i} \frac{\omega L_{ij}}{R_{ij}^2 + \omega^2 L_{ij}^2} v_j^d v_i^q + \ldots \\
& \ldots \left[\frac{R_{ii}}{R_{ii}^2 + \omega^2 L_{ii}^2} + \sum_{j \in \mathcal{N}_i} \frac{R_{ij}}{R_{ij}^2 + \omega^2 L_{ij}^2} \right] (v_i^d)^2 + \ldots \\
& \ldots - \sum_{j \in \mathcal{N}_i} \frac{-\omega L_{ij}}{R_{ij}^2 + \omega^2 L_{ij}^2} v_j^q v_i^d - \sum_{j \in \mathcal{N}_i} \frac{R_{ij}}{R_{ij}^2 + \omega^2 L_{ij}^2} v_j^d v_i^d,
\end{aligned}
$$

$$Q_i = \left[\frac{\omega L_{ii}}{R_{ii}^2 + \omega^2 L_{ii}^2} + \sum_{j \in \mathcal{N}_i} \frac{\omega L_{ij}}{R_{ij}^2 + \omega^2 L_{ij}^2} \right] (v_i^q)^2 + \ldots$$

$$\ldots + \sum_{j \in \mathcal{N}_i} \frac{-\omega L_{ij}}{R_{ij}^2 + \omega^2 L_{ij}^2} v_j^q v_i^q + \sum_{j \in \mathcal{N}_i} \frac{R_{ij}}{R_{ij}^2 + \omega^2 L_{ij}^2} v_j^d v_i^q + \ldots$$

$$\ldots + \left[\frac{\omega L_{ii}}{R_{ii}^2 + \omega^2 L_{ii}^2} + \sum_{j \in \mathcal{N}_i} \frac{\omega L_{ij}}{R_{ij}^2 + \omega^2 L_{ij}^2} \right] (v_i^d)^2 + \ldots$$

$$\ldots - \sum_{j \in \mathcal{N}_i} \frac{R_{ij}}{R_{ij}^2 + \omega^2 L_{ij}^2} v_j^q v_i^d - \sum_{j \in \mathcal{N}_i} \frac{\omega L_{ij}}{R_{ij}^2 + \omega^2 L_{ij}^2} v_j^d v_i^d .$$

At ever node $i \in \mathcal{V}$ the dq components of the voltages can be written as a function of their amplitudes and phase angles:

$$v_i^q(t) = \sqrt{\frac{3}{2}} V_i(t) \cos(\delta_i(t))$$
$$v_i^d(t) = \sqrt{\frac{3}{2}} V_i(t) \sin(\delta_i(t)) .$$

Replacing the voltage expressions and considering the trigonometric identities:

$\cos^2(\delta_i) + \sin^2(\delta_i) = 1$,
$\cos(\delta_i)\cos(\delta_j) + \sin(\delta_i)\sin(\delta_j) = \cos(\delta_i - \delta_j)$, and
$\sin(\delta_i)\cos(\delta_j) - \cos(\delta_i)\sin(\delta_j) = \sin(\delta_i - \delta_j)$,

the expressions for the stationary active and reactive power can be simplified to obtain:

$$\begin{aligned} P_i(t) = \frac{3}{2} &\left[\frac{R_{ii}}{R_{ii}^2 + \omega^2 L_{ii}^2} + \sum_{j \in \mathcal{N}_i} \frac{R_{ij}}{R_{ij}^2 + \omega^2 L_{ij}^2} \right] V_i^2(t) + \ldots \\ &- \frac{3}{2} \sum_{j \in \mathcal{N}_i} \frac{R_{ij}}{R_{ij}^2 + \omega^2 L_{ij}^2} V_i(t) V_j(t) \cos(\delta_i(t) - \delta_j(t)) + \ldots \\ &+ \frac{3}{2} \sum_{j \in \mathcal{N}_i} \frac{\omega L_{ij}}{R_{ij}^2 + \omega^2 L_{ij}^2} V_i(t) V_j(t) \sin(\delta_i(t) - \delta_j(t)) , \end{aligned} \tag{B.4}$$

$$\begin{aligned} Q_i(t) = \frac{3}{2} &\left[\frac{\omega L_{ii}}{R_{ii}^2 + \omega^2 L_{ii}^2} + \sum_{j \in \mathcal{N}_i} \frac{\omega L_{ij}}{R_{ij}^2 + \omega^2 L_{ij}^2} \right] V_i^2(t) + \ldots \\ &- \frac{3}{2} \sum_{j \in \mathcal{N}_i} \frac{\omega L_{ij}}{R_{ij}^2 + \omega^2 L_{ij}^2} V_i(t) V_j(t) \cos(\delta_i(t) - \delta_j(t)) + \ldots \\ &- \frac{3}{2} \sum_{j \in \mathcal{N}_i} \frac{R_{ij}}{R_{ij}^2 + \omega^2 L_{ij}^2} V_i(t) V_j(t) \sin(\delta_i(t) - \delta_j(t)) . \end{aligned} \tag{B.5}$$

Bibliography

Consensus References

[1] Abdelkader Abdessameud and Abdelhamid Tayebi. "On consensus algorithms design for double integrator dynamics." In: *Automatica* 49 (2013) (cit. on pp. 1, 151).

[2] Gustavo Ayres de Castro and Fernando Paganini. "Convex synthesis of controllers for consensus." In: *American Control Conference*. 2004 (cit. on pp. 1, 55).

[3] Tucker Balch and Ronald C. Arkin. "Behavior-based Formation Control for Multi-robot Teams." In: *IEEE Transactions on Robotics and Automation* 14 (1998) (cit. on pp. 2, 151).

[4] Zhou Bo, Wang Wei, and Ye Hao. "Distributed consensus tracking control of linear multi-agent systems with actuator faults." In: *IEEE Multi-Conference on Systems and Control (MSC)*. 2014 (cit. on pp. 2, 52).

[5] Yongcan Cao et al. "An Overview of Recent Progress in the Study of Distributed Multi-agent Coordination." In: *IEEE Transactions on Industrial Informatics* 9 (2012) (cit. on p. 1).

[6] S. Chatterjee and E. Senata. "Towards Consensus: Some Convergence Theorems on Repeated Averaging." In: *Journal of Applied Probability* 14 (1977) (cit. on p. 1).

[7] YangQuan Chen and Zhongmin Wang. "Formation Control: A Review and A New Consideration." In: *International Conference on Intelligent Robots and Systems (IROS 2005)*. 2005 (cit. on pp. 2, 151).

[8] Jimming Cheng, Winston Cheng, and Radhika Nagpal. "Robust and self-repairing formation control for swarms of mobile agents." In: *20th national conference on Artificial intelligence*. 2005 (cit. on pp. 2, 151).

[9] Nilanjan Roy Chowdhury and Srikant Sukumar. "Persistence based analysis of consensus protocols for dynamic graph networks." In: *European Control Conference*. 2014 (cit. on pp. 1, 98).

[10] Yao-Li Chuang et al. "Multi-Vehicle Flocking: Scalability of Cooperative Control Algorithms using Pairwise Potentials." In: *IEEE International Conference on Robotics and Automation*. 2007 (cit. on p. 2).

[11] Luca Consolini et al. "Leader-Follower Formation Control of Nonholonomic Mobile Robots with Input Constraints." In: *Automatica* 44 (2008) (cit. on pp. 2, 151).

[12] Celso De La Cruz and Ricardo Carelli. “Dynamic Modeling and Centralized Formation Control of Mobile Robots.” In: *32nd Annual Conference on IEEE Industrial Electronics*. 2006 (cit. on pp. 2, 151).

[13] Paresh Deshpande et al. “Formation Control of Multi-agent Systems with Double Integrator Dynamics using Delayed Static Output Feedback.” In: *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*. 2011 (cit. on pp. 2, 151).

[14] Darina Goldin. “Stability and Controllability of Double Integrator Consensus Systems in Heterogeneous Networks.” PhD thesis. Technische Universität Berlin, 2013 (cit. on pp. 1, 39, 51, 52).

[15] Darina Goldin, Sid Ahmed Attia, and Jörg Raisch. “Consensus for Double Integrator Dynamics in Heterogeneous Networks.” In: *49th IEEE Conference on Decision and Control (CDC)*. 2010 (cit. on p. 1).

[16] Darina Goldin and Jörg Raisch. “Controllability of Second Order Leader-Follower Systems.” In: *2nd IFAC Workshop on Estimation and Control of Networked Systems*. 2010 (cit. on pp. 1, 151).

[17] P. Grohs. “Wolfowitz’s Theorem and Consensus Algorithms in Hadamard Spaces.” In: *Proceedings of the American Mathematica Society* 143 (2015) (cit. on p. 1).

[18] Meng Ji and Magnus Egerstedt. “Distributed Coordination Control of Multiagent Systems While Preserving Connectedness.” In: *IEEE Transactions on Robotics* 23 (2007) (cit. on p. 1).

[19] Fangcui Jiang and Long Wang. “Finite-time information consensus for multi-agent systems with fixed and switching topologies.” In: *Physica D* 238 (2009) (cit. on pp. 1, 98).

[20] Zhongkui Li et al. “Distributed consensus of linear multi-agent systems with adaptive dynamic protocols.” In: *Automatica* 49 (2013) (cit. on p. 1).

[21] Kim D. Listmann, Jürgen Adamy, and Luca Scardovi. “Synchronisierung indentischer linearer Systeme - ein Zugang über LMIs.” In: *Automatisierungstechnik* 59 (2011) (cit. on p. 1).

[22] Mehran Mesbahi and Magnus Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010 (cit. on pp. 1, 11, 96, 98, 151).

[23] Luc Moreau. “Stability of continuous-time distributed consensus algorithms.” In: *IEEE Conference on Decision and Control*. 2004 (cit. on p. 1).

[24] Nima Moshtagh, Ali Jadbabaie, and Kostas Daniilidis. “Distributed Geodesic Control Laws for Flocking of Nonholonomic Agents.” In: *IEEE Transactions on Automatic Control* 52 (2007) (cit. on pp. 2, 151).

[25] Behrang Monajemi Nejad, Sid Ahmed Attia, and Jörg Raisch. "Max-Consensus in a Max-Plus Algebraic Setting: The Case of Fixed Communication Topologies." In: *ICAT 2009. XXII International Symposium on Information, Communication and Automation Technologies*. 2009 (cit. on p. 1).

[26] Behrang Monajemi Nejad, Sid Ahmed Attia, and Jörg Raisch. "Max-Consensus in a Max-Plus Algebraic Setting: The Case of Switching Communication Topologies." In: *10th International Workshop on Discrete Event Systems*. 2010 (cit. on p. 1).

[27] Dang Binh Nguyen and Khac Duc Do. "Formation Control of Mobile Robots." In: *International Journal of Computers, Communications & Control* 1 (2006) (cit. on pp. 2, 151).

[28] Mojtaba Nourian et al. "Nash, Social and Centralized Solutions to Consensus Problems via Mean Field Control Theory." In: *IEEE Transactions On Automatic Control* 58 (2013) (cit. on p. 1).

[29] Reza Olfati-Saber, J. Alex Fax, and Richard M. Murray. "Consensus and Cooperation in Networked Multi-Agent Systems." In: *Proceedings of the IEEE*. Vol. 95. 2007 (cit. on pp. 1, 51, 52, 151).

[30] Wei Ren. "Consensus Based Formation Control Strategies for Multi-vehicle Systems." In: *American Control Conference*. 2006 (cit. on pp. 2, 151).

[31] Wei Ren. "Consensus Seeking, Formation Keeping, and Trajectory Tracking in Multiple Vehicle Cooperative Control." PhD thesis. Brigham Young University, 2004 (cit. on pp. 1, 98).

[32] Wei Ren and Ella Atkins. "Second-order Consensus Protocols in Multiple Vehicle Systems with Local Interactions." In: *AIAA Guidance, Navigation and Control Conference*. 2005 (cit. on pp. 1, 2, 151).

[33] Wei Ren and Randal W. Beard. *Distributed Consensus in Multi-vehicle Cooperative Control*. Springer, 2010 (cit. on pp. 1, 11, 96).

[34] Wei Ren and Yongcan Cao. *Distributed Coordination of Multi-agent Networks*. Communications and Control Engineering Series, Springer-Verlag, 2011 (cit. on pp. 1, 11, 51, 52, 96).

[35] Heitor J. Savino, Fernando O. Souza, and Luciano C. A. Pimenta. "Consensus with convergence rate in directed networks with multiple non-differentiable input delays." In: *IEEE Multi-Conference on Systems and Control (MSC)*. 2014 (cit. on pp. 2, 52, 123).

[36] Luca Scardovi and Rodolphe Sepulchre. "Synchronization in networks of identical linear systems." In: *Automatica* 45 (2009) (cit. on p. 1).

[37] E. Semsar-Kazerooni and K. Khorasani. "A Game Theory Approach to Multi-Agent Team Cooperation." In: *American Control Conference*. 2009 (cit. on pp. 1, 55).

[38] E. Semsar-Kazerooni and K. Khorasani. "Optimal Consensus Seeking in a Network of Multiagent Systems: An LMI Approach." In: *IEEE transactions on systems, Man., and Cybernetics* 40 (2010) (cit. on pp. 1, 55).

[39] Jinyan Shao et al. "Leader-following Formation Control of Multiple Mobile Robots." In: *International Symposium on Intelligent Control*. 2005 (cit. on pp. 2, 151).

[40] Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems*. Cambridge University Press, 2008 (cit. on p. 1).

[41] Yuan Gong Sun, Long Wang, and Guangming Xie. "Average consensus in networks of dynamic agents with switching topologies and multiple time-varying delays." In: *Systems & Control Letters* 57 (2008) (cit. on pp. 1, 98).

[42] Herbert G. Tanner, Ali Jadbabaie, and George J. Pappas. *Stability of Flocking Motion*. University of Pennsylvania Technical Report No: MS-CIS-03-03. 2003 (cit. on pp. 2, 151).

[43] Yu-Ping Tian and Cheng-Lin Liu. "Robust consensus of multi-agent systems with diverse input delays and asymmetric interconnection perturbations." In: *Automatica* 45 (2009) (cit. on p. 2).

[44] Thijs H. A. van den Broek, Nathan van de Wouw, and Henk Nijmeijer. "Formation Control of Unicycle Mobile Robots: a Virtual Structure Approach." In: *48th IEEE Conference on Decision and Control*. 2009 (cit. on pp. 2, 151).

[45] René Vidal, Omid Shakernia, and Shankar Sastry. "Formation Control of Nonholonomic Mobile Robots with Omnidirectional Visual Servoing and Motion Segmentation." In: *International Conference on Robotics and Automation*. 2003 (cit. on pp. 2, 151).

[46] Alexander Vladimirov, Ludwig Elsner, and Wolf Jügen Beyn. "Stability and paracontractivity of discrete linear inclusions." In: *Linear Algebra And Its Applications* 312 (2000) (cit. on p. 1).

[47] Jinhuan Wang, Daizhan Cheng, and Xiaoming Hu. "Consensus of Multi–agent linear dynamic systems." In: *Asian Journal of Control* 10 (2008) (cit. on pp. 1, 98).

[48] Rakesh R. Warier, Arpita Sinha, and Srikant Sukumar. "Relative Attitude Trajectory Tracking Using Line Of Sight Measurements Under Spacecraft Position Dynamics." In: *International Conference on Advances in Control and Optimization of Dynamical Systems*. 2014 (cit. on p. 2).

[49] Rakesh R. Warier, Arpita Sinha, and Srikant Sukumar. "Spacecraft Attitude Synchronization and Formation Keeping Using Line Of Sight Measurements." In: *International Federation of Automatic Control World Congress*. 2014 (cit. on p. 2).

[50] Guanghui Wen, Guoqiang Hu, and Wenwu Yu. "Leader-Following Consensus Control for Linear Multi-Agents Systems with Switching Directed Topologies." In: *10th IEEE International Conference on Control and Automation*. 2013 (cit. on pp. 1, 2, 52, 98).

[51] Guanghui Wen and Valery Ugrinovskii. "Distributed Consensus of Linear Multi-Agent Systems with Switching Directed Topologies." In: *2014 Australian Control Conference (AUCC 2014)*. 2014 (cit. on pp. 1, 2, 98).

[52] Guanghui Wen et al. "Consensus tracking of higher-order multi-agent systems with switching directed topologies and occasionally missing control inputs." In: *Systems & Control Letters* 62 (2013) (cit. on pp. 1, 98).

[53] Peter Wieland, Rodolphe Sepulchre, and Frank Allgöwer. "An internal model principle is necessary and sufficient for linear output synchronization." In: *Automatica* 47 (2011) (cit. on p. 1).

[54] Peter Wieland et al. "On consensus in multi-agent systems with linear high-order agents." In: *17th Wold Congress The International Federation of Automatic Control*. 2008 (cit. on p. 1).

[55] F. Xiao and L. Wang. "State consensus for multi agent systems with switching topologies and time varying delays." In: *International Journal of Control* 79 (2006) (cit. on pp. 1, 98).

[56] Xiuxia Yin and Dong Yue. "Event-triggered tracking control for heterogeneous multi-agent systems with Markov communication delays." In: *Journal of the Franklin Institute* 350 (2013) (cit. on p. 2).

[57] Junyan Yu and Long Wang. "Group consensus in multi–agent systems with switching topologies." In: *IEEE Conference on Decision and Control*. 2009 (cit. on pp. 1, 98).

[58] Junyan Yu and Long Wang. "Group consensus in multi-agent systems with switching topologies and communication delays." In: *Systems & Control Letters* 59 (2010) (cit. on pp. 1, 98).

[59] Dong Yue and Xiuxia Yin. "Adaptative Event-Triggered Consensus For Multi-Agent Systems with Stochastic Switching Topologies and Delays." In: *IEEE International Symposium on Industrial Electronics*. 2013 (cit. on pp. 1, 98).

[60] Dongmei Zhang, Xingang Wang, and Li Meng. "Consensus problems for high-order LTI systems: A decentralized static output feedback method." In: *International Journal of Innovative Computing, Information and Control.* 9 (2013) (cit. on pp. 1, 2, 98).

[61] Yuanshu Zheng, Jingying Ma, and Long Wang. "Consensus of switched multi-agent systems." 2015 (cit. on pp. 1, 98).

[62] Bo Zhu, Wei Sun, and Chang Meng. "Position Tracking of Multi Double-Integrator Dynamics by Bounded Distributed Control without Velocity Measurements." In: *2013 American Control Conference (ACC)*. 2013 (cit. on pp. 2, 151).

Control Theory References

[63] Brian D. O. Anderson and John B. Moore. *Linear Optimal Control*. Prentice-Hall, Inc., 1971 (cit. on p. 197).

[64] Brian D. O. Anderson and John B. Moore. *Optimal Control*. Prentice-Hall International, Inc., 1989 (cit. on pp. 192, 194, 197).

[65] Corentin Briat and Alexandre Seuret. “Stability Criteria for Asynchronous Sampled-Data Systems - a Fragmentation Approach.” In: *Proceedings of the 18th IFAC World Congress*. Vol. 18. 2011 (cit. on p. 28).

[66] Chi-Tsong Chen. *Linear System Theory and Design*. 3rd ed. Oxford University Press, 1999 (cit. on p. 194).

[67] R. E. Kalman and J. E. Bertram. “Control System Analysis and Design Via the Second Method of Lyapunov. Part I: Continuous-Time Systems.” In: *Journal of Basic Engineering* (1960) (cit. on pp. 17, 28).

[68] R. E. Kalman and J. E. Bertram. “Control System Analysis and Design Via the Second Method of Lyapunov. Part II: Discrete-Time Systems.” In: *Journal of Basic Engineering* (1960) (cit. on pp. 17, 19, 28).

[69] Hassan K. Khalil. *Nonlinear Systems*. 2nd ed. Prentice-Hall, 1996 (cit. on pp. 17, 28, 192).

[70] J. P. LaSalle. “Stability Theory for Ordinary Differential Equations.” In: *Journal Of Differential Equations* 4 (1968) (cit. on pp. 17, 28).

[71] J. P. LaSalle. *The Stability of Dynamical Systems*. Society For Industrial and Applied Mathematics, 1976 (cit. on pp. 17, 28).

[72] A. M. Lyapunov. “The general problem of the stability of motion.” In: *International Journal of Control: Lyapunov Centenary Issue* 55 (1992) (cit. on p. 17).

[73] Alexandre Seuret. “A novel stability analysis of linear systems under asynchronous samplings.” In: *Automatica* 48 (2011) (cit. on p. 28).

[74] Jean-Jacques E. Slotine and Weiping Li. *Applied Nonlinear Control*. 1st ed. Prentice-Hall, 1991 (cit. on pp. 17, 28).

[75] Kemin Zhou and John C. Doyle. *Essentials of Robust Control*. Prentice Hall, 1998 (cit. on pp. 191, 192, 194, 197).

LMI References

[76] Pierre Apkarian, Hoang Duong Tuan, and Jacques Bernussou. "Continuous-Time Analysis, Eigenstructure Assignment, and $\mathscr{H}_2$ Synthesis With Enhanced Linear Matrix Inequalities (LMI) Characterizations." In: *IEEE Transactions on Automatic Control* 46 (2001) (cit. on p. 199).

[77] Selma Ben Attia, Salah Salhi, and Mekki Ksouri. "LMI Formulation for Static Output Feedback Design of Discrete-Time Switched Systems." In: *Journal of Control Science and Engineering* 2009 (2009) (cit. on p. 23).

[78] P.-A. Bliman. "A convex approach to robust stability for linear systems with uncertain scalar parameters." In: *SIAM Journal on Control and Optimization* 42 (2004) (cit. on p. 26).

[79] P.-A. Bliman. "An existence result for polynomial solutions of parameter-dependent LMIs." In: *Systems & Control Letters* 51 (2004) (cit. on p. 26).

[80] S. Boyd et al. "Linear Matrix Inequality in Systems and Control Theory." In: *Studies in Applied Mathematics, SIAM* (1994) (cit. on pp. 20, 196, 197).

[81] G. Chesi. "On the non-conservatism of a novel LMI relaxation for robust analysis of polytopic systems." In: *Automatica* 44 (2008) (cit. on p. 26).

[82] G. Chesi et al. "Polynomially parameter-dependent Lyapunov functions for robust stability of polytopic systems: An LMI approach." In: *IEEE Transactions on Automatic Control* 50 (2005) (cit. on pp. 25, 26).

[83] César A. R. Crusius and Alexandre Trofino. "Sufficient LMI Conditions for Output Feedback Control Problems." In: *IEEE Transactions On Automatic Control* 44 (1999) (cit. on pp. 155, 201).

[84] Laurent El Ghaoui and Silviu-Iulian Niculescu, eds. *Advances in Linear Matrix Inequality Methods in Control*. SIAM Advances in Design and Control, 2000 (cit. on p. 20).

[85] E. Fridman and U. Shaked. "H_∞-control of linear state-delay descriptor systems: an LMI approach." In: *Linear Algebra and its Applications* 351–352 (2002) (cit. on pp. 127, 178).

[86] Pascal Gahinet and Pierre Apkarian. "A Linear Matrix Inequality approach to $\mathscr{H}_\infty$ Control." In: *Int. J. Robust Nonlinear Control* 4 (1994) (cit. on pp. 20, 197, 199).

[87] Pascal Gahinet and Pierre Apkarian. "An LMI-Based parametrization of all H_∞ controllers with applications." In: *IEEE Conference on Decision and Control*. 1993 (cit. on pp. 20, 197, 199).

[88] Pascal Gahinet et al. *LMI Control Toolbox*. The Math Works, 1995 (cit. on p. 20).

[89] J. C. Geromel, P. L. D. Peres, and S. R. Souza. "Convex Analysis of Output Feedback Control Problems: Robust Stability and Performance." In: *IEEE Transactions On Automatic Control* 41 (1996) (cit. on pp. 155, 156).

[90] J.C. Geromel and P. Colaneri. "Robust stability of time varying polytopic systems." In: *Systems and control letters* 55 (2006) (cit. on p. 25).

[91] J.C. Geromel, C. C. de Souza, and R. E. Skeleton. "Static Output Feedback Controllers: Stability and Convexity." In: *IEEE Transactions On Automatic Control* 43 (1998) (cit. on pp. 25, 155, 156).

[92] Keqin Gu. "An integral inequality in the stability problem of time-delay systems." In: *IEEE Conference on Decision and Control*. Vol. 3. 2000 (cit. on pp. 127, 178, 201).

[93] Keqin Gu and Qing-Long Han. "Controller Design for Time-Delay Systems Using Discretized Lyapunov Functional Approach." In: *IEEE Conference on Decision and Control*. 2000 (cit. on pp. 127, 178).

[94] Keqin Gu and S.-I. Niculescu. "Survey on Recent Results in the Stability and Control of Time-Delay Systems." In: *Journal of Dynamic Systems, Measurement, and Control* 125 (2003) (cit. on pp. 127, 178).

[95] V. J. S. Leite and P. L. D. Peres. "An improved LMI condition for robust $\mathscr{D}$-stability of uncertain polytopic systems." In: *IEEE Transactions on Automatic Control* 48 (2003) (cit. on p. 25).

[96] Johan Löfberg. "YALMIP: A toolbox for modeling and optimization in MATLAB." In: *Proc. 2004 IEEE International Symposium on Computer Aided Control Systems Design*. 2004 (cit. on p. 20).

[97] Maurício C. de Oliveira and José C. Geromel. "H_2 and H_∞ Robust Nonrational Filtering for Linear Delay Systems." In: *IEEE Conference on Decision and Control*. 2003 (cit. on pp. 127, 178).

[98] Mauricio C. de Oliveira and Robert E. Skelton. "On stability tests for linear systems." In: *Proceedings of the 15th IFAC World Congress*. Vol. 15. 2002 (cit. on pp. 20, 199).

[99] R. C. L. F. Oliveira and P. L. D. Peres. "LMI conditions for robust stability analysis based on polynomially parameter-dependent Lyapunov functions." In: *Systems & Control Letters* 55 (2006) (cit. on p. 26).

[100] R. C. L. F. Oliveira and P. L. D. Peres. "Parameter-dependent LMIs in robust analysis: Characterization of homogeneous polynomially parameter-dependent solutions via LMI relaxations." In: *IEEE Transactions on Automatic Control* 52 (2007) (cit. on p. 26).

[101] R. M. Palhares, R. H. C. Taicahashi, and P. L. D. Peres. "H_∞ and H_2 guaranteed costs computation for uncertain linear systems." In: *International Journal of Systems Schience* 28 (1997) (cit. on pp. 20, 197).

[102] D. Peaucelle et al. "A new robust $\mathscr{D}$-stability condition for real convex polytopic uncertainty." In: *Systems & Control Letters* 40 (2000) (cit. on p. 199).

[103] Pedro L. D. Peres and Ricardo C. L. F. Oliveira. *Aulas de Análise e Controle de Sistemas Lineares por Desigualdades Matriciais Lineares (LMIs)*. Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, Departamento de Telemática: `http://www.dt.fee.unicamp.br/~ricfow/IA892/ia892_2s2013.htm`. 2013 (cit. on pp. 20, 156).

[104] G. Pipeleers et al. "Extended LMI characterizations for stability and performance of linear systems." In: *Systems & Control Letters* 58 (2009) (cit. on p. 199).

[105] D. C. W. Ramos and P. L. D. Peres. "An LMI approach to compute robust stability domains for uncertain linear systems." In: *Proceedings of the 2001 American Control Conference*. Vol. 1. 2001 (cit. on p. 25).

[106] D. C. W. Ramos and P. L. D. Peres. "An LMI condition for the robust stability of uncertain continuous-time linear systems." In: *IEEE Transactions on Automatic Control* 47 (2002) (cit. on p. 25).

[107] C. W. Scherer. "Relaxations for robust linear matrix inequality problems with verifications for exactness." In: *SIAM Journal of Matrix Analisys and Applications* 27 (2005) (cit. on p. 26).

[108] Carsten Scherer. "The Riccati Inequality and State-Space H_∞-Optimal Control." PhD thesis. Bayerischen Julius Maximilians-Universität Würzburg, 1990 (cit. on pp. 20, 197).

[109] Carsten Scherer, Pascal Gahinet, and Mahmoud Chilali. "Multiobjective Output-Feedback Control via LMI Optimization." In: *IEEE Transactions On Automatic Control* 42 (1997) (cit. on pp. 20, 197).

[110] Carlos E. de Souza, Reinaldo Martinez Palhares, and Pedro L. D. Peres. "Robust H_∞ Filter Design for Uncertain Linear Systems with Multiple Time-Varying State Delays." In: *IEEE Transactions on Signal Processing* 49 (2001) (cit. on pp. 127, 178).

[111] Jos F. Sturm. "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones." In: *Optimization Methods and Software.* 11 (1999) (cit. on p. 20).

[112] A. Trofino and T. J. M. Dezuo. "LMI stability conditions for uncertain rational nonlinear systems." In: *International Journal of Robust and Nonlinear Control* 24 (2013) (cit. on p. 199).

[113] Jeremy G. VanAntwerp and Richard D. Braatz. "A tutorial on linear and bilinear matrix inequalities." In: *Journal of Process Control* 10 (2000) (cit. on p. 20).

[114] Ji Xiang et al. "LMI Approach to Robust Delay Dependent/Independent Sliding Mode Control of Uncertain Time-delay Systems." In: *IEEE International Conference on Systems, Man and Cybernetics*. 2003 (cit. on pp. 127, 178).

General Mathematic References

[115] Adi Ben-Israel and Thomas N. E. Greville. *Generalized Inverses. Theory and Applications*. 2nd ed. Springer, 2003 (cit. on pp. 189, 191).

[116] J. A. Bondy and U. S. R. Murty. *Graph Theory With Applications*. Elsevier Science Publishing, 1976 (cit. on p. 11).

[117] Reinhard Diestel. *Graph Theory*. 2nd ed. Springer Verlag, 2000 (cit. on p. 11).

[118] Miroslav Fiedler. "A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory." In: *Czechoslovak Mathematica Journal* 25 (1975) (cit. on p. 1).

[119] Miroslav Fiedler. "Algebraic Connectivity of Graphs." In: *Czechoslovak Mathematical Journal* 23 (1973) (cit. on pp. 1, 16, 17).

[120] Paul Finsler. *Über das Vorkommen definiter und semidefiniter Formen in Scharen quadratischer Formen*. Commentarii Mathematici Helvetici. 1937 (cit. on p. 199).

[121] F. R. Gantmacher. *The Theory of Matrices*. Chelsea Publishing Company, 1959 (cit. on p. 194).

[122] Chris Godsil and Gordon Royle. *Algebraic Graph Theory*. Springer-Verlag, 2001 (cit. on p. 11).

[123] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. 4th ed. The Johns Hopkins University Press, 2013 (cit. on pp. 191, 194).

[124] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. 2nd ed. Cambridge University Press, 2013 (cit. on p. 194).

[125] Harold W. Kuhn. "The Hungarian method for the assignment problem." In: *Naval Research Logistics Quarterly* 2 (1955) (cit. on p. 202).

[126] Carl D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000 (cit. on pp. 191, 194).

[127] Bojan Mohar. "The Laplacian Spectrum of Graphs." In: *Graph Theory, Combinatorics, and Applications* 2 (1991) (cit. on p. 16).

[128] Cleve Moler and Charles Van Loan. "Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later." In: *SIAM Review* 45 (2003) (cit. on p. 194).

[129] James Munkres. "Algorithms for the Assignment and Transportation Problems." In: *Journal of the Society for Industrial and Applied Mathematics* 5 (1957) (cit. on p. 202).

[130] Gustaf Söderlind. "The Logarithmic Norm. History and Modern Theory." In: *BIT Numerical Mathematics* 46 (2006) (cit. on p. 194).

[131] Frank Uhlig. "A Recurring Theorem About Pairs of Quadratic Forms and Extensions: A Survey." In: *Linear Algebra and its Applications* 25 (1979) (cit. on p. 199).

Switched Systems References

[132] Michael S. Branicky. “Multiple Lyapunov Functions and Other Analysis Tools for Switched and Hybrid Systems.” In: *IEEE Transactions on Automatic Control* 43 (1998) (cit. on pp. 23, 28).

[133] Graziano Chesi et al. “A Nonconservative LMI Condition for Stability of Switched Systems With Guaranteed Dwell Time.” In: *IEEE Transactions On Automatic Control* 57 (2012) (cit. on pp. 23, 34).

[134] O. L. V. Costa, M. D. Fragoso, and R. P. Marques. *Discrete-Time Markov Jump Linear Systems*. Springer, 2005 (cit. on pp. 23, 28, 36, 38).

[135] Oswaldo L. V. Costa and Marcelo D. Fragoso. “Stability Results for Discrete-Time Linear Systems with markovian Jumping Parameters.” In: *Journal of Mathematical Analysis and Applications* 179 (1993) (cit. on pp. 23, 28, 36, 38).

[136] Jamal Daafouz, Pierre-Alexandre Bliman, and Claude Iung. “Stability Analysis and Control Synthesis for Switched Systems: A switched Lyapunov Function Approach.” In: *IEEE Tramsactopms pm Automatic Control* 47 (2002) (cit. on pp. 23, 28, 34).

[137] Raymond A. Decarlo et al. “Perspectives and Results on the Stability and Stabilizability of Hybrid Systems.” In: *Proc. Of The IEEE: Special Issue Hybrid Systems*. Vol. 88. 2000 (cit. on pp. 23, 28).

[138] Yuguang Fang, Kenneth A. Loparo, and Xiangbo Feng. “Stability of Discrete Time Jump Linear Systems.” In: *Journal of Mathematical Systems, Estimation, and Control* 5 (1995) (cit. on p. 38).

[139] Xiangbo Feng et al. “Stochastic Stability Properties of Jump Linear Systems.” In: *IEEE Transactions on Automatic Control* 37 (1992) (cit. on p. 23).

[140] J.C. Geromel and P. Colaneri. “Stability and stabilization of discrete time switched Systems.” In: *International Journal of Control* 79 (2006) (cit. on pp. 23, 28, 34).

[141] J.C. Geromel and P. Colaneri. “Stabilization of Continuous-Time Switched Systems.” In: *Proceedings of the 16th IFAC World Congress*. Vol. 16. 2005 (cit. on pp. 23, 34).

[142] A. P. C. Gonçalves et al. “H_∞ State Feedback Control of Discrete-time Markov Jump Linear Systems through Linear Matrix Inequalities.” In: *18th IFAC World Congress*. 2011 (cit. on pp. 23, 36).

[143] Sergio Grammatico, Anantharaman Subbaraman, and Amdrew R. Teel. “Discrete-time stochastic control systems: examples of robustness to strictly causal perturbations.” In: *IEEE Conference on Decision and Control*. 2013 (cit. on pp. 23, 36).

[144] L. V. Hien, N. T. Dzung, and H. Trinh. “Stochastic stability of nonlinear discrete-time Markovian jump systems with time-varying delay and partially unknown transition rates.” In: *Neurocomputing* 175 (2016) (cit. on pp. 23, 36).

[145] Yasushi Iwatani, Shogo Arai, and Koichi Hashimoto. “Stability of Switched Stochastic Systems in Discrete-Time.” In: *SIAM Journal of Control, Measurement, and System Integration* 3 (2010) (cit. on pp. 23, 36).

[146] Yuandong Ji and Howard Jay Chizeck. “Controllability, Stabilizability, and Contrinuous-Time Markovian Jump Linear Quadratic Control.” In: *IEEE Tramsactions on Automatic Control* 35 (1990) (cit. on p. 23).

[147] I. Ia. Kats and N. N. Krasovskii. “On the Stability of Systems with Random Parameters.” In: *PMM* 24 (1960) (cit. on p. 23).

[148] Ji-Woong Lee and Geir E. Dullerud. “Uniform stabilization of discrete time switched and Markovian jump linear systems.” In: *Automatica* 42 (2006) (cit. on pp. 23, 36).

[149] Ji-Woong Lee and Pramod P. Khargonekar. “Optimal Output Regulation for Discrete-Time Switched and Markovian Jump Linear Systems.” In: *SIAM Journal on Control and Optimization* 47 (2008) (cit. on pp. 23, 36).

[150] Yan Li, Weihai Zhang, and Xikui Liu. “Stability of Nonlinear Stochastic Discrete-Time Systems.” In: *Hindawi Journal of Applied Mathematics* 2013 (2013) (cit. on pp. 23, 36, 37).

[151] Daniel Liberzon. *Switching in Systems and Control*. 1st ed. Birkhäuser, 2003 (cit. on pp. 23, 24).

[152] Hai Lin and Panos J. Antsaklis. “Stability and Stabilizability of Switched Linear Systems: A Survey of Recent Results.” In: *IEEE Transactions On Automatic Control* 54 (2009) (cit. on p. 23).

[153] Jan Lunze and Francoise Lamnabhi-Lagarrigue, eds. *Handbook Of Hybrid Systems Control*. 1st ed. Cambridge University Press, 2009 (cit. on p. 23).

[154] V. F. Montagner et al. “State feedback control of switched linear systems: An LMI approach.” In: *Journal of Computational and Applied Mathematics* 194 (2006) (cit. on p. 23).

[155] Miguel Parada Contzen and Jörg Raisch. “A polytopic approach to switched linear systems.” In: *IEEE Multi-Conference on Systems and Control (MSC)*. 2014 (cit. on p. 24).

[156] Peng Shi and Fanbiao Li. “A Survey of Markovian Jump Systems: Modeling and Design.” In: *International Journal of Control, Automation, and Systems* 13 (2015) (cit. on p. 23).

[157] Robert Shorten et al. “Stability Criteria For Switched and Hybrid Systems.” In: *SIAM Review* 49 (2007) (cit. on p. 23).

[158] Xi-Ming Sun, Georgi M. Dimirovski, and Jun Zhao. “An LMI Method for Stabilizing Second-order Switched Systems.” In: *Proceedings of the 2006 American Control Conference*. 2006 (cit. on p. 23).

[159] A. R. Teel, J. P. Hespanha, and A. Subbaraman. "Equivalent Characterizations of Input-to-State Stability for Stochastic Discrete-Time Systems." In: *IEEE Transactions on Automatic Control* 59 (2014) (cit. on pp. 23, 36).

[160] Umesh Vaidya. "Stochastic Stability Analysis of Discrete Time System Using Lyapanov Measure." In: *American Control Conference*. 2015 (cit. on pp. 23, 36).

[161] Mark A. Wicks, Philipos Peleties, and Raymond A. DeCarlo. "Construction of Piecewise Lyapunov Functions for Stabilizing Switched Systems." In: *Proceedings of the 33rd Conference on Decision and Control*. 1994 (cit. on pp. 23, 28).

[162] Junlin Xiong and James Lam. "Stabilization of discrete-time Markov jump linear systems via time-delayed controllers." In: *Automatica* 42 (2006) (cit. on pp. 23, 36).

[163] Jun Xu and Lihua Xie. "Homogeneous Polynomial Lyapunov Functions for Piecewise Affine Systems." In: *American Control Conference*. 2005 (cit. on pp. 23, 28).

[164] Xueshan Yang and George Miminis. "Stability of Discrete Deterministic and Stochastic Nonlinear Systems." In: *Journal of Mathematical Analysis and Applications* 169 (1992) (cit. on pp. 23, 36).

[165] J. Yin. "Asymptotic stability in probability and stabilization for a class of discrete-time stochastic systems." In: *International Journal of Robust and Nonlinear Control* 25 (2015) (cit. on pp. 23, 36).

[166] Guisheng Zhai et al. "Piecewise Lyapunov Functions for Switched Systems With Average Dwell Time." In: *Asian Journal of Control* 2 (2000) (cit. on pp. 23, 28).

[167] Guisheng Zhai et al. "Stability Analysis of Switched Systems with Stable and Unstable Subsystems: An Average Dwell Time Approach." In: *American Control Conference*. Vol. 1. 2000 (cit. on pp. 23, 28).

[168] Lixian Zhang and James Lam. "Necessary and Sufficient Conditions for Analysis and Synthesis of Markov Jump Linear Systems With Incomplete Transition Descriptions." In: *IEEE Transactions On Automatic Control* 55 (2010) (cit. on pp. 23, 36).

[169] Tianliang Zhang et al. "Robust Stability, Stabilization, and H_∞ Control of a Class of Nonlinear Discrete Time Stochastic Systems." In: *Hindawi Mathematical Problems in Engineering* 2016 (2016) (cit. on pp. 23, 36).

[170] Wei Zhang et al. "On piecewise quadratic control-Lyapunov functions for switched linear systems." In: *Preceedings of the 48th IEEE Conference on Decision and Control*. 2009 (cit. on pp. 23, 28, 34).

[171] Weihai Zhang and Tianliang Zhang. "Lyaounov-type Theorems for Linear Discrete Time-Varying Stochastic Systems with Multiplicative Noise." In: *Chinese Control and Decision Conference (CCDC)*. 2015 (cit. on pp. 23, 36).

Power Systems References

[172] Florian Dörfler and Francesco Bullo. "Kron reduction of graphs with applications to electrical networks." In: *IEEE Transactions On Circuits and Systems* 60 (2013) (cit. on pp. 13, 131).

[173] A. E. Fitzgerald, Charles Kingsley, and Stephen D. Umans. *Electric Machinery*. 5th ed. McGraw-Hill, 1990 (cit. on p. 203).

[174] Josep M. Guerrero et al. "Advanced Control Architectures for Intelligent MicroGrids, Part I." In: *IEEE Transactions on Industrial Electronics* 60 (2013) (cit. on p. 131).

[175] Prabha Kundur. *Power System Stability and Control*. McGraw-Hill, 1994 (cit. on pp. 132, 203).

[176] Yiqi Liu et al. "Enhanced Load Power Sharing Accuracy in Droop-Controlled DC Microgrids with Both Mesh and Radial Configurations." In: *Energies* 8 (2015) (cit. on pp. 23, 36).

[177] R. Palma-Behnke et al. "A Social SCADA Approach for a Renewable based Microgrid - The Huatacondo Project." In: *Power and Energy Society General Meeting*. 2011 (cit. on p. 131).

[178] J. A. Peças Lopes, C. L. Moreira, and A. G. Madureira. "Defining Control Strategies for MicroGrids Islanded Operation." In: *IEEE Transactions on Power Systems* 21 (2006) (cit. on pp. 131, 133, 143).

[179] Nagaraju Pogaku, Milan Prodanović, and Timothy C. Green. "Modeling, Analysis and Testing of Autonomous Operation of an Inverter-Based Microgrid." In: *IEEE Transactions on power electronics* 22 (2007) (cit. on p. 131).

[180] Joan Rocabert et al. "Control of Power Converters in AC Microgrids." In: *IEEE Transactions on Power Electronics* 27 (2012) (cit. on pp. 131, 133, 143).

[181] Johannes Schiffer et al. "A consensus-based distributed voltage control for reactive power sharing in microgrids." In: *13th ECC*. 2014 (cit. on pp. 131, 133, 136, 143, 144).

[182] Johannes Schiffer et al. "Droop-controlled inverter-based microgrids are robust to clock drifts." In: *American Control Conference*. 2015 (cit. on pp. 131, 133).

[183] Johannes Schiffer et al. "On power sharing and stability in autonomous inverter-based microgrids." In: *51st IEEE Conference on Decision and Control*. 2012 (cit. on pp. 131, 133).

[184] Johannes Schiffer et al. "Synchronization of droop-controlled microgrids with distributed rotational and electronic generation." In: *52nd IEEE CDC*. 2013 (cit. on pp. 131, 133).

[185] John W. Simpson-Porco, Florian Dörfler, and Francesco Bullo. “Synchronization and power sharing for droop-controlled inverters in islanded microgrids.” In: *Automatica* 49 (2013) (cit. on p. 131).

Declaration

I declare that this thesis has been composed by myself, that the work contained herein is my own except where explicitly stated otherwise, and that this work has not been submitted for any other degree or professional qualification except as specified.

Berlin, October 2017

Miguel Parada Contzen