

Decomposition Methods for Distributed Control and Identification

Paolo Massioni

Cover design: Andrea Simonetto

DECOMPOSITION METHODS FOR DISTRIBUTED CONTROL AND IDENTIFICATION

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op
maandag 21 juni 2010 om 12:30 uur

door

Paolo MASSIONI

Ingenere Aerospaziale, Politecnico di Milano, Italië
geboren te Milaan, Italië

Dit proefschrift is goedgekeurd door de promotor:

Prof. dr. ir. M. Verhaegen

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. M. Verhaegen,	Technische Universiteit Delft, promotor
Prof. dr. ir. B. De Schutter,	Technische Universiteit Delft
Prof. dr. C. W. Scherer,	Universität Stuttgart
Prof. dr. G. E. Dullerud,	University of Illinois at Urbana-Champaign
Prof. dr. R. S. Smith,	University of California
Dr. T. Keviczky,	Technische Universiteit Delft
F. Ankersen, MSc,	European Space Agency
Prof. dr. ir. J. Hellendoorn,	Technische Universiteit Delft, reservelid



This dissertation has been completed in partial fulfillment of the requirements for the graduate study of the Dutch Institute of Systems and Control (DISC).



The work presented in this thesis has been supported by the MicroNed programme.

ISBN: 978-94-6113-011-2

Copyright © 2010 by Paolo Massioni.

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission from the copyright owner.

Printed in The Netherlands.

Acknowledgments

This thesis is the result of my four years of work in the Delft Center for Systems and Control of the Delft University of Technology. In this short note, I would like to acknowledge the help and support of the people who made all of this possible.

The first person I wish to thank is Michel, to whom I am grateful for being my promotor and for giving me the possibility of achieving all of this, and for the trust and support that he gave me in the course of the four years. I have really appreciated the freedom that I have been given, and the fact that none of my requests has ever been unheard.

Next, I wish to thank the other people who made this work directly possible. I would like to thank Tamás, whose help really made the difference for me in many occasions, and Rufus, for always being able to listen to me and answer my questions while we were sharing the room in these four years. A special thanks goes to Justin, who has been my fellow in the jungle of the research, a friend and an excellent travelling companion too. Other people without the help of whom this work would have not been the same are İlhan, Hakan, Joost, Ivo, Jan-Willem and especially Carsten, to whom I will always be grateful for his time and for the unforgettable experience of the “LMI work camp”. I should not forget to mention Navin, Eric, Rouzbeh, Daan, Andrey, Rudy and Gianni either, whom I all have to thank for different reasons. Finally, I am grateful to Eberhard Gill and Finn Ankersen for the opportunity that they have given me to work with them, and I would also like to thank the committee members for their time and efforts in reading my thesis and providing feedback.

I will also thank those who did not help me directly with this work, but contributed to make these years in Delft either less boring, funnier or more interesting. I will start by mentioning my colleagues, that I would all like to thank for the time shared together. Next I would like to thank all the friends that I have been so lucky to meet here, I will just name a few in strict alphabetical order, Andrea, Delphine, Eleonora, Lara, Maria, Mathieu, Paolo, Sergio “il pizzaiolo”, Sinar, Snezana, Vincenzo and Wei Qing. A special mention goes to the three fellows in Politecnico who decided to follow my path and come to Delft for a PhD: Ligeia, Federico and Valentina. A thank you too to the chaplains of the International Student Chaplaincy, Ben, Waltraut and Avin.

At last, I do wish to thank those who have been far from me in these last years, but whose presence and friendship I did continue to feel for all the time, my good old

friends from Italy: Alessandro, Angela, Chiara, Dario, il “Dolcio”, Erika, Fabio, “Loiax”, Sara, il Sergio, Stefano, la Vale, il “Vigo” and lo “Zambo”. Finally, I would like to thank my parents Biancarosa and Antonio, and my brother Francesco, for all the love and support, and for accepting my decision of a life far from my native country.

Paolo Massioni

Contents

Acknowledgments	vii
1 Introduction	1
1.1 Motivation	1
1.2 Contribution of this thesis	3
1.2.1 Relevant work in the field	3
1.2.2 Scope of this thesis and contribution to the field	7
1.3 Structure of this thesis	8
2 Preliminaries	11
2.1 Introduction	11
2.1.1 Notation	12
2.2 Graphs and Laplacian matrices	12
2.3 Kronecker product and its properties	16
2.4 Earlier results on formation stability	17
2.5 Decomposable systems	20
2.5.1 Definition	21
2.5.2 System norms	25
2.5.3 The decomposition theorem	26
2.6 Special cases	27
2.6.1 Circulant systems	27
2.6.2 Symmetrically interconnected systems	28
3 Control of Decomposable Systems	29
3.1 Introduction	29
3.2 A decomposition approach to control	31
3.2.1 Distributed control	31
3.2.2 The basic idea	32

3.2.3	Controller synthesis methods	34
3.2.4	Links with graph theory	40
3.2.5	Special cases	41
3.3	Examples of applications	42
3.3.1	Formation flying of satellites in a circular orbit	42
3.3.2	Formation flying in deep space	45
3.3.3	Paper machines	50
3.4	Variations on the theme	52
3.4.1	The full block S-procedure	52
3.4.2	Full block S-procedure and decentralized control	53
3.4.3	Full block S-procedure and distributed control	54
3.4.4	Small gain theorem for distributed control	57
3.4.5	Evaluation of numerical results	57
3.5	Conclusions	58
4	Adaptive Optics Applications	61
4.1	Introduction	61
4.2	Adaptive optics	62
4.3	Distributed control of decomposable systems	63
4.4	Spectra of influence matrices	66
4.4.1	The infinite dimensional case	67
4.4.2	From infinite to finite	69
4.5	Simulation results	71
4.6	Conclusions	73
5	Formation Flying Applications	75
5.1	Introduction	75
5.2	Linear Time-Periodic models	77
5.3	Distributed controller synthesis for Linear Time-Periodic models	78
5.4	Test cases	81
5.4.1	The FAST mission	81
5.4.2	Formation in a halo orbit	87
5.5	Conclusions	89

6	Identification	93
6.1	Introduction	93
6.2	Identification of circulant systems	94
6.2.1	Circulant systems and their properties	94
6.2.2	The identification algorithm	101
6.2.3	Some simulation results	108
6.3	Identification of decomposable systems in general	112
6.3.1	Differences with respect to circulant system identification	112
6.3.2	The novel algorithm	114
6.3.3	Numerical results	118
6.4	Other special cases	119
6.5	Conclusions	123
7	Conclusions	125
7.1	Summary and considerations	125
7.2	Further research directions of this work	126
7.3	The future of distributed control	127
	Bibliography	129
	List of Symbols and Notation	137
	List of Abbreviations	139
	Summary	141
	Samenvatting	143
	Curriculum Vitae	145

1

CHAPTER

Introduction

The recent progress in technology, as for example, in miniaturization and microtechnologies, has made it now possible to create devices made up of a huge set of small subunits, which must cooperate in order to achieve a common goal. These new developments, together with advances in automation and autonomous control of formations of vehicles are now forcing control engineers to confront themselves with systems of incredibly high dimensionality, with an ever growing number of input and output channels. For such systems, which we call “large scale systems”, it is necessary to take a new, smarter approach in order to solve control problems in a reasonable time, as well as for being able to design controllers which can be realized in a physically implementable way.

The scope of this thesis is to propose a new set of methods for dealing with a special class of such large scale systems, for which a decomposition property will make it possible to overcome the curse of dimensionality and to design controllers which will be implementable in a distributed way, as a collection of simple, small units.

1.1 Motivation

The students who decide to specialize in system and control engineering usually start their studies with the techniques that in the past have been developed for linear systems with a single input and a single output, better known as SISO. Classical control engineering tools as Bode plots, root loci and Nyquist diagrams are more than enough to deal with such kind of systems. Students who continue their course of studies later on have to face a kind of paradigm shift, when the focus is put on more and more complex systems, and the number of inputs and outputs grows. It is at this point that the students are introduced to the state-space approach, that is considered more flexible for multiple-input, multiple-output (or MIMO) systems.

A second kind of paradigm shift might be needed now, due to recent technological advances. Developments in miniaturization, Micro Electro-Mechanical Systems (MEMS) and microfluidics, just to name a few, have made it possible to construct devices with very high number of inputs and outputs, like arrays of micro cantilevers, or deformable mirrors for adaptive optics. As a motivating example, we could report a few numbers from a telescope that is currently being designed, namely the European Extremely Large Telescope (E-ELT) of the European Southern Observatory (ESO) [3]. This telescope will feature a 42 m diameter primary mirror, made of 1000 hexagonal segments. A deformable secondary mirror with more than 5000 actuators will be used for the adaptive optics system.

Systems of this size might be difficult to manage with standard techniques. The thing that comes to our aid, is that usually such “large scale systems” have a certain *structure* which can be exploited in order to reduce the computational complexity of the problem, e.g. the system might be thought of as a set of subunits interacting with one another, as shown in Figure 1.1 on the left-hand side.

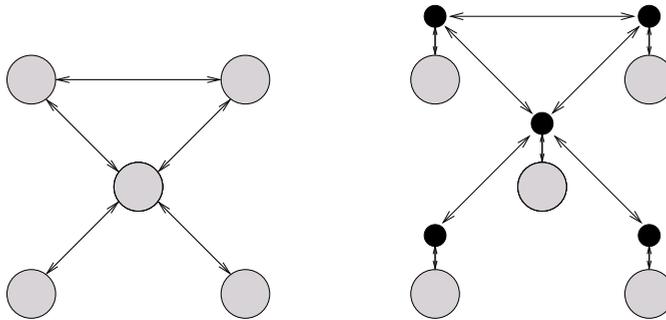


Figure 1.1: On the left, a distributed system made of the interconnection of five subsystems; the arrows represent the interaction that the subsystems have, in terms e.g. of signals that they exchange. On the right, the concept of distributed control is shown: the controller is implemented as a set of local controllers (the smaller circles) which have the same interconnection pattern as the plant.

Another difficulty involved in controlling large scale systems is the *implementation* of the control system itself. For a system with a huge number of inputs and outputs, the classical approach of a centralized controller collecting all the outputs and deciding all the inputs is not feasible in practice. This causes the necessity of considering localized or *distributed* approaches as shown in Figure 1.1 on the right-hand side. The centralized controller is replaced by a number of small, simpler units, each of which controlling a single subunit of the plant; these local controllers communicate with each other with the same interaction structure as the plant.

1.2 Contribution of this thesis

Large scale systems have been a topic of interest in system and control theory since the seventies [78]. Most of the existing literature focuses on the problem of *distributed control* of large scale systems, making use of some special technique that can either help reducing the computational complexity of the controller design, or enforce the constraint of synthesizing a controller with a distributed structure as well. These techniques are usually tailored to the structure of specific classes of system.

In this section we will first review those which we judge the most relevant results in the field, which cover a set of different kinds of distributed or large scale systems. After this, we will show the goals of this thesis work and the contributions that it has made.

1.2.1 Relevant work in the field

The literature in distributed control is vast and a multitude of different approaches has been proposed and analyzed. In general, as pointed out in [76], it is unknown whether the problem of synthesizing a controller that keeps the same structure as the plant can be efficiently cast into a convex optimization problem, without introducing any conservatism of some kind. This explains why the most relevant results in the field usually either involve only *stability* consideration or *suboptimal performance*. A classification of the literature in distributed control can be made by distinguishing the classes of systems that are examined.

A first distinction should be made between the classes of *distributed parameter systems* [15] and *lumped systems*. Distributed parameter systems are uncountably infinite-order systems, which typically are the result of dynamics described by partial differential equations. The state of a distributed parameter system is a vector function x depending on a spatial coordinate ζ (assuming values on a real interval, or cartesian product of intervals for multiple dimensional systems), and on a time variable, either continuous (t) or discrete (k). The dynamical equation of such system is then of the form:

$$\dot{x}(t, \zeta) = f(x(t, \zeta), \frac{\partial}{\partial \zeta} x(t, \zeta), \zeta, t) + u(t, \zeta) \quad (1.1)$$

where u denotes an input. Such equation has to be complemented with an initial condition and boundary conditions in order to define a well-posed problem. A practical example of such kind of formulas is given by the heat equation; if we consider the problem of the spatio-temporal evolution of the temperature profile of a bar of length l , then we can assume the state (the temperature) x as a function of the time t and the spatial coordinate $\zeta \in [0, l]$; the dynamic law is the following:

$$\dot{x}(t, \zeta) = c \frac{\partial^2}{\partial \zeta^2} x(t, \zeta) + u(t, \zeta) \quad (1.2)$$

where c is a constant depending on the material; two boundary conditions are needed to complete the problem, for example:

$$x(t, 0) = x(t, l) = x_0 \quad (1.3)$$

which correspond to a constant temperature at the edges of the bar.

Distributed parameter systems though are not the subject of this thesis, where we focus instead on *lumped systems*, which are a discrete kind of systems, where the number of states is either finite or countable. The discretization of distributed parameter systems leads to lumped systems: for example, we can take (1.2) and decide to approximate the continuous temperature distribution x with a discrete distribution χ ; this state variable will not be depending on ζ anymore, but on a discrete index which we call i , which can assume integer values from 1 to N ; ideally, we would have $\chi(t, i) = x(t, \frac{l}{N-1}(i-1))$. A partial derivative approximation scheme would then yield the following dynamic equation:

$$\dot{\chi}(t, i) = \frac{c(N-1)}{2l}(-\chi(t, i-1) + 2\chi(t, i) - \chi(t, i+1)) + u(t, i) \text{ for } i = 2 \dots N-1 \quad (1.4)$$

and the boundary conditions:

$$\chi(t, 0) = \chi(t, N) = x_0. \quad (1.5)$$

Lumped systems do not only originate from distributed parameter systems, but can also represent systems which have an inherent discrete spatial form, for example a set of subsystems (or *agents*) each one with its own dynamics and interacting with the others. For example, consider a set of N masses disposed on a line, each of them connected to the one preceding and the one following by a spring. If we call x_i the displacement with respect to the equilibrium position of the i^{th} body, then the motion of such body is described by the equation:

$$\ddot{x}_i(t) = -\frac{c_{i-1}}{m_i}(x_i - x_{i-1}) - \frac{c_i}{m_i}(x_i - x_{i+1}) + u_i(t) \quad (1.6)$$

(where c_i is the spring constant and m_i is the mass of the bodies). The description as in (1.6) can be seen as a *local* kind of information, as it describes the dynamics of each single subsystem. Another approach is possible, in the sense that we can consider the system as a whole, joining all the state variables together in a single state vector $x(t) = [x_1(t)^T \ x_2(t)^T \ \dots \ x_N(t)^T]^T$, the inputs in one input vector, etc. The dynamic equation of the system then becomes the familiar state-space description of a dynamical system:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (1.7)$$

where the matrices A and B will have a special, sparse structure that is due to the “distributedness” of the original system; for example, A will be block tridiagonal, with the entries in the diagonal blocks representing the inherent second-order dynamics of the oscillators, and the off-diagonal terms representing the interaction between neighboring masses. We call this state-space system the *lifted* system; this

thesis will make extensive use of this formulation.

The difference between distributed parameter systems and lumped systems is graphically rendered in Figure 1.2.

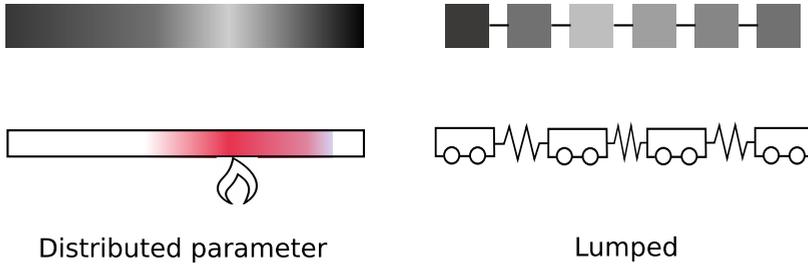


Figure 1.2: A first classification of distributed systems can be distinguishing between distributed parameter systems and lumped systems. The heat conduction on a bar is an example of the former, while the line of oscillators is an example of the latter.

The second criterion for classification can be distinguishing *heterogeneous* systems [48; 65; 74] from *homogeneous* systems [10; 16; 26]. Heterogeneous systems consist of the combination of a set of interacting subsystems that are different from one another, whereas for homogeneous systems all the subsystems are the same. Lumped homogeneous systems typically portray *multi-agent* systems, i.e. platoons or formations of robots/vehicles. The main objective of the literature focusing on heterogeneous systems is usually the synthesis of structured controllers, which means controllers that can be implemented in a distributed way. For homogeneous systems instead, in most cases it is possible to exploit the regularity of the system to obtain a significant reduction in the complexity of the problem as well, which results in methods for controller synthesis with a complexity that is independent of the system size (i.e., the number of subsystems).

At last, we can point out that in the literature some authors assume an *arbitrary interconnection* structure among the subsystems [10; 26; 48; 65], which is usually tackled by making use of graph theory, while others assume a regular *lattice structure* [16; 44; 73; 74; 82]. A lattice structure can allow introducing the formalism of spatial groups, or it makes it possible to consider the system as multi-dimensional, where the spatial variable is assimilated to an additional temporal variable (with both causal and anti-causal dynamics). Figure 1.3 shows graphically the differences among the classes of lumped distributed systems.

This thesis concerns *lumped homogeneous systems with arbitrary interconnections*.

We now review the results of some of the most recent and most cited work in the field.

For what concerns the analysis and controller synthesis of heterogeneous systems with arbitrary interconnections, a well-known result is reported in [48]. Here a technique for analysis and synthesis with \mathcal{H}_∞ performance is developed making

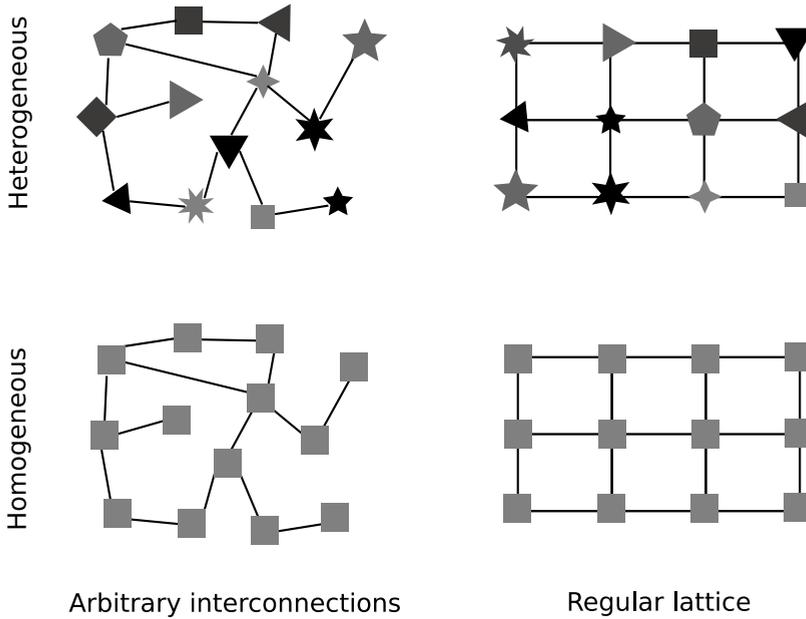


Figure 1.3: Different classes of lumped distributed systems.

use of passivity theory and gain-scheduling control, and cast as LMI (Linear Matrix Inequality) tests. This result is very general and allows designing distributed controllers for a vast class of systems, but the computational load grows very quickly with the number of subsystems involved, as the size of the synthesis LMIs grows proportionally with the number of subsystems.

A different result, still making use of LMIs for analysis and synthesis with \mathcal{H}_∞ performance, is reported in [22]. The methods therein apply to heterogeneous systems with lattice interconnection, and they are derived from results from Linear Time-Varying systems [23]. Another relevant result for the same class of systems (heterogeneous with regular lattice structure) is reported in [74], where it is shown that such systems can be described by means of matrices with a Sequentially Semi-Separable (SSS) structure. Operations on SSS matrices (including solving Riccati equations) can be made in a very fast way (the computational complexity grows linearly with the size of the system) and they are structure-preserving, thus allowing the synthesis of heterogeneous distributed controllers.

Other very famous results, which apply to homogeneous systems on a lattice, are reported in [6]. Here, it is shown that for infinite lattices or finite ones with “wrap around” boundary conditions, a Fourier transform can decouple the system, thus allowing a simpler approach to analysis and design problems. For the same class of systems, a different approach, based on LMIs deriving from a multi-dimensional version of the Kalman-Yakubovich-Popov lemma [71], can be found in [16]. This method exploits the structure of the system and the computational complexity of the controller synthesis problem turns out to be independent of the

system size.

At last let us cite two papers focusing on the same subject as this thesis, namely homogeneous systems with arbitrary interconnections. Reference [10] regards the problem of designing a distributed LQR controller for a set of identical agents which have couplings only in their performance output. It is shown that, for any number of agents, it is always possible to solve a “small” LQR synthesis problem involving only a limited number of subsystems, and then extend it to the full system with guaranteed stability. This method represents a good example of controller synthesis whose computational complexity is independent of the number of subsystems. The same class of systems is subject of [26], where it is shown that sets of identical agents with performance output couplings (which are called “formations”) can be described as a parameter-dependent linear system of the order of one agent only. This leads to a very simple and intuitive stability test for this class of systems, based on an extension of the Nyquist plot, which again does not depend on the number of subsystems. Later on in the paper, a stabilizing controller synthesis method is shown, together with an “information flow” filter which allows a quicker arrival at the steady state. The first part of this last paper can be considered as a starting point for this thesis work, and it will be discussed in more details in Section 2.4 in the next chapter.

1.2.2 Scope of this thesis and contribution to the field

As said, this thesis work regards *lumped homogeneous systems with arbitrary interconnections*, and it can be considered as a continuation in the line of research of [10; 26]. These papers provide methods for computing distributed *stabilizing* controllers for sets of identical agents coupled to one another by their output signals. These methods are computationally attractive because their complexity does not grow with the number of agents. In this thesis, we try to answer the following questions:

1. *In the literature [10; 26] we have found methods for analysis and controller synthesis which apply to homogeneous systems coupled by arbitrary interconnections on the output signals. These methods allow checking the stability of the system or designing a stabilizing controller, and their computational complexity does not grow with the number of agents. Is it possible to extend such results in order to account for \mathcal{H}_2 and \mathcal{H}_∞ performance indices as well?*
2. *Is it possible to account for couplings among the agents other than those in the output signals, e.g. for dynamic couplings?*

The other question that we will try to answer regards the problem of identification:

3. *How can we find, from input/output data, structured models for which we can apply the distributed control techniques developed in 1. and 2.?*

The second question will be answered first, as we will define a class of Linear Time-Invariant distributed systems, which includes the ones in [10; 26] as a subset.

We will call these systems *decomposable systems*: the reason of the name is that for these systems an approach based on a sort of modal decomposition applies. The decomposition depends only on the interconnection structure and not on the specific system, and it is a consequence of the properties of the Kronecker algebra [12].

Exploiting this “decomposition theorem” as we will call it, we will then answer the first question: we will show that by employing a variety of tools from the robust control literature (which make use of Linear Matrix Inequalities), it will be possible to generate controller synthesis methods for obtaining suboptimal performance in a variety of cases (e.g. \mathcal{H}_2 or \mathcal{H}_∞ performance, continuous or discrete time, state or output feedback). Some of these methods will have a computational complexity that does not depend on the actual number of subsystems, so they will be applicable to systems of any size. We will also see that we can use results from graph theory in order to gain insight into the properties of the possible interconnection structures and arrive at controllers that are not depending on these structures either. In addition, we will show how these distributed controller synthesis methods can be applied to the field of adaptive optics, namely to the control of an adaptive mirror for wavefront correction, and to satellite formation flying problems. For this last purpose, one of the synthesis methods will be extended to be able to cope with time-varying dynamics, thus enabling the possibility of considering any periodic orbit whose dynamics can be approximated as Linear Time-Varying.

The last question will be answered at the end of the thesis, as we will show that the decomposition theorem can be used also in the framework of system identification, i.e. for identifying models of decomposable systems from data. The problem is first treated for a special case, namely for the class of circulant systems, and then examined in the general case, which will prove to be a more challenging task.

The summary of the contributions, and by means of which tools they are achieved, is graphically shown in Figure 1.4.

1.3 Structure of this thesis

This thesis contains seven chapters, the first of which is this introduction. Chapter 2 contains the preliminary notions that are used throughout the thesis. Chapters 3 to 5 ideally form a subpart in the thesis that is dedicated to the problem of distributed control, while Chapter 6 is focused on the complementary problem, namely distributed identification. The conclusions are in Chapter 7.

The detailed content of the main chapters is summarized here. The main references where the content of each chapter has been published are reported as well.

- **Chapter 2** starts with an introduction to graph theory and Kronecker product algebra. Later on, the class of *decomposable systems* is defined and explained. The results in [26] are recapitulated and used as an introduction and justification for focusing on this class of systems.

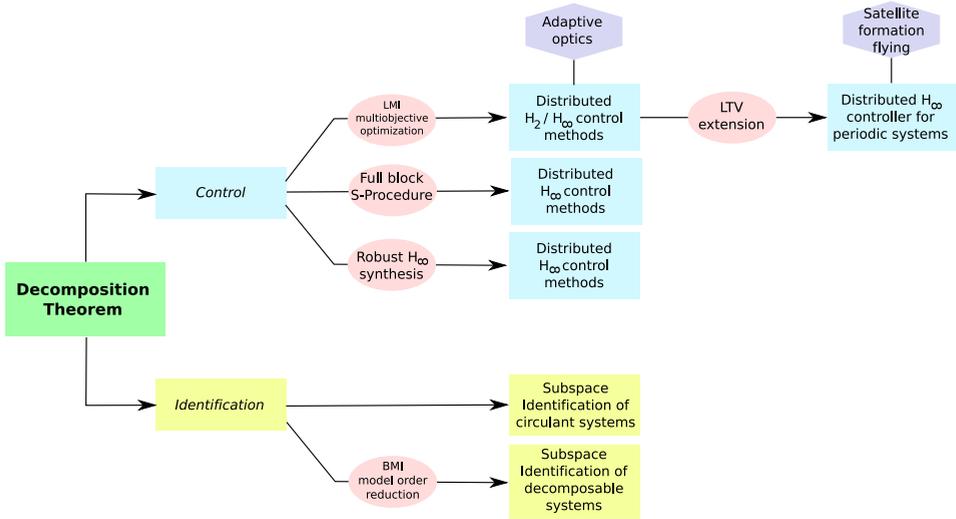


Figure 1.4: Scheme of the contributions of this thesis.

- **Chapter 3** contains the main contribution of the thesis, namely a set of distributed controller synthesis methods which apply to Linear Time-Invariant decomposable systems.

References:

P. Massioni and M. Verhaegen. Distributed control for identical dynamically coupled systems: a decomposition approach. *IEEE Transactions on Automatic Control*, 54(1):124–135, January 2009.

P. Massioni and M. Verhaegen. A full block S-procedure application to distributed control. *Accepted for publication in the proceedings of the 2010 American Control Conference*, Baltimore, USA.

- **Chapter 4** contains a simulation of application of one of the synthesis methods of Chapter 3 to a problem in the field of adaptive optics.

Reference:

P. Massioni, R. Fraanje, and M. Verhaegen. Adaptive optics application of distributed control design for decomposable systems. In *Proceedings of the 48th IEEE Conference on Decision & Control*, Shanghai, China, December 2009.

- **Chapter 5** regards the problem of satellite formation flying, and the extension of one of the methods of Chapter 3 to Linear Time-Varying dynamics. Simulations of application to two test cases are shown at the end of the chapter.

Reference:

P. Massioni, T. Keviczky, E. Gill, and M. Verhaegen. A decomposition based approach to linear time-periodic distributed control of satellite formations.

IEEE Transactions on Control Systems Technology, to appear.

- **Chapter 6** is focused on the problem of identifying decomposable systems from input/output data.

References:

P. Massioni and M. Verhaegen. Subspace identification of circulant systems. *Automatica*, 44(11):2825–2833, November 2008.

P. Massioni and M. Verhaegen. Subspace identification of distributed, decomposable systems. In *Proceedings of the 48th IEEE Conference on Decision & Control*, Shanghai, China, December 2009.

Preliminaries

This chapter contains a brief introduction to the general concepts that will be used in the course of this thesis. We start by recalling one of the famous results in distributed control theory, the paper on formation stability by Fax and Murray [26]. This paper introduces an analysis method for a class of systems that are described by means of graph-theoretical tools. We use this earlier result as a starting point for introducing a more general class of systems, the “decomposable systems”, that is the subject of this thesis. Such systems have the remarkable property that they can be decomposed into a set of smaller, independent modal systems through a sort of “coordinate transformation” of inputs, outputs and state. This property will allow control and identification problems related to these systems to be approached under the point of view of the small modal systems, which will prove to be more convenient than the centralized, global approach.

2.1 Introduction

In this chapter we are going to introduce the main preliminary notions which we will need in the course of this thesis. The first tool that is presented, that has become very popular in control of multi-agent or distributed system, or in the so-called consensus theory, is the theory of graphs. The first interesting aspect of graphs is that they provide a simple and friendly way of modelling interactions between systems, with the visual aid of a set of nodes, which represent the elements, and a set of arrows connecting the nodes, which represent the signals that the nodes exchange or the dynamical interactions that they might have. The other key aspect is that a very strong mathematical theory is associated with graphs [14; 31], allowing the visual aid to be turned into a powerful formal instrument. Papers like [10; 26; 43] are just examples of the vast use of graphs that nowadays is done in control theory.

In particular, we are going to review in this chapter the results of [26], which can be seen as a “precursor” of the methods shown in this thesis (especially of Chapter 3). In this article Fax and Murray have shown that the stability of a formation of vehicles, whose interactions can be described by a graph, can be considered equivalent to the stability of a set of independent systems depending on a parameter. This observation leads to a link between distributed systems of a certain kind and Linear Parameter-Varying (LPV) systems. On inspiration of this earlier result, we will define the class of “decomposable systems”, for which a very general decomposition theorem applies. This decomposition theorem will enable the developments that are shown in Chapters 3 and 6. The main difference between this new decomposition theorem and the observation in Fax and Murray’s paper is that the new results offer a real equivalence between the global system and the set of independent systems, not only in terms of stability properties, but also in terms of input-output relations. This gives the possibility of evaluating the system performance in terms of system norms, and that is the key for the development of performance-based controller synthesis methods.

The chapter is organized as follows. In Section 2.2 a brief review of graph theory is given, with a tutorial style. This review is intentionally not complete, and it aims only at showing the concepts that are relevant for this thesis; there are many other results in the field that can be of use in this scope, but they are somehow not essential, and we have decided not to report them. Section 2.3 introduces one of the other tools that is extensively used in this thesis, namely the Kronecker product, and its properties. In Section 2.4 we report some of the results of [26], which will allow us to introduce the definition of “decomposable systems”, the subject of this thesis, in Section 2.5. In the same section the properties of these systems are introduced as well, and finally Section 2.6 reports some cases of decomposable systems of special interest.

2.1.1 Notation

We denote the field of real numbers by \mathbb{R} , the field of complex numbers by \mathbb{C} and the set of real (complex) $n \times m$ matrices by $\mathbb{R}^{n \times m}$ ($\mathbb{C}^{n \times m}$). Let \otimes indicate the Kronecker product, I_n the identity matrix of order n and let j be the imaginary unit. The notation $A \succ 0$ ($A \prec 0$) indicates that all the eigenvalues of the Hermitian matrix A are strictly positive (negative). For a matrix A , A^T indicates its transpose and A^H indicates its Hermitian (complex conjugate transpose); \bar{a} indicates the complex conjugate of a matrix or scalar a . The bullet \bullet denotes a symbol that is either not relevant or clear from the context, and the star $*$ will be used to replace parts of expressions involving symmetric matrices that can be inferred by the symmetry itself.

2.2 Graphs and Laplacian matrices

A few elements of graph theory are often used in the course of this thesis. For this reason, we will summarize here a few definitions and a theorem that will be

fundamental in the following pages. The interested reader can find out more in [20] and [14].

Definition 2.1 (graphs) A directed graph \mathcal{G} consists of a set of vertices (or nodes) \mathcal{V} and a set of edges $\mathcal{E} \subset \mathcal{V}^2$, which can be interpreted as connections between vertices: for an edge (a, b) , we call the vertex a the initial vertex or tail and b the terminal vertex or head. We assume that each element of \mathcal{E} is unique. A graph with the property that $(a, b) \in \mathcal{E} \Leftrightarrow (b, a) \in \mathcal{E} \forall a, b \in \mathcal{V}$ is called an undirected graph.

Directed graphs are typically depicted as a set of circles and arrows. The circles represent the nodes, whereas the arrows connecting them represent the edges, as in Figure 2.1.

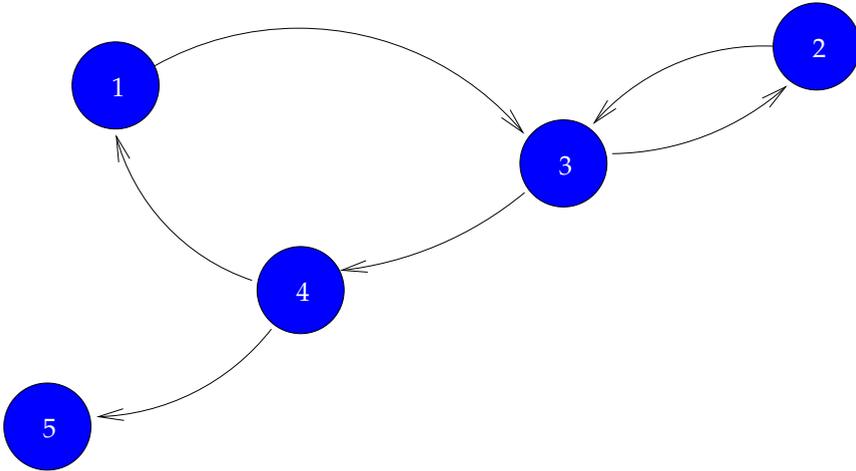


Figure 2.1: An example of directed graph.

Graph theory becomes useful in control thanks to the properties of special kinds of matrices which are associated with graphs. For introducing these matrices, we assume that the N vertices of the graph \mathcal{G} are enumerated, and each of them is denoted a_i .

Definition 2.2 (normalized adjacency matrix) The normalized adjacency matrix A of a graph \mathcal{G} with N vertices is an $N \times N$ matrix defined by $A_{i,k} = 1/d_o(a_i)$ if $(a_i, a_k) \in \mathcal{G}$ and 0 otherwise; $d_o(a_i)$ is the out-degree of a_i , that is, the number of edges that feature a_i as their tail.

Example 2.1 (normalized adjacency matrix)

The graph in Figure 2.1 has the following normalized adjacency matrix:

$$\mathcal{A} = \left[\begin{array}{ccccc} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 \end{array} \right] \left. \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \right\} \begin{array}{l} \text{outgoing} \\ \text{connections} \end{array} \quad (2.2)$$

$$\underbrace{\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array}}_{\text{incoming connections}}$$

Definition 2.3 The normalized Laplacian matrix¹ \mathcal{L} of a graph is defined as $\mathcal{L} = I_N - \mathcal{A}$.

Example 2.2 (normalized Laplacian matrix)

The graph in Figure 2.1 has the following normalized Laplacian matrix:

$$\mathcal{L} = I_N - \mathcal{A} = \left[\begin{array}{ccccc} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 0 & 0 & 1 & -\frac{1}{2} \\ 0 & 0 & 0 & 0 & 1 \end{array} \right] \quad (2.4)$$

Normalized Laplacians have special properties that we are going to use, which are stated in the following theorem and corollaries.

Theorem 2.1 The eigenvalues of the normalized Laplacian are located in a disk of radius 1 centered at $1 + 0j$ in the complex plane.

Proof: [4] Consider the adjacency matrix \mathcal{A} first, and notice that by construction all of its entries are non-negative and every row sums up to either 1 or 0, so $0 \leq \sum_{k=1}^N \mathcal{A}_{i,k} \leq 1$. Let $v \in \mathbb{R}^N$ be a left eigenvector of \mathcal{A} , and λ the associated eigenvalue. It holds that:

$$\sum_{i=1}^N \mathcal{A}_{i,k} v_i = \lambda v_k \quad (2.5)$$

taking the absolute value of this expression, we have:

$$|\lambda v_k| = |\lambda| |v_k| = \left| \sum_{i=1}^N \mathcal{A}_{i,k} v_i \right| \leq \sum_{i=1}^N \mathcal{A}_{i,k} |v_i| \quad (2.6)$$

¹In [26] this matrix is called just ‘‘Laplacian’’; but, as stated there, in literature there are different definitions for it. In this thesis we define the matrix used here as ‘‘normalized’’, in order to distinguish it from the other definition of the Laplacian matrix, that can be found e.g. in [31].

and then summing over k , we get:

$$|\lambda| \sum_{k=1}^N |v_k| \leq \sum_{i=1}^N \sum_{k=1}^N \mathcal{A}_{i,k} |v_i| \leq \sum_{i=1}^N |v_i|. \quad (2.7)$$

This implies $|\lambda| \leq 1$, that is, the eigenvalues of \mathcal{A} are located in a disk of radius 1 centered in the origin. The eigenvalues of the normalized Laplacian $\mathcal{L} = I_N - \mathcal{A}$ are simply $1 - \lambda$, proving what was stated. \square

Corollary 2.1 For undirected graphs the eigenvalues of the normalized Laplacian are real (thus they are located between 0 and 2).

Proof: Normalized Laplacians of undirected graphs can be factorized as the product of a positive definite matrix and a symmetric matrix. Such matrix products have real eigenvalues [69]. \square

Example 2.3 (normalized Laplacian of an undirected graph)

Consider the graph in Figure 2.1 in its undirected version (assume that for every edge, also the inverse edge exists). Its normalized Laplacian matrix would be:

$$\begin{aligned} \mathcal{L} &= \begin{bmatrix} 1 & 0 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & 1 & -1 & 0 & 0 \\ -\frac{1}{3} & -\frac{1}{3} & 1 & -\frac{1}{3} & 0 \\ -\frac{1}{3} & 0 & -\frac{1}{3} & 1 & -\frac{1}{3} \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & -1 & -1 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 \\ -1 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \end{aligned} \quad (2.9)$$

The second matrix in the product is what in literature (e.g. in [31]) is called Laplacian.

It is also possible to construct Laplacian matrices that are indeed symmetric for symmetric graphs, using the notion of a weighted normalized Laplacian matrix. In this case, we consider that a weight is associated to every edge. The introduction of weighted self-loops (edges from one node to itself) can be used to balance the Laplacian and make it symmetric. This is shown in the next example.

Example 2.4 (weighted Laplacians)

Consider again the graph in Figure 2.1 in its undirected version. We introduce a self loop for the first node, and a double self loop for the second and the fifth node. This leads to the following normalized adjacency matrix:

$$\mathcal{A}_w = \begin{bmatrix} \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & \frac{2}{3} & \frac{1}{3} & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 0 & \frac{1}{3} & \frac{2}{3} \end{bmatrix} \quad (2.12)$$

yielding the following normalized Laplacian:

$$\mathcal{L}_w = \begin{bmatrix} \frac{2}{3} & 0 & -\frac{1}{3} & -\frac{1}{3} & 0 \\ 0 & \frac{1}{3} & -\frac{1}{3} & 0 & 0 \\ -\frac{1}{3} & -\frac{1}{3} & 1 & -\frac{1}{3} & 0 \\ -\frac{1}{3} & 0 & -\frac{1}{3} & 1 & -\frac{1}{3} \\ 0 & 0 & 0 & -\frac{1}{3} & \frac{1}{3} \end{bmatrix} \quad (2.13)$$

Notice that this last matrix is symmetric.

Weighted normalized Laplacians have the same properties as normalized Laplacians in general (Theorem 2.1 and Corollary 2.1).

In the coming pages we will show a result that relates the stability properties of a formation to the eigenvalues of a normalized Laplacian matrix; this explains why it can be very important to have *a priori* information on the location of these eigenvalues.

2.3 Kronecker product and its properties

In this thesis we will often deal with distributed or sparse systems, which feature sparse matrices in their state-space realization. A useful tool for expressing blockwise the sparsity of matrices is the Kronecker product [12] of two matrices $P \in \mathbb{R}^{n \times m}$ and $Q \in \mathbb{R}^{j \times k}$, defined as:

$$P \otimes Q = \begin{bmatrix} p_{1,1}Q & \cdots & p_{1,n}Q \\ \vdots & \ddots & \vdots \\ p_{m,1}Q & \cdots & p_{m,n}Q \end{bmatrix} \in \mathbb{R}^{nj \times mk} \quad (2.14)$$

where $p_{a,b}$ is the element of P in the a^{th} row and b^{th} column.

The Kronecker product has some special properties that we are going to use, and which are stated in the two following lemmas.

Lemma 2.1 Consider the matrices P, Q, R and S . If the number of rows and columns of such matrices is compatible such that the products PR and QS are meaningful, then

$$(P \otimes Q)(R \otimes S) = (PR \otimes QS). \quad (2.15)$$

Lemma 2.2 If P and Q are invertible matrices, then

$$(P \otimes Q)^{-1} = P^{-1} \otimes Q^{-1}. \quad (2.16)$$

2.4 Earlier results on formation stability

In this section we report briefly the results of [26], which can be considered as the starting point for the methods that are presented in this thesis. The paper focuses on continuous-time systems, but the same reasonings apply to discrete time as well. Let us consider a set of N identical linear systems (agents, vehicles, etc.), whose dynamics is modeled by the equation:

$$\dot{x}_i = Ax_i + Bu_i \quad (2.17)$$

where $x_i \in \mathbb{R}^l$ are the agents' states, $u_i \in \mathbb{R}^m$ are their control inputs and $i \in \{1, \dots, N\}$ is the index for the vehicles in the formation. From this it follows that the dynamics of the formation is described by the equation:

$$\dot{x} = (I_N \otimes A)x + (I_N \otimes B)u \quad (2.18)$$

where we have $x = [x_1^T \ x_2^T \ \dots \ x_N^T]^T \in \mathbb{R}^{lN}$ and $u = [u_1^T \ u_2^T \ \dots \ u_N^T]^T \in \mathbb{R}^{mN}$.

Let us now assume that each vehicle has a limited visibility with respect to the others; for this purpose we define the set $\mathcal{J}_i \subset \{1, \dots, N\} \setminus \{i\}$ of the vehicles that the i^{th} vehicle can sense. Then, each vehicle has the following measurements available for feedback control:

$$y_i = C_a x_i$$

$$q_i = \begin{cases} \frac{1}{|\mathcal{J}_i|} C_b \sum_{j \in \mathcal{J}_i} (x_i - x_j) & \text{if } |\mathcal{J}_i| \neq 0 \\ C_b x_i & \text{if } |\mathcal{J}_i| = 0 \end{cases} \quad (2.19)$$

where $|\mathcal{J}_i|$ is the number of elements of the set \mathcal{J}_i . In this way, the global output function is equivalent to:

$$\begin{aligned} y &= (I_N \otimes C_a)x \\ q &= (\mathcal{L} \otimes C_b)x \end{aligned} \quad (2.20)$$

where as before $y = [y_1^T \ y_2^T \ \dots \ y_N^T]^T \in \mathbb{R}^{r_y N}$ and $q = [q_1^T \ q_2^T \ \dots \ q_N^T]^T \in \mathbb{R}^{r_q N}$; thanks to the choice for the definition of q in (2.19) \mathcal{L} is then indeed the normalized

Laplacian of the graph that describes the information flow in the formation (i.e., an edge connects vertex i to vertex k if and only if agent k receives the output of agent i).

Let us now assume that each vehicle is locally controlled by identical local controllers K of order s :

$$\begin{aligned} \dot{v}_i &= K_{A_v} v_i + K_{B_y} y_i + K_{B_q} q_i \\ u_i &= K_{C_v} v_i + K_{D_y} y_i + K_{D_q} q_i \end{aligned} \quad (2.21)$$

Then the following theorem holds.

Theorem 2.2 (formation stability) *A local controller K as in (2.21) stabilizes the formation dynamics in (2.17), (2.19) if and only if it simultaneously stabilizes the following set of N systems:*

$$\begin{aligned} \dot{\hat{x}}_i &= A\hat{x}_i + B\hat{u}_i \\ \hat{y}_i &= C_a\hat{x}_i \\ \hat{q}_i &= \lambda_i C_b\hat{x}_i \end{aligned} \quad (2.22)$$

where the λ_i are the eigenvalues of the matrix \mathcal{L} of (2.20).

Proof: [26] *The dynamics of the system in (2.18) and (2.19) in closed-loop with the controller (2.21) is described by:*

$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} \quad (2.23)$$

where

$$\begin{aligned} A_{11} &= I_N \otimes (A + BK_{D_y} C_a) + \mathcal{L} \otimes (BK_{D_z} C_b) \\ A_{12} &= I_N \otimes (BK_C) \\ A_{21} &= I_N \otimes (K_{B_y} C_a) + \mathcal{L} \otimes (K_{B_z} C_b) \\ A_{22} &= I_N \otimes K_A \end{aligned} \quad (2.24)$$

Now consider that for any square matrix \mathcal{L} there exists a Schur transformation [33] such as:

$$\mathcal{L} = T^{-1}UT \quad (2.25)$$

where T is unitary and U is upper triangular, with the eigenvalues of \mathcal{L} on the diagonal; both T and U can be complex-valued. Let us define:

$$\tilde{x} = (T \otimes I_l)x, \quad \tilde{v} = (T \otimes I_s)v \quad (2.26)$$

We can then restate (2.23) in terms of \tilde{x} and \tilde{v} ; in this case the matrix elements will be:

$$\begin{aligned} \tilde{A}_{11} &= (T \otimes I_l)A_{11}(T \otimes I_l)^{-1} \\ \tilde{A}_{12} &= (T \otimes I_l)A_{12}(T \otimes I_s)^{-1} \\ \tilde{A}_{21} &= (T \otimes I_s)A_{21}(T \otimes I_l)^{-1} \\ \tilde{A}_{22} &= (T \otimes I_s)A_{22}(T \otimes I_s)^{-1} \end{aligned} \quad (2.27)$$

As a consequence of the properties of Kronecker product (Lemmas 2.1 and 2.2), we have

that:

$$\begin{aligned}
 \tilde{A}_{11} &= I_N \otimes (A + BK_{D_y}C_a) + U \otimes (BK_{D_z}C_b) \\
 \tilde{A}_{12} &= I_N \otimes (BK_C) \\
 \tilde{A}_{21} &= I_N \otimes (K_{B_y}C_a) + U \otimes (K_{B_z}C_b) \\
 \tilde{A}_{22} &= I_N \otimes K_A
 \end{aligned} \tag{2.28}$$

Notice that the elements of the transformed system matrix are either block diagonal (\tilde{A}_{12} and \tilde{A}_{22}) or block upper-triangular (\tilde{A}_{11} and \tilde{A}_{21}); this implies that the stability of this system is equivalent to the stability of the N subsystems defined taking the diagonal blocks. As the diagonal of U contains the eigenvalues λ_i of \mathcal{L} , then the equations of the N diagonal subsystems are:

$$\begin{aligned}
 \dot{\tilde{x}}_i &= (A + BK_{D_y}C_a + \lambda_i BK_{D_z}C_b)\tilde{x}_i + BK_C\tilde{v}_i \\
 \dot{\tilde{v}}_i &= (K_{B_y}C_a + \lambda_i K_{B_z}C_b)\tilde{x}_i + K_A\tilde{v}_i
 \end{aligned} \tag{2.29}$$

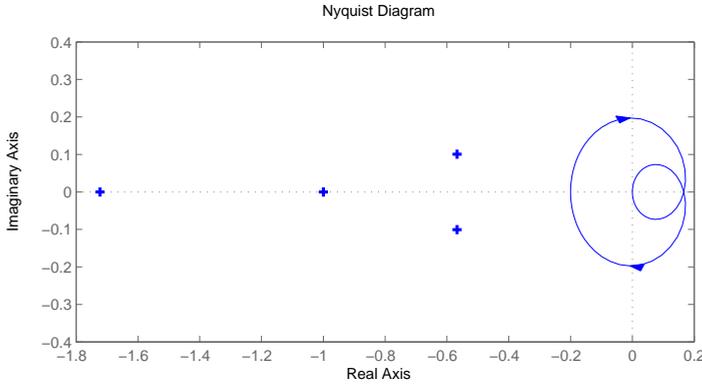
which is equivalent to the controller (2.21) stabilizing (2.22). \square

It has to be pointed out that λ_i can be complex, leading to complex-valued systems. The key element of the proof is the fact that for any square matrix there exists a Schur transformation that makes it possible to see the global system as a cascade of local systems (that is, an “upper triangular system”). This result is valid for any matrix \mathcal{L} , not only for Laplacians, but the use of Laplacians will allow having information on the λ_i without computing them.

It is also possible to derive a kind of Nyquist criterion for the formations in the case of single-input single-output agents; by comparing the agent equations and the subsystem equations (2.22), it is clear that if the transfer function from u_i to $C_b x_i$ of an agent is G , then the transfer functions of the subsystems (from \hat{u}_i to \hat{q}_i) are $\lambda_i G$. Then, if we assume a feedback on q alone, if K is the transfer function of the controller, the formation is stable if and only if the point -1 is correctly encircled by the Nyquist diagram of all of the closed-loop transfer functions $\lambda_i GK$ (see [26] for details.). This is equivalent to saying that the function GK must correctly encircle all the points $-\frac{1}{\lambda_i}$, so a single Nyquist diagram can be enough to grant the stability of the formation. Moreover, in certain situations, these points are restricted *a priori* to be in certain specific areas: for example, for normalized Laplacians of undirected formations, we will have $-\frac{1}{\lambda_i} \in \mathbb{R}$ and $-\infty \leq -\frac{1}{\lambda_i} \leq -\frac{1}{2}$ (Theorem 2.1). A Nyquist diagram that correctly encircles all the region where $-\frac{1}{\lambda_i}$ is constrained to be located will grant stability for all undirected formations; so with a single, simple test it is possible to prove a very general result.

Example 2.5 (Nyquist criterion for formations)

Consider the graph in Figure 2.1, and a stable loop gain transfer function with the following Nyquist diagram:



The crosses in the picture represent the positions of $-\frac{1}{\lambda_i}$ for the normalized Laplacian of the graph we consider. As the diagram does not encircle any one of them, the formation described by this graph and the chosen loop gain is stable. Actually, it is marginally stable as \mathcal{L} has also a zero eigenvalue that corresponds to a non-controllable part of the formation, that is basically the “drift effect” of the formation as a whole. This accounts for the fact that it is impossible to stabilize the absolute position of the formation if only relative measurements between its members are taken. Notice also that the diagram does never reach real values smaller than $-\frac{1}{2}$. Considering that the values of $-\frac{1}{\lambda_i}$ for a normalized Laplacian can only be in the area where the real part is smaller than $-\frac{1}{2}$ (that is where the disk where the eigenvalues are located is mapped), this means that the chosen “local” loop gain function GK yields stable formations for any graph, as long as normalized Laplacian matrices are used to describe them.

2.5 Decomposable systems

In this section we extend the class of systems described in (2.18) and (2.19) to a more general class of systems, which we call “decomposable systems”. These systems are all the linear systems for which a change of variables like in (2.26) generates a decomposition into smaller, “modal” systems. We will also replace the Schur transformation with a diagonalization: although this might seem a loss of generality, this choice introduces a great advantage as it will allow *performance* considerations too, and not only stability checks as in Theorem 2.2.

2.5.1 Definition

In this section we will introduce the class of systems that are subject of this thesis. We start by defining the class of matrices which are of interest in their description.

Definition 2.4 Assume that the matrix $\mathcal{P} \in \mathbb{R}^{N \times N}$, which we call “pattern matrix”, is diagonalizable (i.e. there exists an invertible \mathcal{S} such that $\mathcal{S}^{-1}\mathcal{P}\mathcal{S}$ is diagonal [37]): let us define as $\mathcal{G}_{\mathcal{P},p,q}$ the set of all matrices $\mathcal{M} \in \mathbb{R}^{Np \times Nq}$ for which there exist two matrices $M_a, M_b \in \mathbb{R}^{p \times q}$ such that:

$$\mathcal{M} = I_N \otimes M_a + \mathcal{P} \otimes M_b. \quad (2.30)$$

We state an interesting property of these matrices that will be used in the sequel.

Lemma 2.3 Let $S \in \mathbb{C}^{N \times N}$ be a non-singular matrix such that $\Lambda = S^{-1}\mathcal{P}S$ is diagonal. If $\mathcal{M} \in \mathcal{G}_{\mathcal{P},p,q}$ then:

$$\mathbf{M} = (S \otimes I_p)^{-1} \mathcal{M} (S \otimes I_q) \quad (2.31)$$

is block diagonal.

Proof: From Definition 2.4, we can write:

$$\mathbf{M} = (S \otimes I_p)^{-1} (I_N \otimes M_a + \mathcal{P} \otimes M_b) (S \otimes I_q) \quad (2.32)$$

then from the properties of the Kronecker product (Lemmas 2.1 and 2.2) we have:

$$\mathbf{M} = (S^{-1}I_N S \otimes I_p M_a I_q) + (S^{-1}\mathcal{P}S \otimes I_p M_b I_q) \Leftrightarrow \mathbf{M} = I_N \otimes M_a + \Lambda \otimes M_b. \quad (2.33)$$

Since I_N and Λ are diagonal, we have that \mathbf{M} is block diagonal. \square

So it is immediate to find a kind of similarity transformation that renders a matrix in $\mathcal{G}_{\mathcal{P},p,q}$ block diagonal, once the matrix that diagonalizes \mathcal{P} is known. We also notice that the matrix \mathbf{M} of the previous lemma is not just any block diagonal matrix, but it is a matrix that can be parameterized according to (2.33). If we call \mathbf{M}_i the i^{th} block in the diagonal of \mathbf{M}_i , it is easy to show that:

$$\mathbf{M}_i = M_a + \lambda_i M_b \quad (2.34)$$

where λ_i is the i^{th} entry in the diagonal of Λ , which is the i^{th} eigenvalue of \mathcal{P} . We define as $\mathcal{B}_{\mathcal{P},p,q}$ the set of block diagonal matrices whose blocks satisfy (2.34), and for them we state the following corollary that will be useful later on.

Corollary 2.2 Let $S \in \mathbb{C}^{N \times N}$ be a non-singular matrix such that $\Lambda = S^{-1}\mathcal{P}S$ is diagonal. Then we have that:

$$\mathcal{N} = (S \otimes I_p) \mathbf{N} (S \otimes I_q)^{-1} \in \mathcal{G}_{\mathcal{P},p,q} \quad (2.35)$$

if and only if $\mathbf{N} \in \mathcal{B}_{\mathcal{P},p,q}$.

Proof: The proof is trivial and analogous to the one of Lemma 2.3. \square

Notice that this corollary implies that if $N \notin \mathcal{B}_{\mathcal{P},p,q} \Rightarrow (S \otimes I_p)N(S \otimes I_q)^{-1} \notin \mathcal{G}_{\mathcal{P},p,q}$.

Remark 2.1 From now on, in this thesis we will always use the bold font to identify matrices that can be parameterized according to (2.34).

In this thesis we focus on linear dynamical systems such that the system matrices of their state-space representation are all in the set $\mathcal{G}_{\mathcal{P},\bullet,\bullet}$ for the same matrix \mathcal{P} . We shall consider discrete time systems of the kind:

$$\begin{cases} x(k+1) = \mathcal{A}x(k) + \mathcal{B}_w w(k) + \mathcal{B}_u u(k) \\ z(k) = \mathcal{C}_z x(k) + \mathcal{D}_{zw} w(k) + \mathcal{D}_{zu} u(k) \\ y(k) = \mathcal{C}_y x(k) + \mathcal{D}_{yw} w(k) \end{cases} \quad (2.36)$$

as well as continuous time systems:

$$\begin{cases} \dot{x}(t) = \mathcal{A}x(t) + \mathcal{B}_w w(t) + \mathcal{B}_u u(t) \\ z(t) = \mathcal{C}_z x(t) + \mathcal{D}_{zw} w(t) + \mathcal{D}_{zu} u(t) \\ y(t) = \mathcal{C}_y x(t) + \mathcal{D}_{yw} w(t) \end{cases} \quad (2.37)$$

where, as in the notation usually found in literature, $k \in \mathbb{Z}$, $t \in \mathbb{R}$, u is the input to the system, w is a disturbance, y is the measured output and z is the output on which the performance of the system is evaluated. We can now define the set of systems that we will study in the course of this thesis.

Definition 2.5 (decomposable systems) Let us consider the linear dynamical systems described by (2.36) or (2.37). We call such systems “decomposable systems” if and only if $\mathcal{A} \in \mathcal{G}_{\mathcal{P},l,l}$, $\mathcal{B}_w \in \mathcal{G}_{\mathcal{P},l,m_w}$, $\mathcal{B}_u \in \mathcal{G}_{\mathcal{P},l,m_u}$, $\mathcal{C}_z \in \mathcal{G}_{\mathcal{P},r_z,l}$, $\mathcal{C}_y \in \mathcal{G}_{\mathcal{P},r_y,l}$, $\mathcal{D}_{zw} \in \mathcal{G}_{\mathcal{P},r_z,m_w}$, $\mathcal{D}_{zu} \in \mathcal{G}_{\mathcal{P},r_z,m_u}$ and $\mathcal{D}_{yw} \in \mathcal{G}_{\mathcal{P},r_y,m_w}$ for some matrix \mathcal{P} . Notice that the order of the system is Nl , and that $u \in \mathbb{R}^{N m_u}$, $w \in \mathbb{R}^{N m_w}$, $y \in \mathbb{R}^{N r_y}$ and $z \in \mathbb{R}^{N r_z}$.

In the case that the pattern matrix \mathcal{P} is symmetric, then we call the system a “symmetric decomposable system”.

In the course of this thesis we will mainly look at discrete time systems, for reasons that will be clear from the next chapter.

Decomposable systems can be thought of as the result of the interconnection of a large number of identical subsystems (Figure 2.2). The structure of the interconnection is given by the pattern matrix, which has as many columns and rows as the number N of subsystems. Each entry of the pattern matrix determines whether two subsystems influence each other or not, i.e., the element in the p^{th} row, q^{th} column is non-zero if and only if the p^{th} subsystem is influenced by the q^{th} one. Moreover, the nature of the interaction among subsystems can be of different kinds. For example, it can be dynamic, in the sense that the states of each

subsystem influence the states of the ones to which it is connected; this will result in non-zero off-diagonal terms in the state matrix, that is, in a non-zero A_b matrix. Another possibility is that the coupling is introduced in the performance index, implying non-zero $C_{z,b}$, $D_{zu,b}$ or $D_{zw,b}$ matrices. Also other situations are possible, for example we can think of coupling introduced by the input matrices. Section 3.3 of the following chapter, and Chapters 4 and 5 later on, will show a number of systems of interest which can fit into this class.

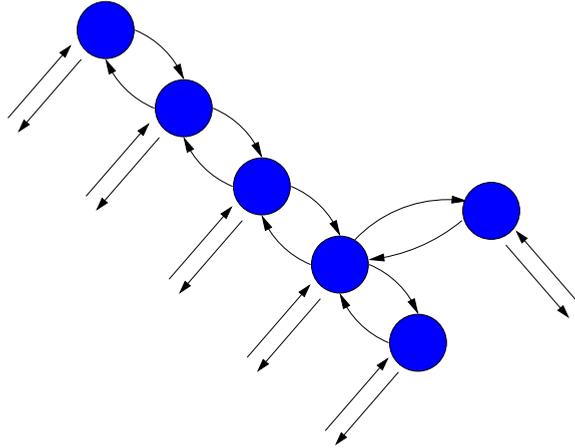


Figure 2.2: A system made of the interconnection of more identical subsystems. The straight arrows represent input/output signals with the external environment, while the curved stand for interactions between subsystems.

We now present a theorem which is of fundamental importance for the results shown in this thesis.

Theorem 2.3 (decomposition property) *A decomposable system of order Nl as described in Definition 2.5 is equivalent in terms of input-output map to N independent subsystems of order l . Each of these subsystems has only m_u inputs, m_w disturbances, r_z performance outputs and r_y control outputs.*

Proof: According to Lemma 2.3, every matrix \mathcal{M} appearing in the state-space description of the system can be rewritten as:

$$\mathcal{M} = (\mathcal{S} \otimes I_p) \mathbf{M} (\mathcal{S} \otimes I_q)^{-1} \quad (2.38)$$

with $\mathbf{M} \in \mathcal{B}_{\mathcal{P},p,q}$, block diagonal, with p, q assuming appropriate values. We can then

rewrite the system equations (e.g. (2.36), those of the discrete time case) as:

$$\begin{cases} (\mathcal{S} \otimes I_l)^{-1}x(k+1) = \mathbf{A}(\mathcal{S} \otimes I_l)^{-1}x(k) + \mathbf{B}_w(\mathcal{S} \otimes I_{m_w})^{-1}w(k) + \\ \quad + \mathbf{B}_u(\mathcal{S} \otimes I_{m_u})^{-1}u(k) \\ (\mathcal{S} \otimes I_{r_z})^{-1}z(k) = \mathbf{C}_z(\mathcal{S} \otimes I_l)^{-1}x(k) + \mathbf{D}_{zw}(\mathcal{S} \otimes I_{m_w})^{-1}w(k) + \\ \quad + \mathbf{D}_{zu}(\mathcal{S} \otimes I_{m_u})^{-1}u(k) \\ (\mathcal{S} \otimes I_{r_y})^{-1}y(k) = \mathbf{C}_y(\mathcal{S} \otimes I_l)^{-1}x(k) + \mathbf{D}_{yw}(\mathcal{S} \otimes I_{m_w})^{-1}w(k) \end{cases} \quad (2.39)$$

Then, with the following (invertible) change of variables:

$$\begin{aligned} x &= (\mathcal{S} \otimes I_l)\hat{x} \\ w &= (\mathcal{S} \otimes I_{m_w})\hat{w} \\ u &= (\mathcal{S} \otimes I_{m_u})\hat{u} \\ z &= (\mathcal{S} \otimes I_{r_z})\hat{z} \\ y &= (\mathcal{S} \otimes I_{r_y})\hat{y} \end{aligned} \quad (2.40)$$

the system finally becomes:

$$\begin{cases} \hat{x}(k+1) = \mathbf{A}\hat{x}(k) + \mathbf{B}_w\hat{w}(k) + \mathbf{B}_u\hat{u}(k) \\ \hat{z}(k) = \mathbf{C}_z\hat{x}(k) + \mathbf{D}_{zw}\hat{w}(k) + \mathbf{D}_{zu}\hat{u}(k) \\ \hat{y}(k) = \mathbf{C}_y\hat{x}(k) + \mathbf{D}_{yw}\hat{w}(k) \end{cases} \quad (2.41)$$

where the system matrices \mathbf{A} , \mathbf{B}_w , \mathbf{B}_u , \mathbf{C}_z , etc. are all block diagonal. This is equivalent to the following set of N independent l^{th} order systems:

$$\begin{cases} \hat{x}_i(k+1) = \mathbf{A}_i\hat{x}_i(k) + \mathbf{B}_{w,i}\hat{w}_i(k) + \mathbf{B}_{u,i}\hat{u}_i(k) \\ \hat{z}_i(k) = \mathbf{C}_{z,i}\hat{x}_i(k) + \mathbf{D}_{zw,i}\hat{w}_i(k) + \mathbf{D}_{zu,i}\hat{u}_i(k) \\ \hat{y}_i(k) = \mathbf{C}_{y,i}\hat{x}_i(k) + \mathbf{D}_{yw,i}\hat{w}_i(k) \end{cases} \quad (2.42)$$

for $i = 1, \dots, N$

where \hat{x}_i is the i^{th} block of size $l \times 1$ of \hat{x} , and \hat{w}_i , \hat{u}_i , \hat{z}_i and \hat{y}_i are similarly defined. We stress that these subsystems are different from the physical subsystems that may compose the global plant (i.e., the diagonal part of \mathbf{A}); for this reason, we will sometimes call them “modal subsystems” to emphasize this fact. \square

Also notice that according to (2.33), these systems can be written as:

$$\begin{cases} \hat{x}_i(k+1) = (A_a + \lambda_i A_b)\hat{x}_i(k) + (B_{w,a} + \lambda_i B_{w,b})\hat{w}_i(k) + (B_{u,a} + \lambda_i B_{u,b})\hat{u}_i(k) \\ \hat{z}_i(k) = (C_{z,a} + \lambda_i C_{z,b})\hat{x}_i(k) + (D_{zw,a} + \lambda_i D_{zw,b})\hat{w}_i(k) + (D_{zu,a} + \lambda_i D_{zu,b})\hat{u}_i(k) \\ \hat{y}_i(k) = (C_{y,a} + \lambda_i C_{y,b})\hat{x}_i(k) + (D_{yw,a} + \lambda_i D_{yw,b})\hat{w}_i(k) \end{cases} \quad (2.43)$$

for $i = 1, \dots, N$

This property means that for this class of systems many problems, like analysis or control design and even identification in some cases, can be approached by looking at the decomposed problem. This idea of decomposing the problem into a set of smaller problems is at the base of the control design methods shown in Chapter 3, and the identification methods of Chapter 6. We now highlight an additional observation that will be of use.

2.5.2 System norms

At the beginning of this section, we stated that the use of a diagonalization instead of a Schur transformation allows making considerations also on the performance of the system. If we call T_{wz} the transfer function of the system from the disturbance w to the performance output z , we can show that the norms of such transfer function is related to the norm of the transfer function in the transformed variables \hat{z} and \hat{w} . It is important to notice that the decomposing transformation in (2.40) is made of both a similarity transformation *and* a transformation of inputs and outputs; while the former does not affect system norms, the latter might do it. For this reason we state the following lemma.

Lemma 2.4 *Let T_{wz} be the transfer function of a decomposable system (Definition 2.5) from disturbance w to output z ; let $\hat{T}_{\hat{w}\hat{z}}$ be the transfer function of the same system after transforming them with (2.40), from the new disturbance \hat{w} to the new output \hat{z} . Then it holds (for both discrete and continuous time systems):*

$$\frac{1}{\kappa(S)} \|\hat{T}_{\hat{w}\hat{z}}\|_{\mathcal{H}_2} \leq \|T_{wz}\|_{\mathcal{H}_2} \leq \kappa(S) \|\hat{T}_{\hat{w}\hat{z}}\|_{\mathcal{H}_2} \quad (2.44)$$

$$\frac{1}{\kappa(S)} \|\hat{T}_{\hat{w}\hat{z}}\|_{\mathcal{H}_\infty} \leq \|T_{wz}\|_{\mathcal{H}_\infty} \leq \kappa(S) \|\hat{T}_{\hat{w}\hat{z}}\|_{\mathcal{H}_\infty} \quad (2.45)$$

where $\kappa(S)$ is the condition number of S . Moreover, let us call $\hat{T}_{\hat{w}_i\hat{z}_i}$ the transfer functions of each of the N modal subsystems into which the system can be decomposed, from \hat{w}_i to \hat{z}_i . Then:

$$\|\hat{T}_{\hat{w}\hat{z}}\|_{\mathcal{H}_2}^2 = \sum_{i=1}^N \|\hat{T}_{\hat{w}_i\hat{z}_i}\|_{\mathcal{H}_2}^2 \quad (2.46)$$

$$\|\hat{T}_{\hat{w}\hat{z}}\|_{\mathcal{H}_\infty} = \max_i \|\hat{T}_{\hat{w}_i\hat{z}_i}\|_{\mathcal{H}_\infty} \quad (2.47)$$

Proof: These expressions can be obtained from the definitions of the \mathcal{H}_2 and \mathcal{H}_∞ norms [29]. \square

Remark 2.2 *If we have a symmetric decomposable system, then S is orthogonal [37]: this means $\kappa(S) = 1$. So, according to (2.44) and (2.45), the system norms are the same for $\hat{T}_{\hat{w}\hat{z}}$ and T_{wz} .*

Remark 2.3 *In the remainder of the thesis we will almost only consider symmetric decomposable systems. These systems are easier to treat for two reasons: first, as seen in the last remark, because of the identity of the norms for the systems in the untransformed and in the transformed form; second: symmetric matrices have real eigenvalues and eigenvectors. But what will be shown in the next chapter can be easily generalized to any decomposable system; extra care will only be needed because of the presence of complex parameters, as well as the fact that the bounds we can impose to the norms have to be scaled by the factor $\kappa(S)$.*

2.5.3 The decomposition theorem

A question that comes natural by looking at Theorem 2.3 is whether such a decomposition property is possible for other kinds of systems as well. For example, we know from literature (see [13]) that any circulant system can be decomposed, but Theorem 2.3 in its form does not accommodate all the possible circulant systems (although \mathcal{P} can be circulant). Another question would be whether systems that have state-space matrices constructed with more than one pattern matrix, e.g.:

$$\mathcal{M} = I_N \otimes M_a + \mathcal{P}_1 \otimes M_{b_1} + \mathcal{P}_2 \otimes M_{b_2} \quad (2.48)$$

can still be object of decomposition.

We thus present a more general version of the decomposition theorem, that allows taking into account a wider set of possibilities; as we will show, a number of results in literature which make use of system decompositions can be given a unified point of view thanks to this theorem.

Theorem 2.4 (general decomposition theorem) *Consider a state-space system described by the equations in (2.36). Assume that all the state-space matrices (\mathcal{A} , \mathcal{B}_w , etc.) can be expressed as:*

$$\mathcal{M} = \sum_{k=1}^{\mu} \mathcal{P}_k \otimes M^{(k)} \quad (2.49)$$

where μ is an integer of choice and the matrices \mathcal{P}_k are simultaneously diagonalizable by a nonsingular matrix \mathcal{S} (i.e., $\mathcal{S}^{-1}\mathcal{P}_k\mathcal{S}$ are all diagonal for $k = 1, \dots, \mu$).

Then the system is equivalent to a set of N systems of equations as in (2.42), where all the state-space matrices (\mathbf{A}_i , $\mathbf{B}_{w,i}$, etc.) are of the form:

$$\mathbf{M}_i = \sum_{k=1}^{\mu} \lambda_i^{(k)} M^{(k)} \quad (2.50)$$

where the matrices $M^{(k)}$ are the same which appear in (2.49), and $\lambda_i^{(k)}$ is the i^{th} eigenvalue of \mathcal{P}_k (i.e., the i^{th} entry of the diagonal matrix $\mathcal{S}^{-1}\mathcal{P}_k\mathcal{S}$).

Conversely, a set of N systems as in (2.42) is equivalent to a system as in (2.36) with matrices as in (2.49), if all the matrices in the N systems are parameterized according to (2.50).

Proof: *The proof follows the same reasoning as the one for Theorem 2.3.* □

So it is possible to have a whole set of pattern matrices, as long as they are *simultaneously diagonalizable*. Matrices that commute in the multiplication and are diagonalizable are simultaneously diagonalizable [37], so a simple rule for having multiple patterns is to choose \mathcal{P}_k matrices which commute. Theorem 2.3 is the special case of this last theorem for $\mathcal{P}_1 = I_N$ and $\mathcal{P}_2 = \mathcal{P}$; as I_N commutes with any $N \times N$ matrix, then any diagonalizable \mathcal{P} is acceptable. An easy way to get multiple pattern matrices which commute is just to use the powers of the same

“base” pattern matrix \mathcal{P} , as all its powers ($\mathcal{P}^2, \mathcal{P}^3$, etc.) will commute with one another.

Following from this last reasoning, we can extend Definition 2.5 to accommodate a wider class of systems to which a decomposition applies.

Definition 2.6 (wide-sense decomposable systems) We call “wide-sense decomposable systems” the systems for which Theorem 2.4 applies (i.e., systems which have a state-space realization with matrices as in (2.49), for a certain set of simultaneously diagonalizable pattern matrices \mathcal{P}_k).

2.6 Special cases

As a conclusion to this chapter, we briefly introduce two very special cases of systems which have been described in literature, and for which the general decomposition theorem applies.

2.6.1 Circulant systems

Circulant systems [13] are systems that have a state-space realization made of block circulant matrices [17]. Such matrices fit the description of (2.49), as they can be expressed as:

$$\mathcal{M} = \sum_{k=1}^n \Pi_N^{k-1} \otimes M^{(k)} \quad (2.51)$$

where Π_N is the circulant shift permutation matrix:

$$\Pi_N = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} = \left[\begin{array}{c|c} 0 & I_{N-1} \\ \hline 1 & 0 \end{array} \right] \quad (2.52)$$

This permutation matrix and its powers are all diagonalizable and all commute with one another, so they can be simultaneously diagonalized by a complex matrix F_N . This matrix is well known in literature and it is called the Fourier matrix [17]. In this perspective, circulant systems are an example of wide-sense decomposable systems (Definition 2.6).

Circulant systems will be used later on to introduce the problem of identification of decomposable systems; more will be told about them in Section 6.2.1.

2.6.2 Symmetrically interconnected systems

Symmetrically interconnected systems [39] can be represented as having a state-space realization with matrices of the kind:

$$\mathcal{M} = I_N \otimes M_a + \mathcal{L}_c \otimes M_b \quad (2.53)$$

where \mathcal{L}_c is the (normalized) Laplacian of the complete graph (i.e. the graph with all the possible interconnections):

$$\mathcal{L}_c = \begin{bmatrix} 1 & -\frac{1}{N-1} & -\frac{1}{N-1} & \cdots & -\frac{1}{N-1} \\ -\frac{1}{N-1} & 1 & -\frac{1}{N-1} & \cdots & -\frac{1}{N-1} \\ -\frac{1}{N-1} & -\frac{1}{N-1} & 1 & \cdots & -\frac{1}{N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{N-1} & -\frac{1}{N-1} & -\frac{1}{N-1} & \cdots & 1 \end{bmatrix} \quad (2.54)$$

This kind of matrix is always diagonalizable (since it is circulant, [17]); however, it has only two distinct eigenvalues, 0 with algebraic multiplicity 1 and $\frac{N}{N-1}$ with multiplicity $N - 1$ [26]. This means that decomposing this system will result in N modal subsystems, $N - 1$ of which are all the same. So symmetrically interconnected systems are inherently described by only two subsystems, and this property is not a consequence of using a Laplacian matrix to represent the system, as it was already pointed out in [39].

Control of Decomposable Systems

In the previous chapter we have introduced the class of “decomposable systems”. The decomposition theorem has shown that these systems are equivalent to a set of smaller, parameter-dependent systems. This equivalence allows developing several suboptimal controller synthesis methods, which are shown in this chapter. The first method, that can be considered the main result of this thesis, is based on the application of a technique similar to the “multiobjective optimization” to this specific problem; this technique makes use of Linear Matrix Inequalities (LMIs) and can exploit an extended parameterization in the discrete time case. Other possible methods are based on robust control tools, like full block S-procedure or robust \mathcal{H}_∞ synthesis, but they prove to be more conservative and less flexible than the first approach. Several simple examples of applications are shown, together with comparisons among the methods.

3.1 Introduction

This chapter contains the main contribution of this thesis, namely a set of suboptimal distributed controller synthesis algorithms. In the previous chapter, we introduced a class of LTI systems, the so-called “decomposable systems”, that can be used to model a number of distributed or large scale systems made by the interconnection of identical elements, which typically have sparse state-space matrices. We have shown that we require the off-diagonal terms in the state-space matrices, which represent the interconnections, to be expressed as a Kronecker product involving a “pattern matrix”. If we use the language of graph theory, the subsystems can be considered as *nodes* of the graph and the interconnections as *edges*, and the pattern matrix can be considered a generalization of the graph adjacency or graph Laplacian matrices.

This kind of systems can be decomposed into a set of smaller subsystems, whose system norms are still related to the ones of the original system. This property can

be used in the first place to simplify the computations involving decomposable systems, by looking at them as an ensemble of smaller, independent systems. Actually the idea of decomposing a system for simplifying the controller synthesis is not new at all in the literature, and it has been applied for example for circulant systems [13], symmetrically interconnected systems [39; 87] and in SVD (Singular Value Decomposition) controllers and their generalizations [5; 38; 89]; in all these cases it is shown how the global control synthesis problem can be reduced to a collection of simpler problems through a transformation of input, outputs and state (typically, a MIMO system with N inputs and N outputs is reduced to a set of N SISO systems).

However, if we look at the problem of controlling such systems, what is desirable is to have the controller that is distributed as well. This means, it would be desirable to restrict the result of the controller synthesis only to controllers which are decomposable systems as well, with same pattern matrix as the plant. This is graphically shown in Figure 3.1. A controller of such structure, as previously

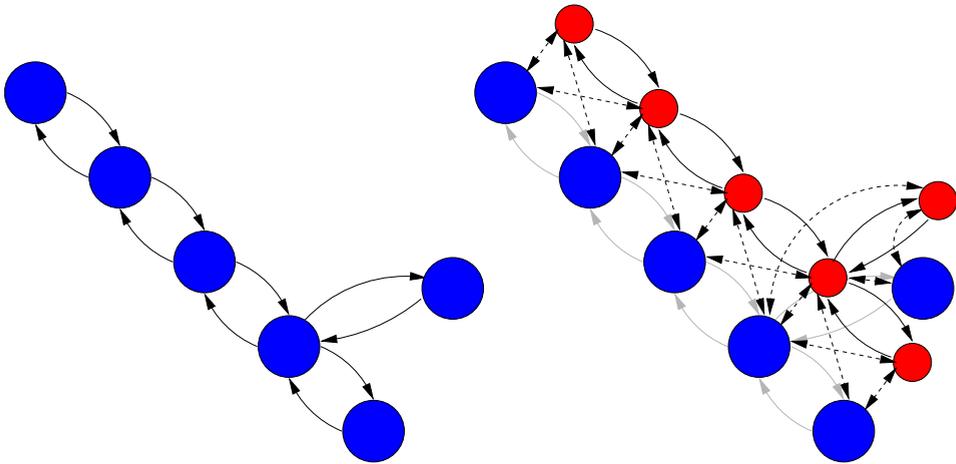


Figure 3.1: On the left, a system made of the interconnection of identical subsystems. On the right, a distributed controller made of a set of local controllers (the smaller circles): the spatial structure of the controller is the same as the one of the plant, and every local controller acts only on one subsystem and its nearest neighbors. The solid lines represent interactions between members of the same system, the dashed lines are input/output flow between plant and controller.

said, is preferable to a centralized controller for implementation reason, as it can be realized in practice as the interconnection of simple, local units acting on each single agent, without the need of a centralized supervisor with access to all output data. The earlier results mentioned above (e.g. [13; 38]) allow the use of the decomposition just for simplifying the computations, but they all yield as the final result a full centralized optimal controller. For example, SVD controllers are centralized controllers which can be seen as a set of simple SISO controllers after

a transformation of input and output. In the course of this chapter instead we will develop algorithms that make sure that the controller has the same sparsity as the system, allowing a distributed implementation of the controller. However, we will see that the distributed structure is achieved at the cost of suboptimality with respect to a global controller.

We are going to present several ways to obtain such kind of controllers. The first and most versatile method is based on the use an LMI-based technique that allows optimizing a controller for different output channels at the same time. This method is based on very intuitive ideas and it will prove very easily generalizable to other classes of problems and systems; this will be shown for example in Chapter 5, where we will introduce an extension to Linear Time-Periodic (LTP) systems. The other methods that we are going to develop are based on assuming the λ_i , which parameterizes the systems in the decomposed form, as an uncertainty. These methods will not prove as efficient as the first, but still they will prove of some interest.

This chapter is organized as follows. Section 3.2 introduces the main results of this thesis, the distributed control algorithms for decomposable systems based on the multiobjective optimization technique. The idea at the base of the method is shown first, and then all the relevant results are given explicitly and in a directly usable form; some cases of special interest are discusses as well. Section 3.3 contains two simple examples of application of this method; the first is a formation flying problem (such problems will be treated extensively in Chapter 5), whereas the second involves paper machines. Section 3.4 shows the two other possible approaches to controller synthesis based on other robust control techniques, and finally Section 3.5 contains the conclusions and some considerations on the value of the methods proposed here.

The results contained in this chapter have been published in [57; 59].

3.2 A decomposition approach to control

In the previous chapter, we have seen that for the class of decomposable systems, problems can be approached in the domain of the transformed variables, where the system is equivalent to a set of smaller independent modal subsystems. Once the solution has been obtained independently for each subsystem, one can retrieve the solution to the original problems through the inverse of the transformation shown in the proof of Theorem 2.3. Notice that the fact of working with a decomposed system does not imply that the final controller will be distributed or sparse; for this purpose, additional care will be needed, which we are going to show shortly.

3.2.1 Distributed control

As stated, we are going to design controllers for systems in the decomposed form, such that we may consider every modal subsystem independently. We consider

either static state-feedback controllers of the kind:

$$\hat{u}_i = K_i \hat{x}_i \quad \text{for } i = 1, \dots, N \quad (3.1)$$

or dynamic output-feedback controllers:

$$\begin{cases} \hat{x}_{c,i}(k+1) = A_{c,i} \hat{x}_{c,i}(k) + B_{c,i} \hat{y}_i(k) \\ \hat{u}_i(k) = C_{c,i} \hat{x}_{c,i}(k) + D_{c,i} \hat{y}_i(k) \end{cases} \quad \text{for } i = 1, \dots, N \quad (3.2)$$

(for the discrete time case). In general, these controllers will not have any special structure once they are rewritten in the domain of the variables “without the hat”:

$$u = \mathcal{K}x \quad (\text{static state feedback}) \quad (3.3)$$

or

$$\begin{cases} x_c(k+1) = \mathcal{A}_c x_c(k) + \mathcal{B}_c y(k) \\ u(k) = \mathcal{C}_c x_c(k) + \mathcal{D}_c y(k) \end{cases} \quad \begin{array}{l} \text{(dynamic} \\ \text{output feedback)} \end{array} \quad (3.4)$$

For example, it will be:

$$\mathcal{K} = (\mathcal{S} \otimes I_{m_u}) K (\mathcal{S} \otimes I_l)^{-1} \notin \mathcal{G}_{P, m_u, l}. \quad (3.5)$$

where K is the block diagonal matrix that contains the K_i 's in its diagonal. Assume now that the controller matrices are chosen such as they can be parameterized according to (2.33); remember that we denote such matrices with the bold font, so we will write $K = \mathbf{K}$, $A_c = \mathbf{A}_c$ etc. Then the matrices:

$$\begin{aligned} \mathcal{K} &= (\mathcal{S} \otimes I_{m_u}) \mathbf{K} (\mathcal{S} \otimes I_l)^{-1} \\ \mathcal{A}_c &= (\mathcal{S} \otimes I_l) \mathbf{A}_c (\mathcal{S} \otimes I_l)^{-1} \\ \mathcal{B}_c &= (\mathcal{S} \otimes I_l) \mathbf{B}_c (\mathcal{S} \otimes I_{r_y})^{-1} \\ \mathcal{C}_c &= (\mathcal{S} \otimes I_{m_u}) \mathbf{C}_c (\mathcal{S} \otimes I_l)^{-1} \\ \mathcal{D}_c &= (\mathcal{S} \otimes I_{m_u}) \mathbf{D}_c (\mathcal{S} \otimes I_{r_y})^{-1} \end{aligned} \quad (3.6)$$

which represent the possible controllers for the untransformed system, will have the same structure as the matrices of the system: thanks to Corollary 2.2, \mathcal{K} , \mathcal{A}_c , \mathcal{B}_c , \mathcal{C}_c , $\mathcal{D}_c \in \mathcal{G}_{P, \bullet, \bullet}$, where the bullets indicate consistent dimensions for each matrix. This means that the controller will have the same “physical” interconnection structure as the plant itself.

3.2.2 The basic idea

For example, let us now consider the problem of finding a stabilizing static state feedback (as in (3.3)) for a symmetric decomposable system (Definition 2.5). The basic LMI approach for solving the problem is to find a feasible solution to the following inequality [81]:

$$\begin{bmatrix} \mathcal{X} & \mathcal{A}\mathcal{X} + \mathcal{B}_u \mathcal{L} \\ * & \mathcal{X} \end{bmatrix} \succ 0 \quad (3.7)$$

where $\mathcal{X} = \mathcal{X}^T$ and \mathcal{L} are decision variables; $\mathcal{K} = \mathcal{L}\mathcal{X}^{-1}$. In the transformed domain, the LMI above is equivalent to the following set of smaller independent LMIs:

$$\begin{bmatrix} X_i & (A_a + \lambda_i A_b)X_i + (B_{u,a} + \lambda_i B_{u,b})L_i \\ * & X_i \end{bmatrix} \succ 0 \quad (3.8)$$

for $i = 1, \dots, N$

where now $X_i = X_i^T$ and L_i are decision variables. If we just solve each of the N LMIs independently, then there will be a gain $K_i = L_i X_i^{-1}$ for each subsystem; but if we stack all these gains in a block diagonal matrix K and perform the inverse transformation of (3.6) to get \mathcal{K} , then this \mathcal{K} in general will not be in the set $\mathcal{G}_{\mathcal{P},m_u,l}$.

If we instead want to have $\mathcal{K} \in \mathcal{G}_{\mathcal{P},m_u,l}$, we can solve the set of LMIs in (3.8) with the following coupling constraints:

$$\begin{aligned} X_i &= X \\ L_i = \mathbf{L}_i &= L_a + \lambda_i L_b \end{aligned} \quad \text{for } i = 1, \dots, N \quad (3.9)$$

and thus, the gains K_i will be “bold”, parameterized according to (2.34):

$$K_i = (L_a + \lambda_i L_b)X^{-1} = K_a + \lambda_i K_b = \mathbf{K}_i \quad (3.10)$$

yielding a $\mathcal{K} \in \mathcal{G}_{\mathcal{P},m_u,l}$. This approach is similar to what is done in the so-called LMI multiobjective optimization [80]; this method introduces some conservatism because we have set the same X matrix for all the LMIs. Since X is associated to the Lyapunov function of the closed loop system, this method is also called *Lyapunov shaping*.

In the literature a result has appeared that allows more generality to these multi-objective optimization problems in discrete time. In [18] it is shown that (3.7) can be replaced by the equivalent:

$$\begin{bmatrix} \mathcal{X} & \mathcal{A}\mathcal{G} + \mathcal{B}_u\mathcal{L} \\ * & \mathcal{G} + \mathcal{G}^T - \mathcal{X} \end{bmatrix} \succ 0 \quad (3.11)$$

where \mathcal{G} (not necessarily symmetric), \mathcal{X} and \mathcal{L} are the decision variables; $\mathcal{K} = \mathcal{L}\mathcal{G}^{-1}$. Then the equivalent of (3.8) is:

$$\begin{bmatrix} X_i & (A_a + \lambda_i A_b)G_i + (B_{u,a} + \lambda_i B_{u,b})L_i \\ * & G_i + G_i^T - X_i \end{bmatrix} \succ 0 \quad (3.12)$$

for $i = 1, \dots, N$

on which we can put the following constraints:

$$\begin{aligned} G_i &= G \\ L_i = \mathbf{L}_i &= L_a + \lambda_i L_b \end{aligned} \quad \text{for } i = 1, \dots, N \quad (3.13)$$

These constraints still introduce conservatism due to the single matrix G for all the LMIs, but leave a wider generality because no constraint is put on the Lyapunov

function (each LMI has its own X_i). This will lead to better results in the search for optimal values for the control problems that will be examined later on. In fact, the approach that has been used here for finding a stabilizing feedback can be extended to a wider range of problems, as it will be shown later on in this chapter.

Remark 3.1 *There are some cases when the set of N LMIs in (3.12) coupled by (3.13) are actually equivalent to just two LMIs. For example, if $B_{u,b} = 0$ (that is, if B_u is block diagonal) then all the LMIs can be expressed as a convex combination of the two LMIs that contain the extreme (maximum and minimum) values of λ_i . Then the feasibility of just these two inequalities will automatically grant the feasibility of all the others.*

Remark 3.2 *Let us evaluate the reduction in complexity of the problem, by going from its general formulation of (3.11) to the approach proposed here (equation (3.12) together with (3.13)). As the computational time involved in solving LMIs depends on the specific solver, we limit ourselves to finding the order of magnitude of the decision variables involved and the number of constraints. In (3.11) the number of decision variables is of the order of the biggest decision matrix involved, that is \mathcal{X} , with $\mathcal{X} \in \mathbb{R}^{Nl \times Nl}$; so the decision variables are of dimension $\mathcal{O}(N^2l^2)$. The number of constraints (the size of the LMI) is $\mathcal{O}(N^2l^2)$ as well. For (3.12) and (3.13), the biggest decision variables are the $X_i \in \mathbb{R}^{l \times l}$, which appear N times; so the decision variables are $\mathcal{O}(Nl^2)$. The constraints are N LMIs of the order of $l \times l$, so they are $\mathcal{O}(Nl^2)$ as well in number. The reduction of complexity is then of the order of the number N .*

Moreover, if Remark 3.1 holds, then the number X_i variables becomes only two as well as the number of LMIs: so we can claim that then the complexity is only $\mathcal{O}(l^2)$, another factor N less. We have to stress that in general we can consider $N \gg l$ for distributed systems.

These reductions of complexity will hold as well for all the problems that are discussed in the following pages.

3.2.3 Controller synthesis methods

The method shown in Section 3.2.2 to stabilize a system can be generalized and used to find suboptimal controllers with respect to the system norms. We say “suboptimal” as we can only provide sufficient and not necessary conditions.

Controller synthesis: discrete time, static state feedback

We first look for static state-feedback controllers which have the same structure as the system; they can be expressed by (3.3), with:

$$\mathcal{K} = I_N \otimes K_a + \mathcal{P} \otimes K_b. \quad (3.14)$$

We adapt the results of [18] to the class of systems considered here, for which we can state the two following theorems, the proofs of which are trivial once the content of Section 3.2.2 is understood.

Theorem 3.1 (\mathcal{H}_∞ state feedback, discrete time) Consider a discrete-time symmetric decomposable system according to Definition 2.5. A sufficient condition for the existence of a static state-feedback controller described by (3.1) and (3.14) that yields a $\|T_{wz}\|_{\mathcal{H}_\infty} < \gamma$ is that the following set of LMIs has a feasible solution:

$$\begin{bmatrix} P_i & \mathbf{A}_i X + \mathbf{B}_{u,i} \mathbf{L}_i & \mathbf{B}_{w,i} & 0 \\ * & X + X^T - P_i & 0 & X^T \mathbf{C}_{z,i}^T + \mathbf{L}_i^T \mathbf{D}_{zu,i}^T \\ * & * & I_{m_w} & \mathbf{D}_{zw,i}^T \\ * & * & * & \gamma^2 I_{r_z} \end{bmatrix} \succ 0 \quad (3.15)$$

for $i = 1, \dots, N$

where $P_i = P_i^T$, X , $\mathbf{L}_i = L_a + \lambda_i L_b$ are optimization variables, $K_a = L_a X^{-1}$, $K_b = L_b X^{-1}$.

Notice again that, according to Remark 2.1, the matrices of index i that are parameterized according to (2.34) have been written in bold, like for example \mathbf{L}_i , \mathbf{A}_i , $\mathbf{B}_{u,i}$ in the last equation.

Theorem 3.2 (\mathcal{H}_2 state feedback, discrete time) Consider a discrete-time symmetric decomposable system according to Definition 2.5. A sufficient condition for the existence of a static state-feedback controller described by (3.1) and (3.14) that yields a $\|T_{wz}\|_{\mathcal{H}_2} < \gamma$ is that the following set of LMIs has a feasible solution:

$$\begin{bmatrix} W_i & \mathbf{C}_{z,i} X + \mathbf{D}_{zu,i} \mathbf{L}_i \\ * & X + X^T - P_i \end{bmatrix} \succ 0,$$

$$\begin{bmatrix} P_i & \mathbf{A}_i X + \mathbf{B}_{u,i} \mathbf{L}_i & \mathbf{B}_{w,i} \\ * & X + X^T - P_i & 0 \\ * & * & I_{m_w} \end{bmatrix} \succ 0 \quad (3.16)$$

for $i = 1, \dots, N$, and

$$\sum_{i=1}^N \text{trace}(W_i) < \gamma^2$$

where $P_i = P_i^T$, $W_i = W_i^T$, X , $\mathbf{L}_i = L_a + \lambda_i L_b$ are optimization variables, $K_a = L_a X^{-1}$, $K_b = L_b X^{-1}$.

Remark 3.3 As for Remark 3.1, there are situations when the sets of LMIs in Theorem 3.1 can be reduced by considering only those of index i for which λ_i assumes the maximum and minimum values. This happens if $B_{u,b} = 0, D_{zu,b} = 0$. This also applies to Theorem 3.2, if the additional constraint of W_i being “bold” as well ($W_i = \mathbf{W}_i = W_a + \lambda_i W_b$) is applied (this will be worked out in detail in Chapter 4).

As in standard multiobjective optimization, more than one system norm can be constrained at the same time. For example, let us assume we have a symmetric decomposable system of the kind:

$$\begin{cases} x(k+1) = (\mathcal{A} + \Delta)x(k) + \mathcal{B}_w w(k) + \mathcal{B}_u u(k) \\ y(k) = \mathcal{C}x(k) + \mathcal{D}_w w(k) + \mathcal{D}_u u(k) \end{cases} \quad (3.17)$$

where Δ represents a static time-invariant uncertainty in the system. Let us assume that Δ depends on only one scalar parameter δ , such as:

$$\Delta = I_N \otimes (V_a \delta W_a) + \mathcal{P} \otimes (V_b \delta W_b) = \Delta_0 \delta \quad (3.18)$$

where V_a, V_b are column matrices and W_a, W_b are row matrices. Then for Δ as well we have that:

$$(\mathcal{S}^{-1} \otimes I_l) \Delta (\mathcal{S} \otimes I_l) = \mathbf{\Delta} \quad (3.19)$$

where $\mathbf{\Delta}$ is block diagonal. If the uncertainty is only in the diagonal (so $V_b, W_b = 0$) or only in the interconnections ($V_a, W_a = 0$), then for each block it will hold:

$$\mathbf{\Delta}_i = \delta \mathbf{V}_i \mathbf{W}_i \quad (3.20)$$

where again \mathbf{V}_i is a column and \mathbf{W}_i is a row. Assuming that $|\delta| < \delta_{\max}$, we can use the multiobjective synthesis method to design a controller with \mathcal{H}_∞ performance with robust stability, by solving the following optimization problem:

minimize γ^2 over:

$$\begin{aligned} & \begin{bmatrix} P_i & \mathbf{A}_i X + \mathbf{B}_{u,i} \mathbf{L}_i & \mathbf{B}_{w,i} & 0 \\ * & X + X^T - P_i & 0 & X^T \mathbf{C}_i^T + \mathbf{L}_i^T \mathbf{D}_{u,i}^T \\ * & * & I_{m_w} & \mathbf{D}_{w,i}^T \\ * & * & * & \gamma^2 I_{r_z} \end{bmatrix} \succ 0 \\ & \begin{bmatrix} Q_i & \mathbf{A}_i X + \mathbf{B}_{u,i} \mathbf{L}_i & \mathbf{W}_i & 0 \\ * & X + X^T - Q_i & 0 & X^T \mathbf{V}_i^T \\ * & * & 1 & 0 \\ * & * & * & \frac{1}{\delta_{\max}^2} \end{bmatrix} \succ 0 \end{aligned} \quad (3.21)$$

for $i = 1, \dots, N$

where $P_i = P_i^T, Q_i = Q_i^T, X$ and $\mathbf{L}_i = L_a + \lambda_i L_b$ are optimization variables. The first LMI sets the performance, while the second imposes that the \mathcal{H}_∞ norm of the transfer function from the output of the uncertainty (which has been “pulled out”) to its input is smaller than 1. This grants robust stability as a consequence of

the small gain theorem [95].

Notice that conservatism is introduced because the X in the first and second LMI are not necessarily the same, but they have to be chosen so in order to make the problem solvable. Notice also that since the robustness criterion is based on the small gain theorem, the system in closed loop will be robustly stable for any uncertainty Δ' that is elementwise smaller in modulus than $\Delta_{\max} = \Delta_0 \delta_{\max}$, even if it does not have the same structure.

Controller synthesis: discrete time, dynamic output feedback

The method can be used also for dynamic output feedback. Let us first start by reporting the general result as in [18], for \mathcal{H}_∞ . Let us assume we have a generic system of the kind:

$$\begin{cases} x(k+1) = Ax(k) + B_w w(k) + B_u u(k) \\ z(k) = C_z x(k) + D_{zw} w(k) + D_{zu} u(k) \\ y(k) = C_y x(k) + D_{yw} w(k) \end{cases} \quad (3.22)$$

for which we want to create a controller:

$$\begin{cases} x_c(k+1) = A_c x_c(k) + B_c y(k) \\ u(k) = C_c x_c(k) + D_c y(k) \end{cases} \quad (3.23)$$

that minimizes the \mathcal{H}_∞ norm from w to z . Such controllers can be found by minimizing γ^2 over an LMI constraint (not shown here for brevity), where $P = P^T$, $H = H^T$, X, L, Y, F, Q, R, S, J are decision variables. The controller matrices are then found with the relations:

$$\begin{cases} D_c = R \\ C_c = (L - RC_y X) U^{-1} \\ B_c = V^{-1} (F - Y B_u R) \\ A_c = V^{-1} (Q - Y (A + B_u D_c C_y) X - V B_c C_y X) U^{-1} - V^{-1} Y B_u C_c \end{cases} \quad (3.24)$$

where U and V are two arbitrary non-singular matrices such that $VU = S - YX$.

If the system is decomposable, we can again evaluate the N independent modal subsystems resulting from the decomposition of the global system, and solve the related N independent LMIs, each one with its own decision variables:

$$\left[\begin{array}{ccccccc} P_i & J_i & \mathbf{A}_i X_i + \mathbf{B}_{u,i} L_i & \mathbf{A}_i + \mathbf{B}_{u,i} \mathbf{R}_i \mathbf{C}_{y,i} & \mathbf{B}_{w,i} + \mathbf{B}_{u,i} \mathbf{R}_i \mathbf{D}_{yw,i} & & 0 \\ * & H_i & \mathbf{Q}_i & Y_i \mathbf{A}_i + \mathbf{F}_i \mathbf{C}_{y,i} & Y_i \mathbf{B}_{w,i} + \mathbf{F}_i \mathbf{D}_{yw,i} & & 0 \\ * & * & X_i + X_i^T - P_i & I_i + S_i^T - J_i & 0 & & X_i^T C_{z,i}^T + L_i^T D_{zu,i}^T \\ * & * & * & Y_i + Y_i^T - H_i & 0 & & C_{z,i}^T + C_{y,i}^T \mathbf{R}_i^T D_{zu,i}^T \\ * & * & * & * & I_{m_w} & & D_{zw,i}^T + D_{yw,i}^T \mathbf{R}^T D_{zu,i}^T \\ * & * & * & * & * & & \gamma^2 I_{r_z} \end{array} \right] \succ 0$$

for $i = 1, \dots, N$

(3.25)

And these are the LMIs for the \mathcal{H}_2 case:

$$\begin{bmatrix} W_i & \mathbf{C}_{z,i}X_i + \mathbf{D}_{zu,i}\mathbf{L}_i & \mathbf{C}_{z,i} + \mathbf{D}_{zu,i}\mathbf{R}_i\mathbf{C}_{y,i} \\ * & X_i + X_i^T - P_i & I_l + S_i^T - J_i \\ * & * & Y_i + Y_i^T - H_i \end{bmatrix} \succ 0,$$

$$\begin{bmatrix} P_i & J_i & \mathbf{A}_iX_i + \mathbf{B}_{u,i}\mathbf{L}_i & \mathbf{A}_i + \mathbf{B}_{u,i}\mathbf{R}_i\mathbf{C}_{y,i} & \mathbf{B}_{w,i} + \mathbf{B}_{u,i}\mathbf{R}_i\mathbf{D}_{yw,i} \\ * & H_i & \mathbf{Q}_i & Y_i\mathbf{A}_i + \mathbf{F}_i\mathbf{C}_{y,i} & Y_i\mathbf{B}_{w,i} + \mathbf{F}_i\mathbf{D}_{yw,i} \\ * & * & X_i + X_i^T - P_i & I_l + S_i^T - J_i & 0 \\ * & * & * & Y_i + Y_i^T - H_i & 0 \\ * & * & * & * & I_{m_w} \end{bmatrix} \succ 0,$$

$$\mathbf{D}_{zw,i} + \mathbf{D}_{zu,i}\mathbf{R}_i\mathbf{D}_{yw,i} = 0$$

for $i = 1, \dots, N$, and:

$$\sum_{i=1}^N \text{trace}(W_i) < \gamma^2. \quad (3.26)$$

It is possible to see that, under certain assumptions which we will shortly show, there is a parameterization of the decision variables such that $A_c, B_c, C_c, D_c \in \mathcal{B}_{\mathcal{P}, \bullet, \bullet}$, yielding a controller in the untransformed domain that will be of the same structure as the plant. The parameterization is the following:

$$\begin{cases} X_i = X \\ Y_i = Y \\ S_i = S \\ \mathbf{L}_i = L_a + \lambda_i L_b \\ \mathbf{F}_i = F_a + \lambda_i F_b \\ \mathbf{Q}_i = Q_a + \lambda_i Q_b \\ \mathbf{R}_i = R_a + \lambda_i R_b \\ P_i = P_i^T, H_i = H_i^T, J_i \text{ unconstrained} \end{cases} \quad \text{for } i = 1, \dots, N \quad (3.27)$$

that together with:

$$\begin{cases} VU = S - YX \quad (\text{with } U, V \text{ non singular}) \\ \mathbf{D}_{c,i} = \mathbf{R}_i \\ \mathbf{C}_{c,i} = (\mathbf{L}_i - \mathbf{R}_i\mathbf{C}_{y,i}X)U^{-1} \\ \mathbf{B}_{c,i} = V^{-1}(\mathbf{F}_i - Y\mathbf{B}_{u,i}\mathbf{R}_i) \\ \mathbf{A}_{c,i} = V^{-1}(\mathbf{Q}_i - Y(\mathbf{A}_i + \mathbf{B}_{u,i}\mathbf{R}_i\mathbf{C}_{y,i})X - V\mathbf{B}_{c,i}\mathbf{C}_{y,i}X)U^{-1} - V^{-1}Y\mathbf{B}_{u,i}\mathbf{C}_{c,i} \end{cases} \quad (3.28)$$

and (3.6) will yield $\mathcal{A}_c, \mathcal{B}_c, \mathcal{C}_c, \mathcal{D}_c \in \mathcal{G}_{\mathcal{P}, \bullet, \bullet}$; we only have to assume that whenever we have a product of more than one bold matrix, then all the bold matrices involved but one must be constant over the index i . For example, we have that:

$$\mathbf{C}_{c,i} = (\mathbf{L}_i - \mathbf{R}_i\mathbf{C}_{y,i}X)U^{-1} = \quad (3.29)$$

$$= (L_{c,a} + \lambda_i L_{c,b} - (R_a + \lambda_i R_b)(C_{y,a} + \lambda_i C_{y,b})X)U^{-1} \quad (3.30)$$

So if we want $C_{c,i}$ to be parameterized as $C_{c,i} = C_{c,a} + \lambda_i C_{c,b}$, then we either need to have $C_{y,i}$ constant ($C_{y,b} = 0$) or to set R_i as constant ($R_b = 0$). All the possible cases in which this holds, as well as the additional constraints which might be required, are listed in Table 3.1.

Table 3.1: Conditions and additional constraints for solving dynamic output feedback problems.

Case	Conditions	Additional constraints
1	$C_{y,b} = 0, B_{u,b} = 0$	none
2	$B_{u,b} = 0$	$R_b = 0, F_{u,b} = 0$
3	$C_{y,b} = 0$	$R_b = 0$

We summarize these results in the following theorems.

Theorem 3.3 (\mathcal{H}_∞ output feedback, discrete time) Consider a discrete-time symmetric decomposable system according to Definition 2.5, in one of the cases of Table 3.1. A sufficient condition for the existence of a decomposable dynamic output-feedback controller described by (3.23) that yields a $\|T_{wz}\|_{\mathcal{H}_\infty} < \gamma$ is that the set of LMI constraints in (3.25) has a feasible solution. The decision variables and their parameterization are shown in (3.27), while Table 3.1 shows the additional constraints which might be needed. The state-space matrices of the controller can be retrieved through (3.28).

Theorem 3.4 (\mathcal{H}_2 output feedback, discrete time) Consider a discrete-time symmetric decomposable system according to Definition 2.5, in one of the cases of Table 3.1. A sufficient condition for the existence of a decomposable dynamic output-feedback controller described by (3.23) that yields a $\|T_{wz}\|_{\mathcal{H}_2} < \gamma$ is that the set of LMI constraints in (3.26) has a feasible solution. The decision variables and their parameterization are the ones shown in (3.27), with the addition of $W_i = W_i^T$; Table 3.1 shows the additional constraints which might be needed. The state-space matrices of the controller can be retrieved through (3.28).

Remark 3.4 As for Remark 3.1, also for the LMIs in Theorem 3.3 and Theorem 3.4 a reduction can be done in the case of $B_{u,b} = 0, D_{zu,b} = 0, D_{yw,b} = 0$ (and $W_i = \mathbf{W}_i = W_a + \lambda_i W_b$ for Theorem 3.4).

Controller synthesis: continuous time, static state feedback

As shown in the previous pages for discrete-time systems, for continuous-time systems as well it is possible to use the same kind of reasoning and get to suboptimal controller synthesis methods. The main difference with the discrete time case is that in continuous time there is, to our knowledge, no extended LMI parameterization like in [18] that we can easily use for our scope. We summarize the results for continuous time in two theorems, only for the static state-feedback case.

Theorem 3.5 (\mathcal{H}_∞ state feedback, continuous time) Consider a continuous-time symmetric decomposable system according to Definition 2.5. A sufficient condition for the existence of a static state-feedback controller described by (3.1) and (3.14) that yields a $\|T_{wz}\|_{\mathcal{H}_\infty} < \gamma$ is that the following set of LMIs has a feasible solution:

$$\begin{bmatrix} \mathbf{A}_i X + \mathbf{B}_{u,i} \mathbf{L}_i + X^T \mathbf{A}_i^T + \mathbf{L}_i^T \mathbf{B}_{u,i}^T & \mathbf{B}_{w,i} & X \mathbf{C}_{z,i}^T + \mathbf{L}_i^T \mathbf{D}_{zu,i}^T \\ * & -I_{m_w} & \mathbf{D}_{zw,i}^T \\ * & * & -\gamma^2 I_{r_z} \end{bmatrix} \succ 0 \quad (3.31)$$

for $i = 1, \dots, N$

where $P_i = P_i^T$, X , $\mathbf{L}_i = L_a + \lambda_i L_b$ are optimization variables, $K_a = L_a X^{-1}$, $K_b = L_b X^{-1}$.

Theorem 3.6 (\mathcal{H}_2 state feedback, continuous time) Consider a continuous-time symmetric decomposable system according to Definition 2.5. A sufficient condition for the existence of a static state-feedback controller described by (3.1) and (3.14) that yields a $\|T_{wz}\|_{\mathcal{H}_2} < \gamma$ is that the following set of LMIs has a feasible solution:

$$\begin{bmatrix} X & X \mathbf{C}_{z,i}^T + \mathbf{L}_i^T \mathbf{D}_{zu,i}^T \\ * & W_i \end{bmatrix} \succ 0, \quad (3.32)$$

$$\begin{bmatrix} \mathbf{A}_i X + \mathbf{B}_{u,i} \mathbf{L}_i + X^T \mathbf{A}_i^T + \mathbf{L}_i^T \mathbf{B}_{u,i}^T & \mathbf{B}_{w,i} \\ * & -I_{m_w} \end{bmatrix} \succ 0$$

for $i = 1, \dots, N$, and

$$\sum_{i=1}^N \text{trace}(W_i) < \gamma^2$$

where $P_i = P_i^T$, $W_i = W_i^T$, X , $\mathbf{L}_i = L_a + \lambda_i L_b$ are optimization variables, $K_a = L_a X^{-1}$, $K_b = L_b X^{-1}$.

Notice that Remark 3.3 holds for these last two theorems as well.

3.2.4 Links with graph theory

In Chapter 2 we mentioned graph theory as a way of interpreting decomposable systems. Actually, graph theory can be also of further use in this situation, as it can give guidelines in the choice of the pattern matrices. In fact, by looking at (2.30) it is apparent that the same matrix \mathcal{M} can be obtained with different \mathcal{P} 's, by adjusting M_a and M_b . The pattern matrix for a system is not unique; however, it can be convenient to choose a \mathcal{P} with guaranteed bounds on its eigenvalues. Following the same reasoning as in [26], then it can be convenient to choose it as a normalized graph Laplacian matrix. For such kind of matrices it may be actually not necessary to compute the eigenvalues, as we already know a boundary for them; this means that in the cases when Remark 3.3 or 3.4 holds, we can assume

directly 0 as minimum eigenvalue and 2 as the maximum one. Secondly, as these boundaries hold for all normalized Laplacians, then the result of this controller synthesis will be valid *regardless of the number of interconnected elements N and regardless of the topology*, as long as the pattern is a normalized Laplacian matrix. These statements will be made clearer by the first example in Section 3.3.

3.2.5 Special cases

Multiple patterns

The LMI procedures shown in Section 3.2.3 can be naturally extended to accommodate wide-sense decomposable systems, with any number of patterns (Definition 2.6); simply by redefining “bold” decision variables as in (2.50), then formally the same LMIs can be used to search for controllers. Remarks 3.1, 3.3 and 3.4 will not hold anymore in their present form, although it can be possible to reduce the number of LMIs only to those which contain the $\lambda_i^{(k)}$ that are at the vertices of the convex hull of all the possible μ -tuples of them. This last sentence will be made clearer in the first example of Section 3.3, where we will briefly show the synthesis of a controller with multiple pattern matrices.

Symmetrically interconnected systems

Optimal controllers for symmetrically interconnected systems are inherently symmetrically interconnected too, for this reason, the synthesis method shown earlier is not necessary for this kind of systems. In Section 3.2.3 we have shown how the introduction of constraints will force the controller to be of the same kind as the plant; these constraints are not needed in this case. Let us show an example to clarify this. Consider the problem of finding a state feedback; the decomposition of the problem will lead to just two subsystems, thus two LMIs (or two Riccati equations) need to be solved for the optimal control problem. We do not introduce the constraints, so we will get two independent state gains for the two subsystems, let us call them K_1 and $K_{2\dots N}$. Whatever these gains are, it will always be possible to parameterize them as:

$$\begin{aligned} K_1 &= K_a \\ K_{2\dots N} &= K_a + \frac{N}{N-1} K_b \end{aligned} \tag{3.33}$$

This means that no conservatism needs to be added in order to get a symmetrically interconnected controller. This explains why controllers of this kind, which have been obtained through decompositions, are always optimal (sometimes they are called *superoptimal* in literature [39]).

Circulant systems

Circulant systems have inherently circulant optimal controllers as well, so in order to find a full circulant controller for a circulant system there is no need of introducing the constraints; in fact in literature control design methods for circulant

systems based on decomposition have already been explored [13; 19]. However, it can be useful to use the method shown in this section for control design of systems which have a limited bandwidth in the circulant matrix, i.e., with matrices stemming from a limited number of permutation matrices:

$$\mathcal{M} = \sum_{k=-b}^b \Pi_N^k \otimes M^{(k)} \quad (3.34)$$

where b is the bandwidth. The earlier methods would yield a full circulant controller, while the method of this thesis would keep the bandwidth limited thanks to the introduction of constraints, at the cost of suboptimality.

3.3 Examples of applications

In this section we present three simple examples of application of the proposed controller synthesis method, which are meant to give a first flavor of what can be done with them. More in-depth examples of applications are given in Chapter 4 and Chapter 5.

3.3.1 Formation flying of satellites in a circular orbit

As a first example, we present a problem where the physical subsystems are dynamically disconnected, but a cross-coupling between them is introduced by the performance index. A formation flying problem is a typical example of this situation. Let us consider a swarm of satellites orbiting around a planet on a circular orbit (a similar example is shown in [25]). The small perturbations of their motion with respect to the nominal circular trajectory are described by the so-called Clohessy-Wiltshire (CW) equations [72]:

$$\begin{cases} \ddot{x}_1 = 3\omega_n^2 x_1 + 2\omega_n \dot{x}_2 + a_1 \\ \ddot{x}_2 = -2\omega_n \dot{x}_1 + a_2 \\ \ddot{x}_3 = -\omega_n^2 x_3 + a_3 \end{cases} \quad (3.35)$$

where x_1 , x_2 and x_3 are respectively the displacements in the radial, tangential and out-of-plane direction with respect to an ideal body covering perfectly the circular orbit at an angular speed ω_n ; a_1 , a_2 and a_3 are the accelerations of the spacecraft due to either propulsion or external disturbances.

Let us now assume that N satellites are uniformly distributed on the same circular orbit, and that we would like to design a controller that minimizes the error on their relative positions, with an \mathcal{H}_∞ criterion. We are going to compare three different controllers: 1) a centralized controller that considers the formation as a whole; 2) a decentralized controller that acts on every satellite on its own, minimizing the norm from its local input to its local output; 3) a distributed controller

made of the interaction of local controllers which can communicate with the nearest neighbors. Figure 3.2 visually shows the difference between these types of controllers.

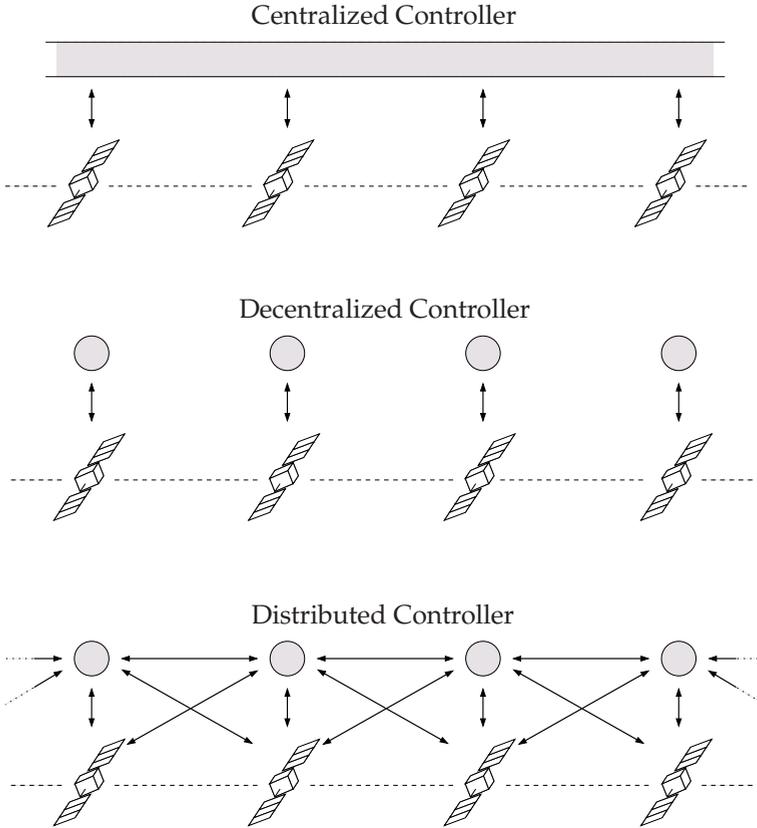


Figure 3.2: Three different types of controllers. The arrows represent information flow.

The last of the three controllers can be designed with the method shown in this chapter. We said that the goal is designing a controller for minimizing the errors on the relative positions; as there is no dynamic interaction between the satellites, the cross-coupling between them will be introduced by the performance output. So if we consider the set of satellites as a single system, all the matrices will be block diagonal but C_z . Since we need to put a penalty on the difference of the position, we can choose as performance output the following:

$$z_{x_\nu, i} = -\frac{1}{2}x_{\nu, i-1} + x_{\nu, i} - \frac{1}{2}x_{\nu, i+1} \quad \text{for } \nu = 1, 2, 3 \quad (3.36)$$

In this way, the performance output matrix C_z will be block symmetric, and it will be possible to express it by means of a symmetric pattern matrix. This would not happen if we choose something like $x_{\nu, i} - x_{\nu, i+1}$ as output, which might seem a

more natural choice.

Once the output has been decided, the pattern matrix has to be chosen. As stated in Section 3.2.4, normalized Laplacian matrices are a better choice, and in this case it is possible to use a symmetric normalized Laplacian matrix which we call $\mathcal{L}^{(1)}$:

$$\mathcal{L}^{(1)} = \begin{bmatrix} 1 & -\frac{1}{2} & 0 & \cdots & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 1 & -\frac{1}{2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -\frac{1}{2} & 0 & 0 & \cdots & -\frac{1}{2} & 1 \end{bmatrix} \quad (3.37)$$

With this, if we call C_z the output matrix of a single satellite, then the global performance output matrix will be $\mathcal{C}_z = \mathcal{L}^{(1)} \otimes C_z$. Notice that $\mathcal{L}^{(1)}$ is circulant as well.

In addition, we consider a non-zero D_{zu} matrix in order to penalize the use of the actuators (the consumption of propellant), for which reason we can add the following three performance outputs:

$$z_{a_\nu, i} = w a_\nu \quad \text{for } \nu = 1, 2, 3 \quad (3.38)$$

where w is a weighting parameter.

We turned the problem into a discrete-time problem with a Tustin approximation [27]: then we can use Theorem 3.3, and as we are in the case of Remark 3.4, the output-feedback synthesis problem can be solved with a reduced set of only two LMIs. Moreover, as $\mathcal{L}^{(1)}$ is a normalized Laplacian, it is possible to execute the computation only once for all the formations of any size, by assuming as maximum and minimum eigenvalues 0 and 2 respectively.

This controller is of course suboptimal, but it will be distributed and it will require only communications between nearest neighbors: the i^{th} satellite will communicate with the satellites of index $i + 1$ and $i - 1$. The performance of the controller can be increased by allowing one more communication link, that means, allowing the i^{th} satellite to communicate with those of of index $i + 2$ and $i - 2$. This can be done by introducing a second pattern matrix as in Section 2.5.3:

$$\mathcal{L}^{(2)} = \begin{bmatrix} 1 & 0 & -\frac{1}{2} & 0 & \cdots & 0 & -\frac{1}{2} & 0 \\ 0 & 1 & 0 & -\frac{1}{2} & \cdots & 0 & 0 & -\frac{1}{2} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & & \\ 0 & -\frac{1}{2} & 0 & 0 & \cdots & -\frac{1}{2} & 0 & 1 \end{bmatrix} \quad (3.39)$$

This is again a symmetric normalized Laplacian, and it is a valid choice for a second pattern matrix¹ as it is circulant too, and all circulant matrices commute [17]. However, as explained in Section 2.5.3, Remark 3.4 cannot be applied in its form anymore, as the eigenvalues $\lambda_i^{(1)}$ and $\lambda_i^{(2)}$ of $\mathcal{L}^{(1)}$ and $\mathcal{L}^{(2)}$ respectively must not be considered on their own but as couples $(\lambda_i^{(1)}, \lambda_i^{(2)})$. The set of LMIs can be

¹Actually, it is a third pattern matrix, as I_N can be considered a pattern as well.

reduced to only those that generate the convex hull of all the $(\lambda_i^{(1)}, \lambda_i^{(2)})$, but as it can be seen from Figure 3.3, no reduction is possible in this case and all the N LMIs have to be considered.

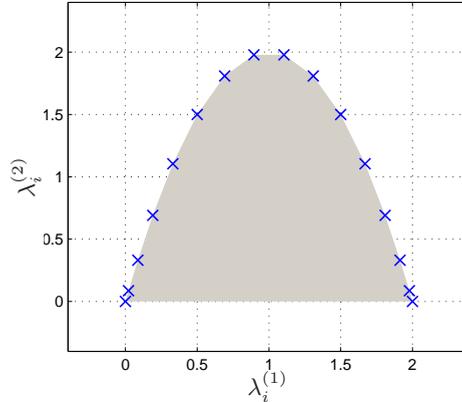


Figure 3.3: Eigenvalues of $\mathcal{L}^{(1)}$ versus those of $\mathcal{L}^{(2)}$ ($N = 30$). The shaded area is the convex hull of these points: the vertices of the convex hull are all the points themselves in this case.

The computations for the synthesis of the two distributed controllers have been executed with Matlab, using SeDuMi [86] as solver, with the help of the user-friendly interface provided by the Yalmip toolbox [52]. Also an optimal centralized controller and a decentralized one have been computed for comparison. The results of the computations, for different numbers of satellites, are shown in Figure 3.4. As it was expected, the centralized optimal control offers the best performance, while the decentralized has the worst ones. The distributed controllers are in between and quite close to the global optimum, with the 2-pattern one performing slightly better.

Figure 3.5 shows the sparsity of the four different controllers of Figure 3.4.

3.3.2 Formation flying in deep space

We now consider again a formation flying problem, where the physical subsystems are dynamically disconnected and the cross-coupling is introduced by the performance index, as in the previous example. We consider a formation in deep space, whose dynamics is basically Newton's second law:

$$\ddot{x} = a \tag{3.40}$$

which we consider in the discretized version; x is the position in the tridimensional space and a is the acceleration of the spacecraft due to either propulsion or disturbances. We use this example to show the effects of choosing a directed graph

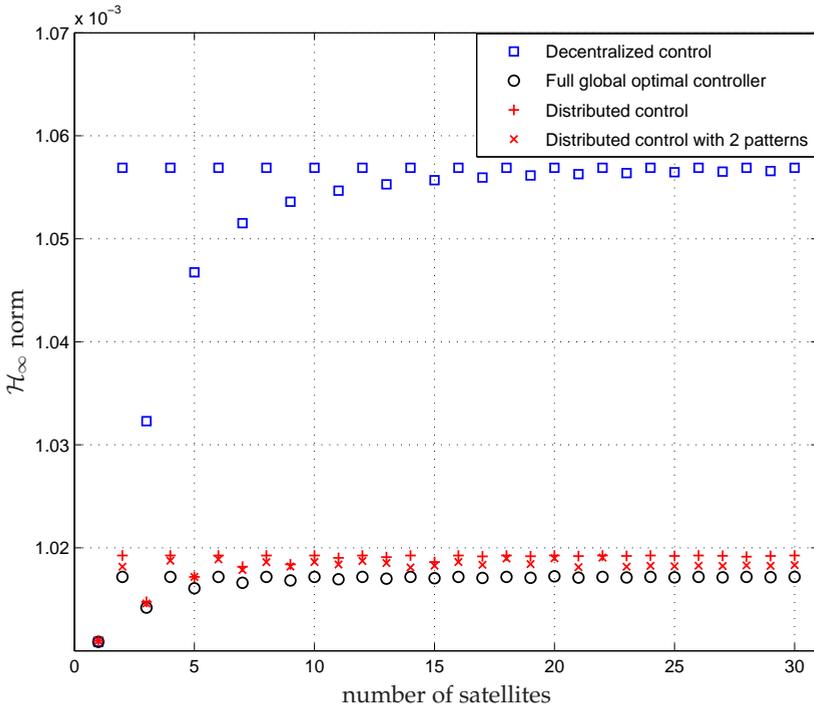


Figure 3.4: \mathcal{H}_∞ norm with the 4 different controllers, for different numbers N of satellites.

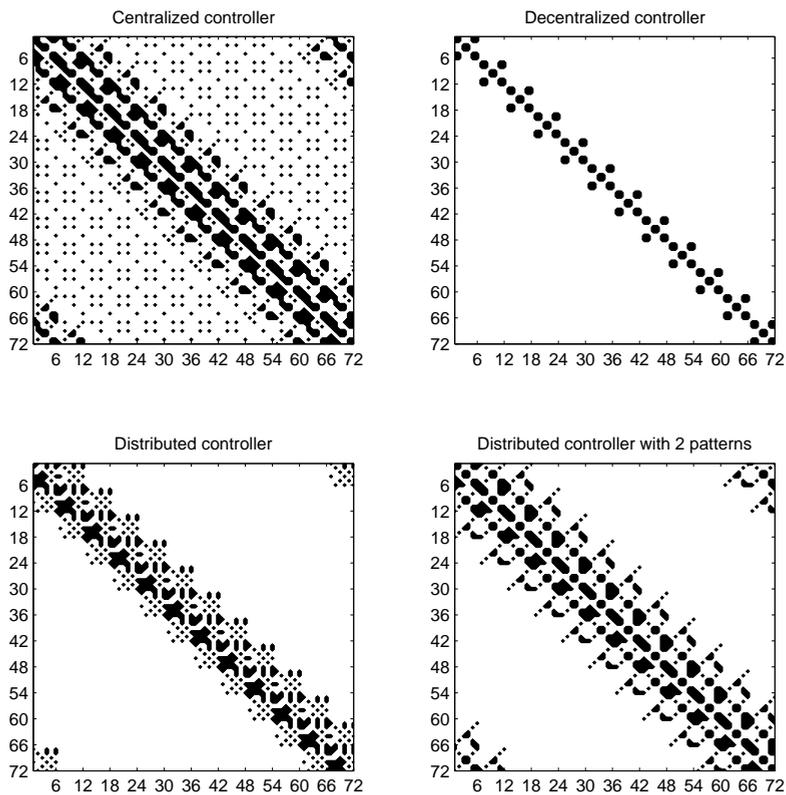


Figure 3.5: Sparsity of the state matrix of the 4 different controllers, for $N = 12$. A black dot indicates a non-zero entry.

or a non-symmetric pattern matrix in the synthesis method, whereas in the rest of the thesis only symmetric decomposable systems are considered (as explained in Remark 2.3).

We consider a formation of 4 spacecraft associated to the directed graph shown in Figure 3.6.

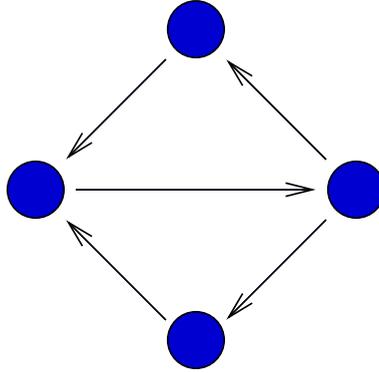


Figure 3.6: Directed graph for the deep space formation.

The pattern matrix that we choose is going to determine the cross-couplings in the performance output, exactly as in the previous example. We are going to execute the distributed \mathcal{H}_∞ synthesis for this formation for three different patterns, and then we will compare the results.

The first pattern that we will use is the following:

$$\mathcal{P}_1 = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ -\frac{1}{2} & 0 & 1 & -\frac{1}{2} \\ 0 & -1 & 0 & 1 \end{bmatrix} \quad (3.41)$$

which is the normalized Laplacian of the directed graph in Figure 3.6. This matrix has complex eigenvalues, and it is decomposed by a non unitary S matrix. This implies that the LMIs used for the distributed synthesis, namely those of Theorem 3.3, will feature complex values of λ_i (and the transpose operator needs to be replaced by the conjugate transpose or Hermitian). We also have to constrain the decision variables to be real in order to have a real controller (notice that λ_i can be factorized in (3.28), and will not influence the $A_{c,a}$, $A_{c,b}$, etc. matrices). Moreover, as explained in Lemma 2.4, the bound γ on the \mathcal{H}_∞ of the closed-loop system needs to be rescaled by the condition number of S , making the true bound on the norm equal to $\gamma\kappa(S)$.

The second pattern matrix that we consider is:

$$\mathcal{P}_2 = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} & 0 \\ -\frac{1}{3} & 1 & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & 1 & -\frac{1}{3} \\ 0 & -\frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix} \quad (3.42)$$

which is the normalized Laplacian of the undirected graph associated to the one in Figure 3.6. This matrix is not symmetric but it has real eigenvalues, so Remark 3.4 applies. Still the bounds on the norm need to be scaled by the condition number.

Finally, we will consider:

$$\mathcal{P}_3 = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 \\ -\frac{1}{3} & 1 & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & 1 & -\frac{1}{3} \\ 0 & -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} \end{bmatrix} \quad (3.43)$$

which is the weighted normalized Laplacian of the undirected graph, obtained by adding arbitrary self loops on the first and last node (as shown in Example 2.4). This matrix is symmetric.

The results with the three different matrices are shown in Table 3.2, together with the comparison with a centralized controller.

Table 3.2: Controller performance for the deep space formation example.

Case	centralized controller	distributed controller			
	\mathcal{H}_∞ norm	\mathcal{H}_∞ norm	γ	$\kappa(S)$	$\gamma\kappa(S)$
\mathcal{P}_1	28.63	34.79	28.00	1.41	39.60
\mathcal{P}_2	24.02	25.47	22.69	1.22	27.68
\mathcal{P}_3	22.52	22.52	22.52	1	22.52

The first thing to notice in the evaluation of the results, is that the three cases represent different problems: changing the pattern matrix changes the optimization criterion (namely, the performance index), so choosing a different representation of the graph links is a relevant part of the problem formulation, and not only a different point of view on the same problem. This is also clear from the fact that the centralized controller yields different performance in the three cases. We can see that in the first case (directed graph) the difference in performance between distributed controller and centralized controller is significant. For the undirected graph case (\mathcal{P}_2 and \mathcal{P}_3), we see that using a symmetric Laplacian yields a distributed controller that has almost the same performance as the centralized one. This does not happen if we use the non-symmetric pattern matrix. In the examples of Chapter 5 we will always use symmetric weighted Laplacians as pattern matrices of undirected graphs, in order to always have $\kappa(S) = 1$.

3.3.3 Paper machines

Another example of a system that can be analyzed with the same methods comes from the cross-directional control of paper machines [85]. In such devices, the wet paper pulp is distributed on a conveyor belt and then forced through a gap in order to create an extrusion (the paper sheet). A good quality paper should have constant properties (e.g. thickness, weight per unit area), so it is necessary to have an array of actuators that compensates for the irregularities in the distribution of the paper pulp.

Under the point of view of system theory, a paper machine can be modeled as a discrete-time system with N inputs (the actuator commands) and N outputs (the error in the paper thickness). If we denote such inputs as $u(k) \in \mathbb{R}^N$ and the outputs as $y(k) \in \mathbb{R}^N$, then a model for the system can be:

$$y(k) = \mathcal{P}u(k-d) + ay(k-1) + w(k) \quad (3.44)$$

where a is a scalar (representing a stable pole), d is a delay, w is a disturbance and \mathcal{P} is a band matrix that accounts for the cross-coupling between the actuators. Actually, \mathcal{P} is the only source of cross-coupling between the physical subsystems, which interact with as many neighbors as the number of off-diagonal bands in \mathcal{P} . Some references (like [89], with some adaptations) assume \mathcal{P} as a generic band matrix, while others (like [85]) take it as a band symmetric Toeplitz matrix.

Many different methods have been used in literature for dealing with the problem; the most common approaches are either to approximate the Toeplitz matrix with a circulant matrix [49; 85], or to use a centralized optimal SVD controller which uses the decomposition to simplify the complexity of the computations [89]. With our approach, we will find a suboptimal controller that can be implemented as distributed; Figure 3.7 shows the difference between the structure of this controller and of an SVD one.

The first thing that we need to do is to turn the model of the machine into a state-space formulation as in (2.36), with the result that \mathcal{P} becomes the pattern matrix. We prefer working with a symmetric \mathcal{P} , so we chose the numerical model from [85]; more generic models could still be handled, but they would require the use of LMIs with complex values, as explained in Remark 2.3.

The state-space formulation is also non-unique, in fact the off-diagonal terms can be put either in \mathcal{B}_u or in \mathcal{A} ; of course we prefer the second option, as we will have fewer restrictions in the synthesis problem (see Table 3.1), and we will also be able to use the reduced method (Remark 3.4).

This time we have used \mathcal{H}_2 synthesis for different values of N . The results are shown in Table 3.3. Figure 3.8 gives an idea of the structure of the systems, showing the sparsity pattern of the state matrix of the plant and the controller.

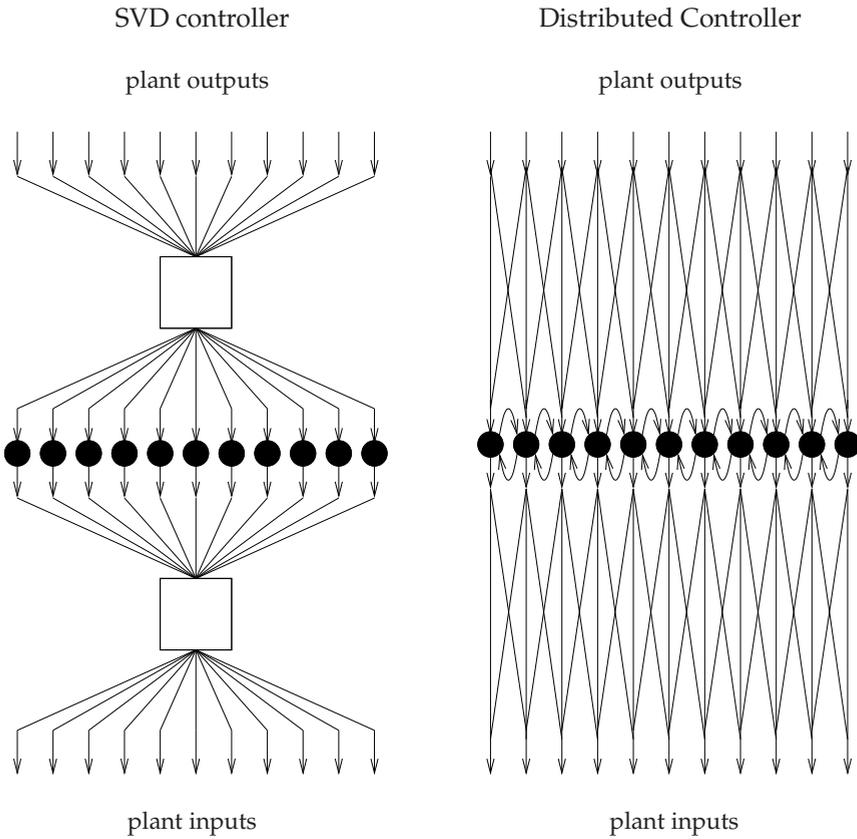


Figure 3.7: Difference between an SVD controller and a distributed controller. The boxes represent static transformations, while the circles are dynamic controllers. The SVD controller still needs to handle all the inputs and outputs in the same processing units, so it can be considered as a statically centralized controller, with decoupled dynamics.

Table 3.3: Controller performance for the paper machine example.

Number of actuators	\mathcal{H}_2 norm:	
	No Control	Distributed Control
50	0.14587	0.09786
100	0.20629	0.13893
150	0.25265	0.17035
200	0.29173	0.19625

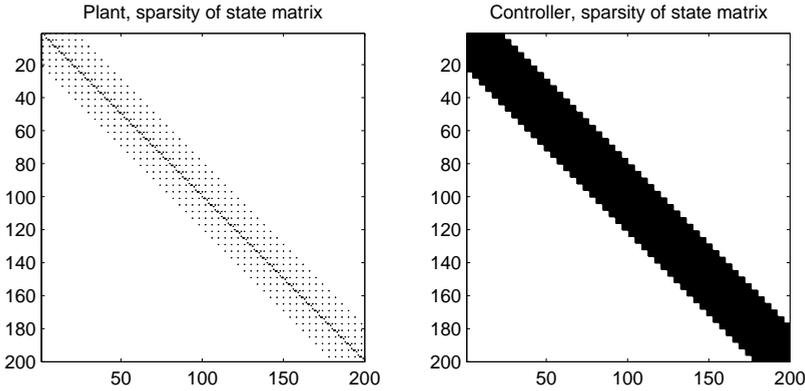


Figure 3.8: Sparsity of the state matrix of the plant (A) and of the controller, for $N = 50$. A black dot indicates a non-zero entry.

3.4 Variations on the theme

The controller synthesis methods shown in Section 3.2 are one possibility of exploiting the decomposition property of decomposable systems in order to obtain distributed controllers, but not the only one. Actually, we can exploit the similarities between a decomposed system and an LPV system, and use different methods from robust control in order to perform the synthesis. In this section we show possible examples of state-feedback \mathcal{H}_∞ control that still brings some conservatism with it. Future research should be focused on looking at whether it can be possible to find a method that is not conservative, and would lead to an “if and only if” kind of result.

3.4.1 The full block S-procedure

We start by introducing the theoretical tool that we are going to use; it consists of a convex solution to the general robust state feedback synthesis problem that is obtained by means of the so-called “full block S-procedure”.

Theorem 3.7 (full block S-procedure) (adapted from [81]) *Let us consider an uncertain continuous-time system described by the equations:*

$$\begin{cases} \dot{x}(t) = Ax(t) + B_1p(t) + B_2w(t) + Bu(t) \\ q(t) = C_1x(t) + D_1p(t) + D_{12}w(t) + E_1u(t) \\ z(t) = C_2x(t) + D_{21}p(t) + D_2w(t) + E_2u(t) \end{cases} \quad (3.45)$$

where $x(t)$ is the state, $w(t)$ the disturbance, $u(t)$ the control input, $z(t)$ the performance output, $q(t)$ and $p(t)$ signals for which it holds:

$$p(t) = \Delta(t)q(t) \quad (3.46)$$

where $\Delta(t)$ is a (time-varying) uncertainty assuming values in the convex hull generated by a set $\{\Delta_j\}$ for $j = 1, \dots, J$, which includes the zero vector.

The state feedback controller yielding an \mathcal{H}_∞ norm from w to z smaller than γ for all the valid uncertainties can be found via the following LMI set:

$$Y \succ 0, \quad \tilde{R} \prec 0 \quad (3.47)$$

$$* \begin{bmatrix} \tilde{Q} & \tilde{S} \\ \tilde{S}^T & \tilde{R} \end{bmatrix} \begin{bmatrix} I \\ -\Delta_j^T \end{bmatrix} \prec 0 \text{ for all } j = 1, \dots, J \quad (3.48)$$

$$* \begin{array}{c|ccc|ccc} \begin{matrix} 0 & I & 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & \tilde{Q} & \tilde{S} & 0 & 0 \\ 0 & 0 & \tilde{S}^T & \tilde{R} & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \frac{-1}{\gamma^2} I & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{matrix} & \begin{matrix} -(AY+BM)^T & -(C_1Y+E_1M)^T & -(C_2Y+E_2M)^T \\ I & 0 & 0 \\ \hline -B_1^T & -D_1^T & -D_{12}^T \\ 0 & I & 0 \\ \hline -B_2^T & -D_{21}^T & -D_2^T \\ 0 & 0 & I \end{matrix} & \succ 0 \end{array} \quad (3.49)$$

where $Y = Y^T$, $M, \tilde{R} = \tilde{R}^T$, $\tilde{S}, \tilde{Q} = \tilde{Q}^T$ are the decision variables, and the stars * replace the symbols that would make the left-hand side of the inequalities symmetric. The controller gain K is given by $K = MY^{-1}$.

Notice that this theorem offers only a sufficient condition, so its use implies conservatism. Also notice that the theorem involves a time-varying uncertainty, which means that some more conservatism will be introduced when applying it to the decentralized or distributed control methods, as in that case the uncertainties (the eigenvalues of the pattern matrix) are time-invariant. Alas the S-procedure, in the form presented here, offers a solution only for the state-feedback case, so we will not be able to use this result for dynamic output-feedback controller synthesis, as we did in the previous section.

3.4.2 Full block S-procedure and decentralized control

We consider a subclass of continuous-time decomposable systems, where only the state matrix is built with the pattern and the other state-space matrices are diagonal. We assume a symmetric pattern matrix \mathcal{P} , with real eigenvalues. The methods in Section 3.2 would find a suboptimal \mathcal{H}_∞ controller with state gain as in (3.14), with a diagonal part and a part with the sparsity described by the pattern matrix. We call such controllers “distributed controllers”, meaning that the control action on each agent is determined on the basis of the local output and the output of the neighboring agents. It is possible to constrain the controller to have $K_b = 0$, which would yield a decentralized controller. By “decentralized controller” we mean a controller made of a set of local controllers that interact only with each single agent, and not with their neighbors; for this reason, such controllers have a block diagonal gain matrix. Such controllers can be obtained with the full block S-procedure, and the result is summarized in the following theorem.

Theorem 3.8 (\mathcal{H}_∞ S-procedure decentralized control) Let us consider a continuous-time symmetric decomposable systems described by the equations:

$$\begin{cases} \dot{x}(t) = (I_N \otimes A_a + \mathcal{P} \otimes A_b)x(t) + (I_N \otimes B_u)u(t) + (I_N \otimes B_w)w(t) \\ z(t) = (I_N \otimes C)x(t) + (I_N \otimes D)u(t) \end{cases} \quad (3.50)$$

where $x(t)$ is the state, $w(t)$ the disturbance, $u(t)$ the control input and $z(t)$ the performance output. There exists a suboptimal decentralized controller of the kind:

$$u(t) = (I_N \otimes K)x(t) \quad (3.51)$$

with \mathcal{H}_∞ norm from w to z smaller than γ , if the following LMIs are feasible:

$$Y \succ 0, \quad \tilde{R} \prec 0, \quad \underline{\lambda} \leq 0 \leq \bar{\lambda} \quad (3.52)$$

$$* \begin{bmatrix} \tilde{Q} & \tilde{S} \\ \tilde{S}^T & \tilde{R} \end{bmatrix} \begin{bmatrix} I \\ -\bar{\lambda}I \end{bmatrix} \prec 0, \quad * \begin{bmatrix} \tilde{Q} & \tilde{S} \\ \tilde{S}^T & \tilde{R} \end{bmatrix} \begin{bmatrix} I \\ -\underline{\lambda}I \end{bmatrix} \prec 0 \quad (3.53)$$

$$* \left[\begin{array}{cc|cc|cc} 0 & I & 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & \tilde{Q} & \tilde{S} & 0 & 0 \\ 0 & 0 & \tilde{S}^T & \tilde{R} & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \frac{-1}{\gamma^2}I & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{array} \right] \begin{bmatrix} -(A_a Y + B_u M)^T & -(A_b Y)^T & -(C Y + D M)^T \\ I & 0 & 0 \\ \hline -I & 0 & 0 \\ 0 & I & 0 \\ \hline -B_w^T & 0 & 0 \\ 0 & 0 & I \end{bmatrix} \succ 0 \quad (3.54)$$

where $\bar{\lambda}$ and $\underline{\lambda}$ are the maximum and minimum eigenvalue of \mathcal{P} , and $Y = Y^T$, M , $\tilde{R} = \tilde{R}^T$, $\tilde{S}, \tilde{Q} = \tilde{Q}^T$ are the decision variables. The controller gain K is given by $K = M Y^{-1}$.

3.4.3 Full block S-procedure and distributed control

With the new method it is not possible to introduce directly a K_b term as in (3.51) that would produce a distributed controller instead of a decentralized one. There is a “trick” though that allows using the decentralized controller synthesis method of Theorem 3.8 to generate a distributed controller. It is just sufficient to extend (3.50) by adding another input channel that is influenced by the pattern matrix. In fact, we can prove that:

$$\begin{cases} \dot{x}(t) = (I_N \otimes A_a + \mathcal{P} \otimes A_b)x(t) + (I_N \otimes B_u)u(t) + (I_N \otimes B_w)w(t) \\ z(t) = (I_N \otimes C_{z,a} + \mathcal{P} \otimes C_{z,b})x(t) + (I_N \otimes D_{zu})u(t) \end{cases} \quad (3.55)$$

in closed loop with:

$$u(t) = (I_N \otimes K_a + \mathcal{P} \otimes K_b)x(t) \quad (3.56)$$

is equivalent to:

$$\begin{cases} \dot{x}(t) = (I_N \otimes A_a + \mathcal{P} \otimes A_b)x(t) + (I_N \otimes [B_u \ 0] + \mathcal{P} \otimes [0 \ B_u])\tilde{u}(t) + (I_N \otimes B_w)w(t) \\ z(x) = (I_N \otimes C_{z,a} + \mathcal{P} \otimes C_{z,b})x(t) + (I_N \otimes [D_{zu} \ 0] + \mathcal{P} \otimes [0 \ D_{zu}])\tilde{u}(t) \end{cases} \quad (3.57)$$

in closed loop with:

$$\tilde{u}(t) = I_N \otimes \begin{bmatrix} K_a \\ K_b \end{bmatrix} x(t) \quad (3.58)$$

They both yield:

$$\begin{cases} \dot{x}(t) = (I_N \otimes (A_a + B_u K_a) + \mathcal{P} \otimes (A_b + B_u K_b))x(t) + (I_N \otimes B_w)w(t) \\ z(x) = (I_N \otimes (C_{z,a} + D_{zu} K_a) + \mathcal{P} \otimes (C_{z,b} + D_{zu} K_b))x(t) \end{cases} \quad (3.59)$$

The system in (3.57) in closed loop with the decentralized controller (3.58) is equivalent to the controller in (3.55) in closed loop with the distributed controller in (3.56). So one can use (3.57) and (3.58) as a reference for computing the distributed controller of the system in (3.55). Again we summarize this in a theorem.

Theorem 3.9 (\mathcal{H}_∞ S-procedure distributed control) *Let us consider a continuous-time symmetric decomposable systems described by the equations:*

$$\begin{cases} \dot{x}(t) = (I_N \otimes A_a + \mathcal{P} \otimes A_b)x(t) + (I_N \otimes B_u)u(t) + (I_N \otimes B_w)w(t) \\ z(x) = (I_N \otimes C_{z,a} + \mathcal{P} \otimes C_{z,b})x(t) + (I_N \otimes D_{zu})u(t) \end{cases} \quad (3.60)$$

where $x(t) \in \mathbb{R}^l$ is the state, $w(t) \in \mathbb{R}^{m_w}$ the disturbance, $u(t) \in \mathbb{R}^{m_u}$ the control input and $z(t) \in \mathbb{R}^{r_z}$ the performance output. There exists a suboptimal distributed controller of the kind:

$$u(t) = (I_N \otimes K_a + \mathcal{P} \otimes K_b)x(t) \quad (3.61)$$

with \mathcal{H}_∞ norm from w to z smaller than γ , if the following LMIs are feasible:

$$Y \succ 0, \quad \tilde{R} \prec 0, \quad \Delta \leq 0 \leq \bar{\Delta} \quad (3.62)$$

$$* \begin{bmatrix} \tilde{Q} & \tilde{S} \\ \tilde{S}^T & \tilde{R} \end{bmatrix} \begin{bmatrix} I \\ -\bar{\Delta}I \end{bmatrix} \prec 0, \quad * \begin{bmatrix} \tilde{Q} & \tilde{S} \\ \tilde{S}^T & \tilde{R} \end{bmatrix} \begin{bmatrix} I \\ -\Delta I \end{bmatrix} \prec 0 \quad (3.63)$$

$$* \begin{bmatrix} 0 & I & 0 & 0 & 0 & 0 \\ I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \tilde{Q} & \tilde{S} & 0 & 0 \\ 0 & 0 & \tilde{S}^T & \tilde{R} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{-1}{\gamma^2}I & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{bmatrix} \begin{bmatrix} -(\mathcal{A}Y + \mathcal{B}M)^T & -(\mathcal{C}_1Y + \mathcal{E}_1M)^T & -(\mathcal{C}_2Y + \mathcal{E}_2M)^T \\ I & 0 & 0 \\ -\mathcal{B}_1^T & -\mathcal{D}_1^T & -\mathcal{D}_{21}^T \\ 0 & I & 0 \\ -\mathcal{B}_2^T & -\mathcal{D}_{12}^T & -\mathcal{D}_2^T \\ 0 & 0 & I \end{bmatrix} \succ 0 \quad (3.64)$$

where:

$$\begin{bmatrix} \mathcal{A} & \mathcal{B}_1 & \mathcal{B}_2 & \mathcal{B} \\ \mathcal{C}_1 & \mathcal{D}_1 & \mathcal{D}_{12} & \mathcal{E}_1 \\ \mathcal{C}_2 & \mathcal{D}_{21} & \mathcal{D}_2 & \mathcal{E}_2 \end{bmatrix} = \begin{bmatrix} A_a & A_b & B_u & B_w & B_u & 0 \\ I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I \\ C_a & C_b & D & 0 & D & 0 \end{bmatrix} \quad (3.65)$$

and $\bar{\lambda}, \underline{\lambda}$ are the maximum and minimum eigenvalue of \mathcal{P} , and $Y = Y^T$, $M, \bar{R} = \bar{R}^T, \bar{S}, \bar{Q} = \bar{Q}^T$ are the decision variables. The controller gains K_a, K_b , are given by the relation:

$$K = \begin{bmatrix} K_a \\ K_b \end{bmatrix} = MY^{-1} \quad (3.66)$$

Proof: Consider the equivalent system (3.57) in its decomposed form:

$$\begin{cases} \dot{x}_i(t) = (A_a + \lambda_i \otimes A_b)x_i(t) + ([B_u \ 0] + \lambda_i [0 \ B_u])\tilde{u}_i(t) + B_w w_i(t) \\ z_i(x) = (C_{z,a} + \lambda_i C_{z,b})x_i(t) + ([D_{zu} \ 0] + \lambda_i [0 \ D_{zu}])\tilde{u}_i(t) \end{cases} \quad (3.67)$$

with $\tilde{u}_i = [\tilde{u}_{a,i}^T \ \tilde{u}_{b,i}^T]^T$. This set of systems (depending on λ_i) can be interpreted as a system depending on the (uncertain) parameter λ_i . Define the output signal $q_i(t)$ as:

$$q_i(t) = [x_i^T \ \tilde{u}_{b,i}^T]^T \quad (3.68)$$

and the input signal $p_i(t)$ as:

$$p_i(t) = \lambda_i q_i(t) \quad (3.69)$$

Then the system can be rewritten as:

$$\begin{cases} \dot{x}_i(t) = A_a x_i(t) + [A_b \ B_u] p_i(t) + [B_u \ 0] \tilde{u}_i(t) + B_w w_i(t) \\ q_i(t) = \begin{bmatrix} I \\ 0 \end{bmatrix} x_i(t) + \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} \tilde{u}_i(t) \\ z_i(x) = C_{z,a} x_i(t) + [C_{z,b} \ D_{zu}] p_i(t) + [D_{zu} \ 0] \tilde{u}_i(t) \end{cases} \quad (3.70)$$

from which the matrices in (3.65) can be defined by inspection. Then apply Theorem 3.7. \square

A discrete-time version of Theorem 3.9 can be similarly derived (but it is not shown in this thesis).

Remark 3.5 Notice that the cases to which Theorem 3.9 applies are fewer with respect to the method in Section 3.2.3, because there is no possibility of having a part in the input matrix or in the feedthrough matrix depending on the pattern (Theorem 3.9 applies only if $B_{u,b} = 0, D_{zu,b} = 0$).

Remark 3.6 The technical condition that $\underline{\lambda} \leq 0 \leq \bar{\lambda}$ does not reduce the applicability of the theorem, as the parameterization in (2.34) is not unique. For example, in the case that $0 \leq \underline{\lambda} \leq \bar{\lambda}$, then pick an a such that $\underline{\lambda} \leq a \leq \bar{\lambda}$; choose a new pattern matrix $\mathcal{P}' = \mathcal{P} - aI_N$, and replace the state-space matrices M_a with $M'_a = M_a + aM_b$. The global state-space matrices are unchanged and the eigenvalues of \mathcal{P}' satisfy the condition.

3.4.4 Small gain theorem for distributed control

It is possible to use the small gain theorem [95] as an alternative to the full block S-procedure in order to guarantee the robust \mathcal{H}_∞ performance for all the possible values of the parameter λ , similarly to what is done in [68]. This would lead to the following theorem.

Theorem 3.10 (\mathcal{H}_∞ robust performance-based control) *Let us consider a continuous-time symmetric decomposable system described by (3.60), and put it in the form as in (3.45) via the relations in (3.65) considering λ_i as the perturbation. There exists a suboptimal distributed controller as in (3.61) that yields a closed-loop \mathcal{H}_∞ norm smaller or equal to γ if the \mathcal{H}_∞ norm of the closed loop system from the joint inputs $[w^T \ p^T]^T$ to the joint outputs $[z^T \ q^T]^T$ is $\leq \gamma$ and $\max_i(|\lambda_i|) < \frac{1}{\gamma}$.*

The synthesis can be executed by iterating an \mathcal{H}_∞ synthesis LMI on the value of γ , rescaling the disturbance signal p if $\max_i(|\lambda_i|)$ is too far from $\frac{1}{\gamma}$.

3.4.5 Evaluation of numerical results

It is natural to compare the method of Theorem 3.9 with the method of Theorem 3.5 and the method in Theorem 3.10. The three methods have been tried on randomly generated systems and compared. It appears that most of the times (more than 95% of the cases, in our tests) the method of Theorem 3.5 yields better performance. Yet there are some cases where the opposite happens; for example, it happens for this system:

$$\left\{ \begin{array}{l} \dot{x}(t) = \left(I_N \otimes \begin{bmatrix} 0.1 & -0.2 & 0.7 \\ -0.9 & -0.6 & -0.4 \\ -0.9 & 0.6 & -0.5 \end{bmatrix} + \mathcal{P} \otimes \begin{bmatrix} 0.1 & 0.1 & -0.1 \\ -0.3 & -0.1 & -0.1 \\ -0.2 & -0.1 & 0 \end{bmatrix} \right) x(t) + \\ \quad + I_N \otimes \begin{bmatrix} 0.7 \\ 0 \\ 0.5 \end{bmatrix} u(t) + I_N \otimes \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \end{bmatrix} w(t) \\ z(x) = \left(I_N \otimes \begin{bmatrix} 0.9 & 0.2 & 0.3 \\ 0 & 0 & 0 \end{bmatrix} + \mathcal{P} \otimes \begin{bmatrix} 0.5 & 0.3 & 0.5 \\ 0 & 0 & 0 \end{bmatrix} \right) x(t) + I_N \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \end{array} \right. \quad (3.71)$$

with pattern matrix shown in Figure 3.9. This pattern matrix is a (non normalized) adjacency matrix this time. The eigenvalues of the pattern matrix are in the interval $[-2.2361, 2.2361]$ and the local order is $l = 3$. The results of the different synthesis methods are shown in Table 3.4.

A more common situation would instead yield results as in Table 3.5, which shows the performance for a different system, whose matrices we do not report, again with $l = 3$ and the same pattern matrix. It can be seen that the best performance is given by the multiobjective optimization method, while the small gain is the most conservative of the three.

and a parameterization of the decision variables is introduced. Another approach to decomposable systems is to employ robust control techniques, like full block S-procedure or robust \mathcal{H}_∞ synthesis. We have shown that the complexity of the design problems in terms of number of decision variables and constraints is reduced by a factor N with respect to a centralized controller. In some cases, the reduction is of a factor N^2 , making actually the complexity independent from N .

In fact, Theorems 3.1, 3.2, 3.5, and 3.6 under the conditions of Remark 3.3, Theorems 3.3 and 3.4 under Remark 3.4, and finally Theorem 3.9, offer the possibility of executing performance-based controller synthesis for a system with an LMI test whose size is not dependent on the number N of subsystems. Under this perspective, the methods that we propose can then be thought of as extensions of the results of [26] (summarized in Example 2.5). One of the most interesting results of this reference is a variant of the Nyquist criterion that relates the stability of the formation to the number of encirclements of certain points in the Nyquist plot. As these points are related to the eigenvalues of a normalized Laplacian, it is then possible to define a region where they are restricted to be, and this makes it possible to find a controller that stabilizes all the possible formations of a kind with a single simple test. The synthesis methods shown here, when they are not depending on N , can be considered as an equivalent of this method for performance optimization, instead of simple stability: a single LMI test, which does not grow with the number of agents, can be enough to guarantee disturbance rejection performance for all the systems whose structure is described by normalized Laplacian of an undirected graph.

The following two chapters show two examples of application of the approach developed here in two different fields. Together with the practical applicability of the proposed methods, the examples will show that these method can be easily tailored to fit the specific requirements that a certain problem might require. Namely, in Chapter 4 we will employ pattern matrices with bounded spectra which are not graph Laplacians, and in Chapter 5 we will extend the \mathcal{H}_∞ synthesis methods to Linear Time Varying systems.

Adaptive Optics Applications

The results for distributed control design of decomposable systems can be applied in the field of adaptive optics, specifically to the problem of controlling the deformation of a mirror under the effect of cross-coupled actuators. We will consider pattern matrices which are not graph Laplacians, but we will be able to put a bound on their eigenvalues as well. In this way, the new technique allows the design of a distributed \mathcal{H}_2 controller for a mirror of any size or shape, with a computational cost that does not increase with the size of the mirror.

4.1 Introduction

In this chapter we will use the results on control of decomposable systems in order to show an application to a class of problems arising from adaptive optics, namely the control of the shape of a deformable mirror, whose actuators have a reciprocal cross-coupling. One of the results that we are going to see is that if the function describing the cross-coupling has a certain regularity (or “spatial invariance”), then it is possible to obtain controllers which are proved to work for mirrors of any size. This would allow the implementation of such controllers in a truly modular fashion, as local computational units connected to each other, which achieve their performance goals regardless of the size of the mirror. The proof is obtained thanks to considerations on the spectrum of the matrices describing the actuator cross-coupling; under this perspective, the approach is similar to what shown in Chapter 3 with respect to Laplacian matrices: once bounds on the spectrum of the pattern matrices are obtained, it is possible to find a controller holding for a whole category of systems, regardless of their number of elements.

The chapter is organized as follows. Section 4.2 introduces adaptive optics and the problem of the control of a deformable mirror. Section 4.3 explains how the results of Chapter 3 can be applied to the problem. Section 4.4 shows a reasoning that will allow deriving controllers for mirrors of any size, with a constant computational

cost. Numerical results are given in Section 4.5 and finally the conclusions are drawn in Section 4.6.

The results contained in this chapter have been published in [55].

4.2 Adaptive optics

Adaptive optics [36] is a technique that can be employed to improve the quality of imaging systems, e.g. telescopes. If a telescope is based on the Earth, the light collected at the primary mirror is affected by distortions in the wavefront due to atmospheric turbulence. These distortions can be measured with wavefront sensors and be compensated for by means of a deformable mirror. Deformable mirrors [34] are typically made of a thin membrane that can be shaped by means of an array of linear actuators (e.g. voice coils), arranged in a regular lattice (see Figure 4.1).

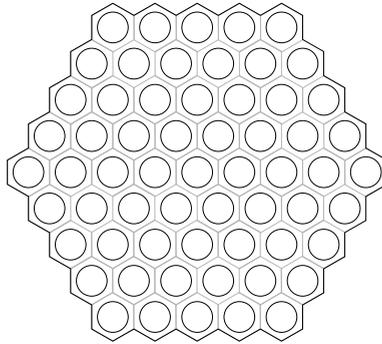


Figure 4.1: An adaptive optics mirror. Each circle contains an actuator.

If N is the number of actuators, let us call $x \in \mathbb{R}^N$ the vector of the commands (e.g. voltage) given to displace them. If we consider the control points for the mirror's shape to be co-located with the actuators, we can define the vector $y \in \mathbb{R}^N$ of the (normal) displacements of the mirror in correspondence with the actuators' positions. Basically, y describes a possible shape that the mirror can have. During the functioning of an adaptive optics system, the reference vector y is computed at each time step in order to compensate for the turbulence, and there is the necessity of computing the appropriate actuators commands x that can give the desired shape to the mirror y . The actuator dynamics is usually considered as very fast, such that the relation between actuator commands and mirror shape can be considered as static; alas things become complicated because of cross-coupling between neighboring actuators. The relation linking actuators and mirror shape is described by the following formula:

$$y = Gx \quad (4.1)$$

where $G \in \mathbb{R}^{N \times N}$ is the so called “influence matrix”, a sparse matrix that accounts for reciprocal interactions between neighboring elements.

We will assume that the displacement in the normal direction ζ induced by the action of a single actuator is described by a kind of Gaussian formula:

$$\zeta = ae^{-\frac{d^2}{2\sigma^2}} \quad (4.2)$$

where d is the distance from the center of the actuator, a and σ are parameters [75].

With this function, G is symmetric and so it will have real eigenvalues. For the method we present, the Gaussian shape is not essential; it will be only necessary and sufficient that the influence function is spatially invariant, which is approximately true for large homogeneous, continuous or segmented, deformable mirrors.

We consider hexagonal grids for the actuators (as in Figure 4.1) and we normalize the distance d such that the space between each neighboring actuator is 1. We can also introduce the parameter R indicating the size of the mirror, namely the length in terms of actuators of each side of the hexagonal mirror. With these kind of assumptions, Figure 4.2 shows the displacement of the mirror induced by one single actuator, for $\sigma = 0.7$. The influence function quickly decreases with d , and we consider it to be 0 for $d > 3\sigma$. Figure 4.3 shows the elements that are affected by the use of one single actuator.

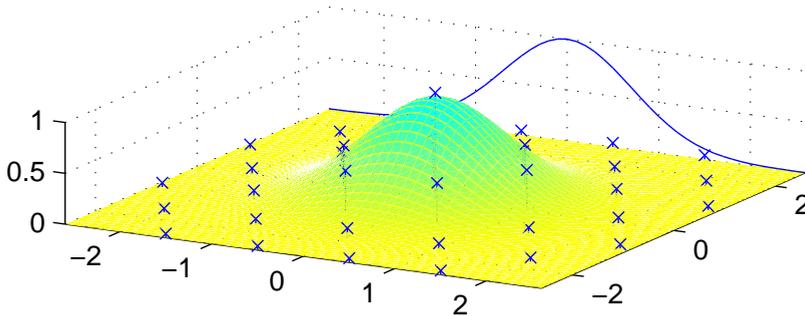


Figure 4.2: Deformation of the mirror caused by a single actuator.

4.3 Distributed control of decomposable systems

Decomposable systems represent systems that are the result of the interconnection of N identical subsystems of order l . The interconnection follows a pattern that is described by a “pattern matrix” \mathcal{P} ; a sparse pattern matrix accounts for a limited interaction among the subsystems, that for example can only influence a few neighboring elements.

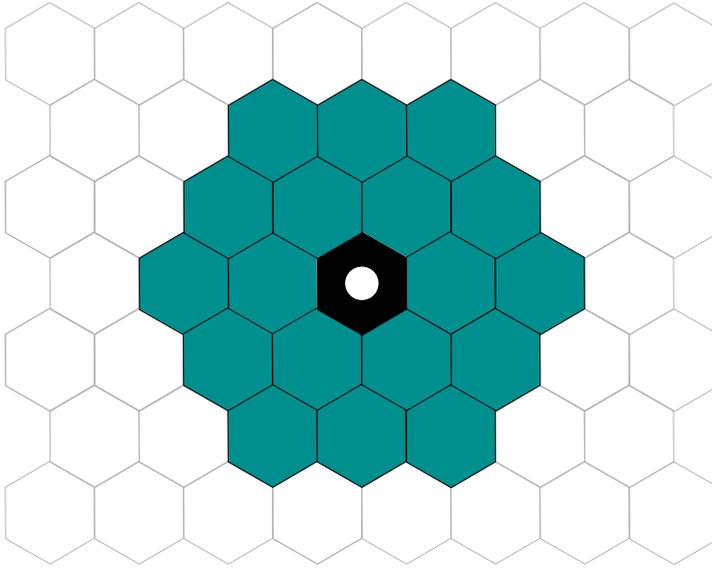


Figure 4.3: Cross coupling in a deformable mirror: the elements affected by a single actuator (indicated by the circle) are filled.

We can write a simplified model of the deformable mirror as a decomposable system, that will allow using the distributed control technique described in Chapter 3. We consider the problem of tracking the reference signal for the mirror shape, with \mathcal{H}_2 performance. The reference signal is then to be considered as a disturbance; we assume it to be a random walk. This will be a good first order approximation of atmospheric turbulence, see for example [21]. We also assume the control input $u \in \mathbb{R}^N$ to effect the mirror statically (through the influence matrix G), but after a time delay; we employ the technique of the integrator-in-the loop in order to enforce integral action, so we will integrate the control input in the model. Both the measured and performance outputs ($z, y \in \mathbb{R}^N$) are the error in the mirror shape. With these assumptions, we can write the state-space model as follows:

$$\left\{ \begin{array}{l}
 x(k+1) = \left(I_N \otimes \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{A_a} + G \otimes \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}}_{A_b} \right) x(k) + \\
 \quad + I_N \otimes \underbrace{\begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}}_{B_{w,a}} w(k) + I_N \otimes \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{B_{u,a}} u(k) \\
 y(k) = z(k) = I_N \otimes \underbrace{\begin{bmatrix} 1 & -1 & 0 \end{bmatrix}}_{C_{y,a}=C_{z,a}} x(k)
 \end{array} \right. \quad (4.3)$$

The model is of order $3N$ ($l = 3$). The signal $w \in \mathbb{R}^N$ represents the random noise moving the random walk of the reference signal, which is the first element of the local state. The control input u is first stored in the third element of the local state (for introducing a delay), and then integrated in the second state, through the influence matrix G which plays the role of pattern matrix. The second element of the local state is then the “compensating effect” due to the actuators, which means that the output is basically the difference between this value and the reference (the first local state). The parameter b is a weight for the intensity of the noise driving the motion of the reference signal.

With this model it is possible to solve the \mathcal{H}_2 suboptimal control problem, by minimizing γ under the conditions of Theorem 3.4. As we are in the case of Remark 3.4, the LMIs that need to be evaluated are only those concerning the maximum and minimum eigenvalue of \mathcal{G} , if we constrain the decision variable W_i to be “bold” ($W_i = \mathbf{W}_i = W_a + \lambda_i W_b$). We look for a dynamic output-feedback controller of the form:

$$\begin{cases} x_c(k+1) = (I_N \otimes A_{c,a} + G \otimes A_{c,b})x_c(k) + (I_N \otimes B_{c,a} + G \otimes B_{c,b})y(k) \\ u(k) = (I_N \otimes C_{c,a} + G \otimes C_{c,b})x_c(k) + (I_N \otimes D_{c,a} + G \otimes D_{c,b})y(k) \end{cases} \quad (4.4)$$

This controller can be found by minimizing γ^2 over the following LMI constraints:

$$\begin{bmatrix} \mathbf{W}_i & C_{z,a}X & C_{z,a} \\ * & X + X^T - P_i & I_l + S^T - J_i \\ * & * & Y + Y^T - H_i \end{bmatrix} \succ 0,$$

$$\begin{bmatrix} P_i & J_i & \mathbf{A}_i X + B_{u,a} \mathbf{L}_i & \mathbf{A}_i + B_{u,a} \mathbf{R}_i C_{y,a} & B_{w,a} \\ * & H_i & \mathbf{Q}_i & Y \mathbf{A}_i + \mathbf{F}_i C_{y,a} & Y B_{w,a} \\ * & * & X + X^T - P_i & I_l + S^T - J_i & 0 \\ * & * & * & Y + Y^T - H_i & 0 \\ * & * & * & * & I_{m_w} \end{bmatrix} \succ 0, \quad (4.5)$$

for $i = \underline{i}, \bar{i}$, and:

$$\text{trace}(W_a + \lambda_{\text{av}} W_b) < \frac{\gamma^2}{N}$$

where the decision variables are $X, Y, S, P_i = P_i^T, H_i = H_i^T, J_i, \mathbf{F}_i, \mathbf{R}_i, \mathbf{W}_i, \mathbf{L}_i$ and \mathbf{Q}_i ; \underline{i} and \bar{i} are defined as the indices for which $\lambda_{\bar{i}} = \bar{\lambda}, \lambda_{\underline{i}} = \underline{\lambda}$ and λ_{av} is the average of the eigenvalues, that can be easily computed as the trace of the pattern matrix divided by N (the trace is the sum of the eigenvalues). The controller matrices can be retrieved via (3.28); if we work these relations out, separating the constant part of each bold matrix from the part linear in λ_i , we obtain for this

situation the following explicit expressions:

$$\left\{ \begin{array}{l} \text{find non-singular } V, U \text{ such that } U = S - YX \\ D_{c,a} = R_a \\ D_{c,b} = R_b \\ C_{c,a} = (L_a - R_a C_{y,a} X) U^{-1} \\ C_{c,b} = (L_b - R_b C_{y,a} X) U^{-1} \\ B_{c,a} = V^{-1} (F_a - Y B_{u,a} R_a) \\ B_{c,b} = V^{-1} (F_b - Y B_{u,a} R_b) \\ A_{c,a} = -V^{-1} (Q_a - Y (A_a - B_{u,a} R_a C_{y,a}) X) U^{-1} - B_{c,a} C_{y,a} X U^{-1} - V^{-1} Y B_{u,a} C_{c,a} \\ A_{c,b} = -V^{-1} (Q_b - Y (A_b - B_{u,a} R_b C_{y,a}) X) U^{-1} - B_{c,b} C_{y,a} X U^{-1} - V^{-1} Y B_{u,a} C_{c,b} \end{array} \right. \quad (4.6)$$

It should be noticed that the size of the mirror, that is, the number N of elements, and the pattern/influence matrix G , only marginally influence the optimization problem described in (4.5). If we solve with respect to γ^2/N , that is basically a scaled version of the \mathcal{H}_2 norm, we notice that the parameters influenced by the pattern matrix are only its spectral properties, namely $\bar{\lambda}$, $\underline{\lambda}$ and λ_{av} . We can postulate that these spectral properties are to a certain extent independent of the size of the mirror, as long as the same influence function is used. For example, all influence matrices derived from (4.2), with $a = 1$, will have 1 on the diagonal, meaning that the trace of such matrices will always be N , implying that $\lambda_{av} = 1$ always. Figure 4.4 shows the eigenvalues of the influence matrices derived from (4.2), for hexagonal mirrors of different sizes. From the picture it is possible to see that the eigenvalues are always located in the same interval, and the effect of increasing the size is merely making the spectrum denser.

Actually it can be useful to find an upper and a lower bound for such eigenvalues; in fact, if we replace $\bar{\lambda}$ and $\underline{\lambda}$ in (4.5) with their upper and lower bound and solve, then the LMIs with the actual maximum and minimum eigenvalues are guaranteed to be solved as well, as they are convex combinations of the two which we solved (this issue of the convexity is better explained in [59]). So the solution obtained with the bounds of the eigenvalues would be applicable to *any* mirror with a certain influence function. The literature (e.g. [11]) contains results that can be used to prove that the spectrum of G is constrained to be in a fixed interval. The object of the next section is to provide a simple, self-contained proof which does not require a deep knowledge of all the related theory.

4.4 Spectra of influence matrices

It can be shown that the spectrum of all influence matrices coming from the same spatially invariant influence functions (as in (4.2)) is bounded. In order to do so, we will first evaluate the spectrum of the influence matrix (or better, operator) in the infinite case ($R \rightarrow \infty$), by making use of the theory of spatially invariant systems [6], and then relate this result to finite truncations.

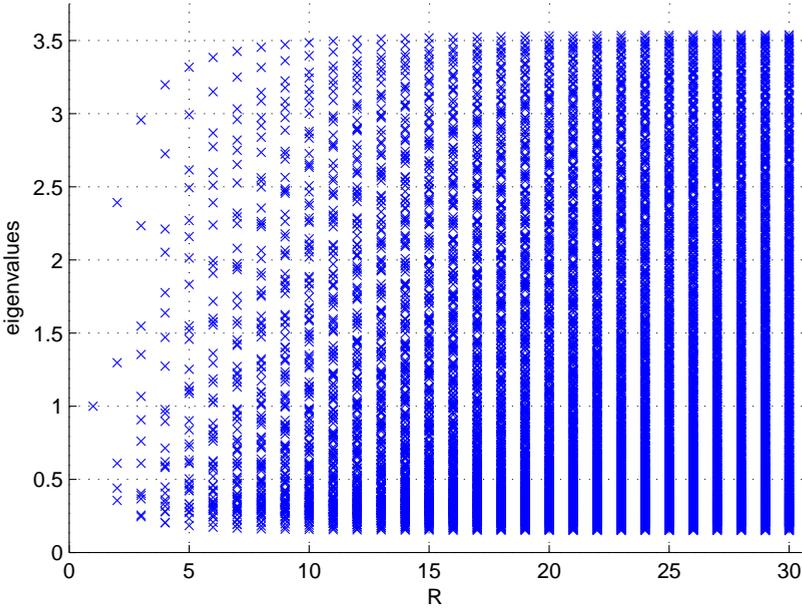


Figure 4.4: Spectrum of the influence matrix of a mirror as a function of the mirror size.

4.4.1 The infinite dimensional case

In the infinite case the mirror can be described as a infinite 2-dimensional hexagonal grid, identifiable with the Abelian group \mathbb{Z}^2 (where \mathbb{Z} is the set of the integers). We identify two directions or coordinates s_1 and s_2 as in Figure 4.5, and then we consider real functions of these coordinates ($x = x(s_1, s_2) : \mathbb{Z}^2 \rightarrow \mathbb{R}$); we define S_1 and S_2 as the two shift operators corresponding to the two directions, such as $S_1x(s_1, s_2) = x(s_1 + 1, s_2)$, and $S_2x(s_1, s_2) = x(s_1, s_2 + 1)$.

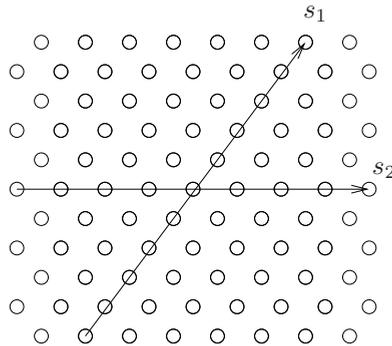


Figure 4.5: Infinite hexagonal grid, schematized as \mathbb{Z}^2 .

We define for two functions x and y of this kind an inner product:

$$\langle x, y \rangle := \sum_{s_1=-\infty}^{+\infty} \sum_{s_2=-\infty}^{+\infty} x(s_1, s_2)y(s_1, s_2) \quad (4.7)$$

and a norm:

$$\|x\|^2 := \langle x, x \rangle. \quad (4.8)$$

We define as \mathcal{L}_2 the set of the functions whose norm is finite, and we restrict our attention to functions in this set. Let us consider the operators mapping \mathcal{L}_2 to \mathcal{L}_2 . We define the linear operator \mathcal{G} as the infinite extension of the influence matrix G , as follows:

$$(\mathcal{G}x)(s_1, s_2) := \sum_{s'_1=-\infty}^{+\infty} \sum_{s'_2=-\infty}^{+\infty} \zeta(s_1 - s'_1, s_2 - s'_2)x(s'_1, s'_2) \quad (4.9)$$

where ζ is an influence function related to (4.2), which depends on the physical distance between two elements on the grid.

For the case of $\sigma = 0.7$, this operator can be expressed as:

$$\begin{aligned} y = \mathcal{G}x = x &+ a_1(S_1 + S_2 + S_1^{-1} + S_2^{-1} + S_1 S_2^{-1} + S_1^{-1} S_2)x + \\ &+ a_2(S_1 S_2 + S_1^2 S_2^{-1} + S_1 S_2^{-2} + S_1^{-1} S_2^{-1} + S_1^{-2} S_2 + S_1^{-1} S_2^2)x + \\ &+ a_3(S_1^2 + S_1^2 S_2^{-2} + S_2^{-2} + S_1^{-2} + S_1^{-2} S_2^2 + S_2^2)x \end{aligned} \quad (4.10)$$

where $a_1 = 0.3604$, $a_2 = 0.0468$, and $a_3 = 0.0169$ are the values of (4.2) computed on the basis of the relative distance. A Fourier transform [77] in the two dimensions diagonalizes the equation above, that formally stays the same with S_1 and S_2 replaced by z_1 and z_2 (complex numbers on the unit circle), and x and \mathcal{G} replaced by its transformed version X and $\hat{\mathcal{G}}$:

$$\begin{aligned} \hat{\mathcal{G}}X = X &+ a_1 X(z_1 + z_2 + z_1^{-1} + z_2^{-1} + z_1 z_2^{-1} + z_1^{-1} z_2) + \\ &+ a_2 X(z_1 z_2 + z_1^2 z_2^{-1} + z_1 z_2^{-2} + z_1^{-1} z_2^{-1} + z_1^{-2} z_2 + z_1^{-1} z_2^2) + \\ &+ a_3 X(z_1^2 + z_1^2 z_2^{-2} + z_2^{-2} + z_1^{-2} + z_1^{-2} z_2^2 + z_2^2). \end{aligned} \quad (4.11)$$

As in the unit circle the inverse of a number is equivalent to its complex conjugate ($z^{-1} = \bar{z}$), then it can be pointed out that whenever a complex quantity appears in (4.11), it is summed to its conjugate (thanks to the symmetry of the operator). This means that the above relation is equivalent to:

$$\begin{aligned} \hat{\mathcal{G}}X = X &+ 2a_1 X \operatorname{Re}(z_1 + z_2 + z_1^{-1} z_2) + \\ &+ 2a_2 X \operatorname{Re}(z_1 z_2 + z_1^2 z_2^{-1} + z_1 z_2^{-2}) + \\ &+ 2a_3 X \operatorname{Re}(z_1^2 + z_1^2 z_2^{-2} + z_2^2). \end{aligned} \quad (4.12)$$

The spectrum of \mathcal{G} is then a continuous real interval between the maximum and minimum value of the function:

$$\begin{aligned} g = 1 &+ 2\operatorname{Re}(a_1(z_1 + z_2 + z_1^{-1} z_2) + \\ &+ a_2(z_1 z_2 + z_1^2 z_2^{-1} + z_1 z_2^{-2}) + a_3(z_1^2 + z_1^2 z_2^{-2} + z_2^2)) \end{aligned} \quad (4.13)$$

with $z_1 = e^{j\theta_1}$, $z_2 = e^{j\theta_2}$, and $\theta_1, \theta_2 \in [0, 2\pi]$. Gridding and plotting the function as in Figure 4.6, we find global maximum and minimum, that allows stating that the spectrum is given approximately by the interval $[0.1496, 3.5446]$. This matches with what is shown by Figure 4.4.

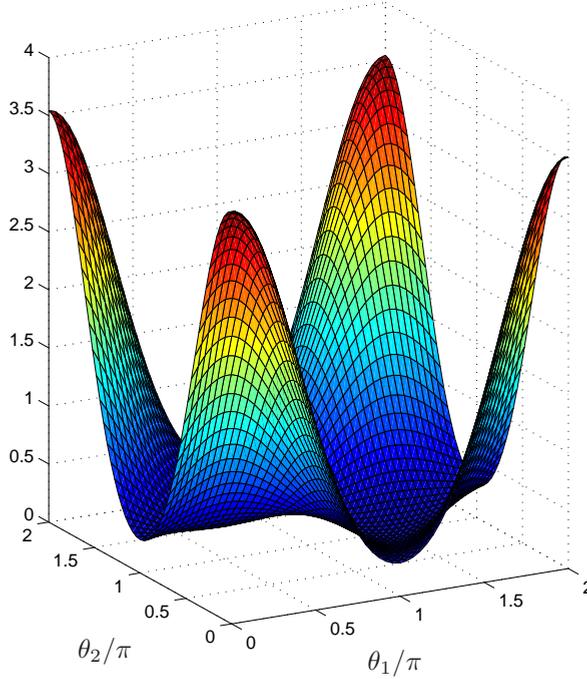


Figure 4.6: Spectrum of the operator \mathcal{G} .

4.4.2 From infinite to finite

We now consider finite influence matrices, that can be considered as truncations of the operators, and we try to relate the spectrum of the former to the one of the latter. In order to do so, we first state some properties of \mathcal{G} .

Lemma 4.1 (Properties of \mathcal{G})

1. \mathcal{G} is self-adjoint: $\langle x, \mathcal{G}y \rangle = \langle \mathcal{G}x, y \rangle \forall x, y \in \mathcal{L}_2$;
2. \mathcal{G} is translation invariant: $(S_1\mathcal{G})x = \mathcal{G}(S_1x)$, $(S_2\mathcal{G})x = \mathcal{G}(S_2x) \forall x \in \mathcal{L}_2$
3. \mathcal{G} is bounded if $I_\zeta = \sum_{s_1=-\infty}^{+\infty} \sum_{s_2=-\infty}^{+\infty} |\zeta(s_1, s_2)|$ exists and is a finite number.

Proof:

1. From (4.9) and (4.7):

$$\begin{aligned} \langle x, \mathcal{G}y \rangle &= \sum_{s_1} \sum_{s_2} x(s_1, s_2) (\mathcal{G}y)(s_1, s_2) = \\ &= \sum_{s_1} \sum_{s_2} \sum_{s'_1} \sum_{s'_2} \zeta(s_1 - s'_1, s_2 - s'_2) x(s_1, s_2) y(s'_1, s'_2) \end{aligned} \quad (4.14)$$

as ζ depends only on the distance, then we have that $\zeta(s_1 - s'_1, s_2 - s'_2) = \zeta(s'_1 - s_1, s'_2 - s_2)$. Recognizing that:

$$\sum_{s_1} \sum_{s_2} \zeta(s'_1 - s_1, s'_2 - s_2) x(s'_1, s'_2) = (\mathcal{G}x)(s'_1, s'_2) \quad (4.15)$$

this proves this part of the lemma.

2. From (4.9):

$$\begin{aligned} (S_1 \mathcal{G})x(s_1, s_2) &= \sum_{s'_1} \sum_{s'_2} \zeta(s_1 - s'_1 + 1, s_2 - s'_2) x(s_1, s_2) = \\ &= (\mathcal{G}x(s_1 + 1, s_2)) = \mathcal{G}(S_1 x(s_1, s_2)) \end{aligned} \quad (4.16)$$

proving immediately what stated.

3. It is immediate to see that for any $x \in \mathcal{L}_2$, it holds that $\|\mathcal{G}x\| \leq I_\zeta \|x\|$. \square

Theorem 4.1 [41] For any arbitrary bounded self-adjoint linear operator \mathcal{F} , the following:

$$\underline{\lambda}(\mathcal{F}) = \inf_{\|x\|=1} \langle \mathcal{F}x, x \rangle \quad (4.17)$$

$$\bar{\lambda}(\mathcal{F}) = \sup_{\|x\|=1} \langle \mathcal{F}x, x \rangle \quad (4.18)$$

are spectral points, and the spectrum is contained in the interval $[\underline{\lambda}(\mathcal{F}), \bar{\lambda}(\mathcal{F})]$.

Theorem 4.2 Consider a finite subset $\mathbb{G} \subset \mathbb{Z}^2$, and the functions $g : \mathbb{G} \rightarrow \mathbb{R}$. Let us define the embedding operator x_g mapping functions of \mathbb{G} to functions of \mathbb{Z}^2 :

$$(x_g g)(s_1, s_2) := \begin{cases} g(s_1, s_2) & \text{if } (s_1, s_2) \in \mathbb{G} \\ 0 & \text{if } (s_1, s_2) \notin \mathbb{G} \end{cases} \quad (4.19)$$

Let us also define the restriction operator x_g^* mapping functions of \mathbb{Z}^2 to \mathbb{G} :

$$(x_g^* x)(s_1, s_2) := x(s_1, s_2) \quad (4.20)$$

Notice that the restriction operator is the adjoint of the embedding operator. We define as G the linear function mapping the space of the real functions in \mathbb{G} to itself as follows:

$$(Gg)(s_1, s_2) := (x_g^* \mathcal{G} x_g g)(s_1, s_2) \quad (\forall (s_1, s_2) \in \mathbb{G}) \quad (4.21)$$

Notice that g corresponds to a vector in \mathbb{R}^k , where k is the number of elements in \mathbb{G} , and that G corresponds to the finite influence matrix previously defined, mapping the space of vectors \mathbb{R}^k into itself. Then it holds that:

$$\underline{\lambda}(G) \leq \underline{\lambda}(\mathcal{G}) \leq \bar{\lambda}(\mathcal{G}) \leq \bar{\lambda}(G) \quad (4.22)$$

Proof: We only show that $\underline{\lambda}(\mathcal{G}) \leq \underline{\lambda}(G)$, the rest of the theorem can be similarly proven. From Theorem 4.1 we have that:

$$\begin{aligned} \underline{\lambda}(G) &= \inf_{\|g\|=1} \langle x_g^* \mathcal{G} x_g, g \rangle = \inf_{\|g\|=1} \langle \mathcal{G} x_g, x_g \rangle = \\ &= \inf_{x \in \{x_g : \|g\|=1\}} \langle \mathcal{G} x, x \rangle \geq \inf_{\|x\|=1} \langle \mathcal{G} x, x \rangle = \underline{\lambda}(\mathcal{G}) \end{aligned} \quad (4.23)$$

where we exploited the fact that the set of $x \in \{x_g : \|g\| = 1\}$ is a subset of all the $x : \|x\| = 1$. \square

The concept expressed in Theorem 4.2 can be summarized as follows: the maximum and minimum values of the spectrum of a linear, bounded, self-adjoint (symmetric) operator are bounds for the spectrum of any finite linear operator (matrix) resulting from a truncation. This means that the upper and lower bounds of the spectrum of the operator are also upper and lower bounds for any finite influence matrix stemming from the same spatially invariant influence function. So, by choosing these bounds for $\bar{\lambda}$ and $\underline{\lambda}$ for the optimization problem in (4.5), we can find a suboptimal \mathcal{H}_2 controller that is applicable to any mirror with the same influence function, regardless of size and shape (hexagonal, rectangular or of another kind).

4.5 Simulation results

Simulations have been performed with a mirror model consisting of 127 force actuators and wavefront sensors positioned on a hexagonal grid of 7 rings. In optical imaging the objective is to maximize the Strehl ratio [36], which, for small phase aberrations, is approximately equivalent to minimizing the \mathcal{H}_2 norm of the transfer between the distortion w and the residual phase y . Therefore, a distributed controller minimizing the \mathcal{H}_2 norm between w and y has been computed by solving the LMI problem (4.5). For comparison also the centralized \mathcal{H}_2 optimal controller has been computed and a diagonal PI controller has been tuned close to optimality.

Table 4.1 gives the resulting \mathcal{H}_2 norms. As expected the performance obtained by the distributed controller is between the performance obtained by the \mathcal{H}_2 optimal centralized solution and decentralized PI controller.

Table 4.1: \mathcal{H}_2 performance from disturbance w to output y with various control methods.

Method	\mathcal{H}_2 norm
Distributed \mathcal{H}_2	19.6
Decentralized PI	25.2
Centralized \mathcal{H}_2	15.9

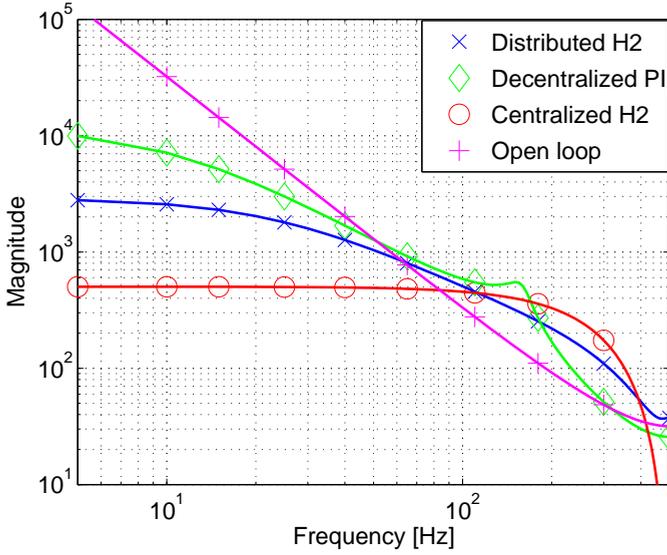


Figure 4.7: Spectral density of the mean square error with unit-covariance zero-mean white noise as input.

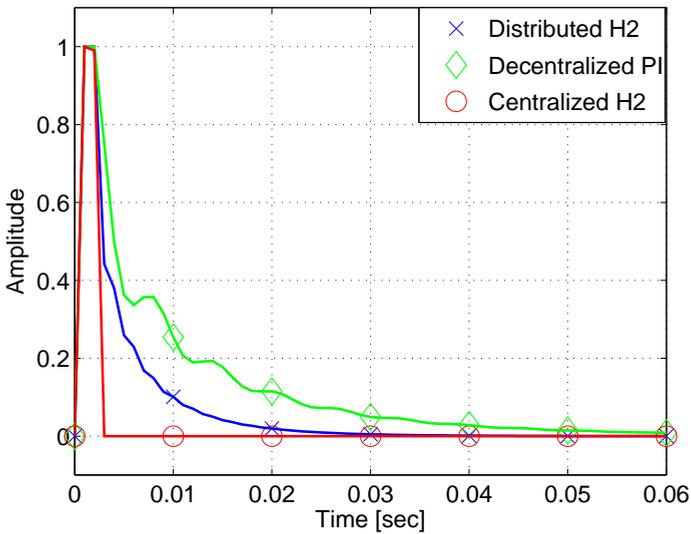


Figure 4.8: Impulse response of the first disturbance input to the first output.

Figure 4.7 shows the mean squared error spectral density of the output y as a response to w being a zero-mean white noise process with unit covariance, i.e., the frequency distribution of the \mathcal{H}_2 norm. We observe that the decentralized PI controller shows some resonance, whereas the distributed and the centralized \mathcal{H}_2 optimal controller provide a well damped closed loop. Figure 4.8 shows the closed-loop impulse responses from the first disturbance input to the first output obtained by the various controllers. The centralized controller completely cancels the impulse response after one sample due to the one sample delay in the system. The impulse response of the distributed controller is a bit longer, but shorter than the one of the decentralized PI controller.

4.6 Conclusions

In this chapter we have shown an application to adaptive optics of the methods for controller design that have been introduced in Chapter 3. We have shaped the model of a deformable mirror into a decomposable system, whose pattern matrix is not linked to a specific graph structure, even if it does have a significant sparsity. The properties of these pattern matrices have allowed bounding their eigenvalues, leading us to a similar result as the ones in Section 3.2.4 or [26]: controllers which hold for *any* mirror with the same influence function.

Further developments of the research might see the application of the method on an adaptive optics breadboard, and the possibility of applications of the methods to the problem of wavefront reconstruction.

Formation Flying Applications

Formation flying is a key technology that will allow a new generation of scientific space missions. The control methods of this thesis perfectly fit these kind of problems, proving to be an efficient way to approach formation flying control, looking both at positioning requirements as well as propellant consumption. We will first develop an extension of the \mathcal{H}_∞ synthesis method that will specifically suit the dynamics of spacecraft in non-circular orbits, and after that we are going to show the application in simulation to two case studies.

The first case study is a formation of microsattellites flying in a sun-synchronous orbit, inspired by the Formation for Atmospheric Science and Technology demonstration (FAST) mission, a Dutch-Chinese cooperation. The second case is inspired by future deep space telescope missions, which might feature formations of satellites covering halo orbits around Lagrangian points of the Sun-Earth system.

5.1 Introduction

Distributed space systems such as satellite formations play an ever increasing role in the design of space missions for applications such as Earth observation, communication, navigation, servicing and exploration. These distributed systems often call for an efficient control approach, which minimizes control expenses in terms of propellant, communication bandwidth and computational load while assuring stability and good performance of the distributed system. Performance in this context typically refers to the maintenance of prescribed relative positions, velocities and orientations.

There is a large number of publications on satellite formation flight, which has emerged over the last decade, see for example [7; 45; 63; 79; 90] and references therein. The literature on position control of satellite formations typically con-

siders following a nominal trajectory while minimizing propellant consumption [40; 54; 83].

In this chapter we show how the \mathcal{H}_∞ synthesis method introduced in Chapter 3 can be applied to formation flying problems, in an extended form that better suits their specific features. Namely, we extend Theorem 3.3, which applies to time-invariant system, to the so-called “Linear Time-Periodic” (LTP) models, which can be used as an approximation of the dynamics of spacecraft following an orbit of any kind. An LTP approach to the control of spacecraft in non-Keplerian orbits has been described in [47]. In this perspective, the contents of this chapter can be considered also as an extension of [47] to *distributed* formation flying.

Applying the proposed method brings a number of advantages. The use of distributed control techniques makes it possible to have a local implementation, on each single satellite, without the need for a central processing unit that has the complete knowledge of the state of the entire formation. Moreover, the method is based on a performance metric (the \mathcal{H}_∞ norm), which makes it especially suitable for long term station keeping control as it allows minimizing the propellant consumption, whereas other methods, e.g., those based on artificial potential functions [42; 61; 84] do not have any means of optimizing the propellant consumptions, though they might be more efficient for proximity maneuvers and collision avoidance. At last, we have seen that the use of the properties of Laplacian matrices allows finding controllers with guaranteed performance for all the possible formations, regardless of the number of elements in it. This makes it possible to have flexible and reconfigurable formations.

In order to illustrate the efficacy of our approach we consider two application examples involving satellite formation control, the first of which is tailored to the specifications of a mission currently under development, the Dutch-Chinese Formation for Atmospheric Science and Technology demonstration (FAST) [53]. This mission features a core of two microsattelites with sensors collecting data on atmospheric aerosols. The use of formation flying can provide superior scientific data both in quality and quantity with respect to what the two spacecraft alone could deliver. The formation will, in specific mission phases, be operated as a train (a line of satellites one after the other) and the use of standard interfaces will allow other nations to join the train configuration, making the size of the formation grow. This makes the efficient control of such a train a particular challenge, and the method proposed here is able to give the mission the required flexibility. The simulation results show the effectiveness of the controller in keeping the formation at an affordable propellant cost. We present also a second, more qualitative example, inspired by the work in [47], featuring a formation in a halo orbit, which has numerous envisioned applications in future missions [1; 2].

The chapter is structured as follows. Section 5.2 briefly presents the Linear Time-Periodic models and it shows how spacecraft dynamics can be approximated with such models. Section 5.3 contains the controller synthesis method that is the theoretical result of this chapter. Sections 5.4 describes the two benchmark cases, the first of which is explained in more details and focused on the FAST mission. In Section 5.5, the conclusions are drawn and some possible extensions of the method are described.

The results contained in this chapter have been published in [56].

5.2 Linear Time-Periodic models

In Section 3.3 we have shown an example featuring a formation of satellites flying in a circular orbit. The relative close motion of spacecraft in circular orbits can be described by means of the LTI model in (3.35) (the CW equations), but this approximation is no longer useful in case the orbit is non-circular or even non-Keplerian (not elliptical). We have then to consider a more general approach.

If we consider a high-level description of a spacecraft for relative positioning purposes as a point mass with 3 degrees of freedom, we can describe its unperturbed motion in a stationary gravitational force field in general by means of a time-invariant differential equation in its state vector x , which comprises the coordinates of the body (in any reference system) and its velocity. A solution $x(t)$ of such equation is a trajectory in the state-space that satisfies the differential equation at any time t , and it will depend on the initial conditions x_0 :

$$x = x(t, x_0). \quad (5.1)$$

A solution is called “periodic” of period T if it has the property that $x(t, x_0) = x(t + T, x_0)$ for any t .

If we introduce additional forces, i.e., disturbance forces w (e.g., solar radiation, unmodeled dynamics, etc.) and control forces u (e.g., thrust), then the solution to the equations will depend on these forces too:

$$x = x(t, x_0, w, u). \quad (5.2)$$

The purpose of a station-keeping controller is to keep the body as close as possible to a nominal, desired trajectory which we call $\bar{x}(t)$, by employing the control actions that are possible. The nominal trajectory is a periodic solution that we would have ideally in case the disturbance and control forces are absent, with the correct initial conditions: $\bar{x}(t) = x(t, \bar{x}_0, 0, 0)$. The presence of external forces will change the trajectory making it drift from the nominal one. If the nominal trajectory is stable, then the perturbed one will not diverge too much from it, whereas if it is unstable it will diverge quickly. In general, the effect of these perturbations is of nonlinear nature. It is however possible to use linear methods for dealing with the problem of controlling the motion of a body following a nominal trajectory by means of a so-called Linear Time-Periodic (LTP) approximation of the dynamics.

Let us divide the period T into p intervals of duration ΔT . We can then replace the continuous-time dynamics with a discrete-time dynamics by writing, with a little abuse of notation, $\bar{x}(k)$ in place of $\bar{x}(k\Delta T)$, where k is an integer that from now on will replace the time. It is obvious then, due to the periodicity, that $\bar{x}(k) = \bar{x}(k+p)$. For a body moving under no perturbations we will just have $x(k, x_0, 0, 0) = \bar{x}(k)$, but introducing the perturbation terms will cause an additional term to appear: $x(k, x_0, w, u) = \bar{x}(k) + \Delta x(k)$.

We can think of this additional term $\Delta x(k)$ as something that evolves from one time step to the other, as a function of its value at the previous time step and of the values of the input terms u and w which we assume constant during the interval. In general this dependence will be nonlinear, but we can write a linearized approximation of the dynamics:

$$\begin{aligned} x(k, x_0, w, u) &= \bar{x}(k) + \Delta x(k) \approx \\ &\approx \bar{x}(k) + \frac{\partial \Delta x(k)}{\partial \Delta x(k-1)} \Delta x(k-1) + \frac{\partial \Delta x(k)}{\partial u(k-1)} u(k-1) + \frac{\partial \Delta x(k)}{\partial w(k-1)} w(k-1). \end{aligned} \quad (5.3)$$

This allows us to write the dynamics of the perturbed part in a state-space formulation, with time-varying state-space matrices ($A(k)$, $B_u(k)$, $B_w(k)$):

$$\Delta x(k+1) = A(k)\Delta x(k) + B_u(k)u(k) + B_w(k)w(k) \quad (5.4)$$

where $A(k) = \frac{\partial \Delta x(k)}{\partial \Delta x(k-1)}$, $B_u = \frac{\partial \Delta x(k)}{\partial u(k-1)}$ and $B_w = \frac{\partial \Delta x(k)}{\partial w(k-1)}$, constituting the LTP model that can be used for linear controller design.

The goal of a controller for station keeping of a single body would then be to keep this perturbation part as close as possible to zero. The nominal part can be considered as “mapped” a priori and we will neglect it from now on and consider only the perturbation part. This part is described as we have seen by an LTP model. In its general form, a state-space LTP model is given by the equations:

$$\begin{cases} x(k+1) = A(k)x(k) + B_w(k)w(k) + B_u(k)u(k) \\ z(k) = C_z(k)x(k) + D_{zw}(k)w(k) + D_{zu}(k)u(k) \\ y(k) = C_y(k)x(k) + D_{yw}(k)w(k) \end{cases} \quad (5.5)$$

where $x(k)$ is a generic state, and y and z are two different outputs: y is the measured output, the one that the controller can use for determining the control action, and z is the performance output, the quantity that the controller has to minimize. For all the matrices $M(k)$ in (5.5) it holds that $M(k+p) = M(k)$; in the case of $p = 1$ we have a Linear Time-Invariant (LTI) system.

5.3 Distributed controller synthesis for Linear Time-Periodic models

If we describe the motion of spacecraft in generic orbits by means of an LTP model, then what we need to get a controller for them is to upgrade Theorem 3.3 in order to make it able to handle this kind of model. We will first define the model of a formation as a kind of LTP version of a decomposable system, and after that we are going to report the controller synthesis theorem.

Consider a formation of N spacecraft (or vehicles in general), whose identical dynamics can be described by a p -periodic linear time-varying system as follows:

$$x_i(k+1) = A(k)x_i(k) + B_u(k)u_i(k) + B_w(k)w_i(k) \quad (5.6)$$

where $x_i(k)$, $u_i(k)$ and $w_i(k)$ are respectively the vectors containing the states, the control inputs and the disturbance inputs of the i^{th} vehicle, and $A(k)$, $B_u(k)$ and $B_w(k)$ are time-varying state-space matrices.

Let us assume that the measurements that are available for control are given by:

$$y_i(k) = C_y(k)x_i(k) + D_{yw}(k)w_i(k) \quad (5.7)$$

where $C_y(k)$ is an output matrix and $D_{yw}(k)$ allows accounting for measurement noise.

We also define $x(k) = [x_1^T(k) \ x_2^T(k) \ \dots \ x_N^T(k)]^T$, $w(k) = [w_1^T(k) \ w_2^T(k) \ \dots \ w_N^T(k)]^T$ and $y(k) = [y_1^T(k) \ y_2^T(k) \ \dots \ y_N^T(k)]^T$. So far the dynamics of the vehicles are decoupled. Let us now introduce a global coupling performance index z as follows:

$$z(k) = (I_N \otimes C_{z,a}(k) + \mathcal{P} \otimes C_{z,b}(k))x(k) + (I_N \otimes D_{zu}(k))u(k) \quad (5.8)$$

where \mathcal{P} is a symmetric ‘‘pattern matrix’’, which has respectively $\bar{\lambda}$ and $\underline{\lambda}$ as upper and lower bounds for its real eigenvalues. $D_{zu}(k)$ allows putting a penalty on the use of the actuators. Notice that the resulting dynamical system has all its matrices in the form defined by (2.34), thus they can be decomposed according to Theorem 2.3. We look for distributed LTP controllers of the same form:

$$\begin{cases} x_c(k+1) = (I_N \otimes A_{c,a}(k) + \mathcal{P} \otimes A_{c,b}(k))x_c(k) + \\ \quad + (I_N \otimes B_{c,a}(k) + \mathcal{P} \otimes B_{c,b}(k))y(k) \\ u(k) = (I_N \otimes C_{c,a}(k) + \mathcal{P} \otimes C_{c,b}(k))x_c(k) + \\ \quad + (I_N \otimes D_{c,a}(k) + \mathcal{P} \otimes D_{c,b}(k))y(k) \end{cases} \quad (5.9)$$

Theorem 5.1 *Let T_{wz} be the map from disturbance w to output z of the system in (5.6), (5.7) and (5.8) in closed loop with the controller in (5.9). There exists a distributed (or sparse) p -periodic time-scheduled controller of such form that stabilizes the system and yields an l_2 gain strictly smaller than γ if there is a feasible solution for the LMIs in (5.10) on the next page, where $X(k)$, $Y(k)$, $S(k)$, $P_i(k) = P_i(k)^T$, $H_i(k) = H_i(k)^T$, $J_i(k)$, $L_a(k)$, $L_b(k)$, $F_a(k)$, $F_b(k)$, $Q_a(k)$, $Q_b(k)$, $R_a(k)$ and $R_b(k)$ are the decision variables. The matrices of the controller can be retrieved from (5.11), shown again on next page.*

Proof: *The proof follows along the lines of Theorem 3.3, but using the Linear Time-Varying framework that was developed in [23] as a starting point. We do not report the complete proof as it is quite long and it does not contain any special insight, but the interested reader can find it in [56].* \square

As we are again under the conditions of Remark 3.4, neither the matrix \mathcal{P} nor the number of vehicles N appear explicitly in the theorem: the only way they influence the computation is through the maximum and minimum eigenvalues of \mathcal{P} . This implies that the same controller can be valid for a whole class of formations, with different number of elements, at the condition that these maximum and minimum eigenvalues do not change (or that they are bounded between the two values that were used in the synthesis). That is why this control design approach is very flexible, as it allows both changes in the number of vehicles as well as changes

$$\begin{bmatrix}
 P_i(k+1) & J_i(k+1) & A(k)X(k)+B_u(k)\mathbf{L}_i(k) & A(k)+B_u(k)\mathbf{R}_i(k)C_y(k) & B_w(k)+B_u(k)\mathbf{R}_i(k)D_{yw}(k) & 0 \\
 * & H_i(k+1) & \mathbf{Q}_i(k) & Y(k+1)A(k)+\mathbf{F}_i(k)C_y(k) & Y(k+1)B_w(k)+\mathbf{F}_i(k)D_{yw}(k) & 0 \\
 * & * & X(k)+X(k)^T-P_i(k) & I_i+S(k)^T-J_i(k) & 0 & X(k)^T\mathbf{C}_{z,i}(k)^T+\mathbf{L}_i(k)^TD_{zu}(k)^T \\
 * & * & * & Y(k)+Y(k)^T-H_i(k) & 0 & \mathbf{C}_{z,i}(k)^T+C_y(k)^T\mathbf{R}_i(k)^TD_{zu}(k)^T \\
 * & * & * & * & I_{m_w} & D_{yw}(k)^T\mathbf{R}(k)^TD_{zu}(k)^T \\
 * & * & * & * & * & \gamma^2 I_{r_z}
 \end{bmatrix} \gamma \mathbf{0}$$

for $k = 1, \dots, p$
 for $i = \underline{i}, \bar{i}$

(5.10)

$$\left\{ \begin{array}{l}
 \text{find non-singular } V(k), U(k) \text{ such that } U(k) = S(k) - Y(k)X(k) \\
 D_{c,a}(k) = R_a(k) \\
 D_{c,b}(k) = R_b(k) \\
 C_{c,a}(k) = (L_a(k) - R_a(k)C_y(k)X(k))U(k)^{-1} \\
 C_{c,b}(k) = (L_b(k) - R_b(k)C_y(k)X(k))U(k)^{-1} \\
 B_{c,a}(k) = V(k+1)^{-1}(F_a(k) - Y(k+1)B_u(k)R_a(k)) \\
 B_{c,b}(k) = V(k+1)^{-1}(F_b(k) - Y(k+1)B_u(k)R_b(k)) \\
 A_{c,a}(k) = -V(k+1)^{-1}(Q_a(k) - Y(k+1)(A(k) - B_u(k)R_a(k)C_y(k))X(k))U(k)^{-1} + \\
 \quad - B_{c,a}(k)C_y(k)X(k)U(k)^{-1} - V(k+1)^{-1}Y(k+1)B_u(k)C_{c,a}(k) \\
 A_{c,b}(k) = -V(k+1)^{-1}(Q_b(k) + Y(k+1)(B_u(k)R_b(k)C_y(k))X(k))U(k)^{-1} + \\
 \quad - B_{c,b}(k)C_y(k)X(k)U(k)^{-1} - V(k+1)^{-1}Y(k+1)B_u(k)C_{c,b}(k)
 \end{array} \right. \quad (5.11)$$

in the interconnection structure: the only action needed is a controller reconfiguration that accommodates the new pattern matrix as it explicitly appears in the controller equations (5.9). This makes such a controller very suitable for missions with open architectures such as the FAST mission.

5.4 Test cases

5.4.1 The FAST mission

We consider an extended version of the already mentioned FAST mission, with a number $N = 10$ of satellites following one another along-track in a quasi-circular orbit (this special configuration is referred to as a *train*). The satellites have direct measurements of their absolute position thanks to GPS receivers [50; 67], and they exchange this information in order to compute their relative positions with respect to their neighbors. Typically the satellites would also exchange raw data, i.e. pseudoranges and carrier phases, and they would also have local filters to determine position and velocity, but we do not consider this aspect as we are going to develop an \mathcal{H}_∞ controller that includes in itself a state estimator. The spacecraft are equipped with thrusters which can execute impulsive maneuvers for trajectory corrections.

The orbit is a sun-synchronous orbit at an altitude of 650 km and inclination of 98° (Figure 5.1), with a period of approximately 5320 s (1h 28' 40'').



Figure 5.1: Orbit of the FAST mission.

A possible approach to this orbit could be to use the linear CW equations [46] in order to describe the dynamics of the perturbation of the motion of each spacecraft with respect to its own nominal circular Keplerian orbit. These equations

are valid for a point-mass gravity source and spacecraft separations that are much smaller than the distance of the reference point to the center of gravity. Using the CW equations would neglect the fact that low-Earth orbits are significantly different from Keplerian ones, mainly due to the effects of the Earth's oblateness, or J_2 effects [46]. In fact, it is the J_2 component of the gravity field that causes the line of the nodes to drift at the rate of approximately 0.985 deg/day, making the orbit sun-synchronous. For this reason, the J_2 effect cannot be neglected, and we will use, instead of the CW model, an LTP model as in (5.5) which takes into account this component, obtained numerically as explained in Section 5.2. We divide the orbit into 100 segments, finding the nominal position where each satellite will be at each time step. We consider a distance of 10 km between each neighboring satellite, which allows us to assume that the satellites are all governed by the same dynamic law (e.g., they are located in the same piece of the 100 orbital segments making up the LTP model). We still consider the local orbital coordinate frame as in the CW equations. It is also to be taken into account that the orbit is non-circular, so it is not covered at the same velocity at all times. This causes natural oscillations in terms of the relative positions of the satellites. It is possible to show by means of simulation that for two satellites following each other, along track, with an initial distance of 10 km, we will have an oscillation of an amplitude of approximately 10 m in the relative distance in the course of an orbit. The controller should not try to counteract this effect (as it would be a useless waste of propellant), so this natural oscillation will be accounted for when computing relative distances.

In order to compute the feedback controller, we formulate the problem as a global optimization problem, introducing measurement errors and disturbances, and specifying measured outputs and performance outputs. We assume that the satellites correct their trajectories by means of impulsive maneuvers executed at the beginning of every segment of orbit. The inputs to the satellites shall then be the variation of velocity (or Δv) caused by thrust, the accelerations caused by external forces and the errors in the position measurements. The Δv is the control input u , determined by the controller, while the external forces are considered *disturbance inputs* w and they are assumed to be unknown and uncontrollable (and they are assumed to be constant during each orbit segment). The outputs are the relative and absolute positions of the satellites. Again we will define as measurements (or outputs used for control) the signals y that the controller can monitor in order to decide its control actions. We will have also performance outputs z that are the error signals that the controller will try to minimize. In this case, the measurements are the absolute positions of the satellites (coming from GPS receivers), while the performance outputs are given by the errors in the relative positions. As the controller must also minimize the propellant consumption, the Δv caused by thrust will be considered as a performance output as well.

Once these signals have been defined, the system of N satellites in formation can be formulated as follows:

$$\begin{cases} x(k+1) = I_N \otimes A(k)x(k) + I_N \otimes \mathcal{B}_w(k)w(k) + I_N \otimes \mathcal{B}_u(k)u(k) \\ y(k) = I_N \otimes \mathcal{C}_y x(k) + I_N \otimes \mathcal{D}_{yw}w(k) \end{cases} \quad (5.12)$$

In this last equation, $x \in \mathbb{R}^{Nl}$ is the vector containing all the states of all the satellites; we will call x_i the vector containing the $l = 6$ states of the i^{th} satellite. The states comprise the variations with respect to nominal position and nominal velocity in the three directions. There are Nr_y control outputs and Nr_z performance outputs, which are respectively in the vectors $y \in \mathbb{R}^{Nr_y}$ and $z \in \mathbb{R}^{Nr_z}$. There are also Nm_u control inputs (the signals that the controller produces) to the system, stored in the vector $u \in \mathbb{R}^{Nm_u}$; the disturbances are in the vector $w \in \mathbb{R}^{Nm_w}$. We define the disturbances as follows:

$$w_i = [p_{R_i} W_p \quad p_{T_i} W_p \quad p_{N_i} W_p \quad q_{R_i} W_q \quad q_{T_i} W_q \quad q_{N_i} W_q]^T \quad (5.13)$$

where p_{\bullet} represents the disturbance accelerations in the three directions and q_{\bullet} represent the measurement noise to be added to the output of the GPS receiver. Such disturbances have different units, that is why two weighting constants W_p and W_q have been introduced in order to have that p_{\bullet} and q_{\bullet} , without weights, can be considered as dimensionless, homogeneous, random noise.

The control inputs are simply the Δv 's in the three directions due to the impulsive maneuvers:

$$u_i = [u_{R_i} \quad u_{T_i} \quad u_{N_i}]^T = [\Delta v_{R_i} \quad \Delta v_{T_i} \quad \Delta v_{N_i}]^T \quad (5.14)$$

As measurement output we define the following:

$$\begin{aligned} y_i &= [y_{R_i} \quad y_{T_i} \quad y_{N_i}]^T = \\ &= [\Delta r_{R_i} + q_{R_i} W_q \quad \Delta r_{T_i} + q_{T_i} W_q \quad \Delta r_{N_i} + q_{N_i} W_q]^T \end{aligned} \quad (5.15)$$

where the three y_{\bullet} are the absolute positions corrupted by the measurement noise.

Before defining the performance output, let us point out that the dynamics of each single satellite is independent from the other ones, so it turns out that all the state-space matrices in (5.12) are block diagonal, with identical diagonal blocks. The performance output, generated by the matrix which we call C_z , introduces cross coupling: since we aim at maintaining the relative positions of the satellites, then it must contain off-diagonal terms that allow calculating the differences between the positions of neighboring spacecraft. This cross-coupling implies that the control synthesis problem cannot be approached anymore by considering every satellite as independent from the other. The system can be considered as the LTP version of a decomposable system if such matrix can be written as:

$$C_z = I_N \otimes C_{z,a} + \mathcal{P} \otimes C_{z,b} \quad (5.16)$$

where \mathcal{P} is the ‘‘pattern matrix’’. We can formulate the performance output in such a way that it contains the relative positions (as our goal is minimizing them) and at the same time can be expressed by means of a symmetric, spectrally bounded pattern matrix. For example, if we choose the following outputs:

$$z_{\bullet i} = \begin{cases} \frac{1}{2} \Delta r_{\bullet i} - \frac{1}{2} \Delta r_{\bullet i+1} & \text{for } i = 1 \\ -\frac{1}{2} \Delta r_{\bullet i-1} + \Delta r_{\bullet i} - \frac{1}{2} \Delta r_{\bullet i+1} & \text{for } 2 \leq i \leq n-1 \\ \frac{1}{2} \Delta r_{\bullet i} - \frac{1}{2} \Delta r_{\bullet i-1} & \text{for } i = n \end{cases} \quad (5.17)$$

then it can be easily seen that this output can be obtained with the following pattern matrix:

$$\mathcal{P} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & 0 & 0 & \cdots & 0 & 0 \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & \cdots & 0 & 0 \\ 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ 0 & 0 & \cdots & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} \\ 0 & 0 & \cdots & 0 & 0 & -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (5.18)$$

with $C_{z,a} = 0$, $C_{z,b} = C$ (where C is just a matrix that selects the positions of the satellites as outputs). The matrix \mathcal{P} is a weighted normalized graph Laplacian and its eigenvalues are real and confined between 0 and 2 regardless of its size. The controller can thus be computed for virtually any formation size, with always the same level of guaranteed performance.

With such choice of \mathcal{P} , the performance output z can be chosen as the following:

$$z = \mathcal{P} \otimes \begin{bmatrix} C \\ 0 \end{bmatrix} x + RI_N \otimes \begin{bmatrix} 0 \\ I_3 \end{bmatrix} u \quad (5.19)$$

In this way, both the errors and the control effort are considered as a cost. Again, the $z_{\bullet,i}$ and the $u_{\bullet,i}$ are heterogeneous, but become commensurable thanks to the weights. The scalar parameter R is the penalty on control effort. This R can be thought of as a design parameter that allows trading off relative position performance with fuel savings: a higher value of R will generate a controller that minimizes propellant consumption, while a smaller R will result in more accurate relative positioning.

At this point, the method shown in Section 5.3 can be applied. Thanks to the sparsity of the pattern matrix in (5.18), this controller will be implementable as a distributed controller made of local controllers located in each satellite; each of these controllers needs to know only the position measurements of the satellite where it is located, and the ones of the satellites preceding and following, together with their controller states.

In order to perform the synthesis, it is also necessary to choose the design parameters that have been described earlier in this section. This data can be determined from the mission specifications [30], or otherwise estimated with empirical criteria. The weights W_p and W_q are chosen such as to represent the normal value of disturbance accelerations and measurement noise; we assume $W_p = 10^{-7} \text{ m/s}^2$ ([64] mainly due to aerodynamic drag) and $W_q = 2 \text{ m}$ (standard deviation of GPS measurements). We computed controllers for different values of R and we chose $R = 100$ as best compromise.

We simulated the effect of the chosen controller on 6000 orbits (approximately 1 solar year). One of the parameters that we consider is the total Δv that the satellites need, in order to see whether the propellant consumption is feasible or not. It is possible to translate the Δv requirements into propellant requirements

with Tsiolkovsky's rocket equation [72]:

$$\Delta v = g_0 I_{sp} \ln \left(\frac{m_0 + m_p}{m_0} \right) \quad (5.20)$$

where g_0 is the standard acceleration of gravity (9.81 m/s²), I_{sp} is the specific impulse of the thruster, m_0 is the satellite's dry mass and m_p is the total propellant mass. In this way, it is possible to compute the propellant that is needed (in terms of fraction of the total satellite mass) in order to maintain the formation for one year. The results of the simulation are shown in Table 5.1.

Table 5.1: Distributed control of the formation: parameters and results of the first simulation.

\mathcal{H}_∞ norm of closed loop system	$5.47 \cdot 10^{-2}$
Δv needed for 1 year	251.5 m/s
Fraction of propellant (cold gas thruster, $I_{sp} = 70$ s)	30.69 %
Fraction of propellant (hydrazine thruster, $I_{sp} = 230$ s)	10.56 %
Average error in relative positions	0.72 m
Maximum error in relative positions	3.35 m

The \mathcal{H}_∞ norm of the closed-loop system is a parameter that does not say much if considered alone; it has been included in the table in order to compare it with the value that we would get if we synthesize a centralized controller, for example with the method in [47]. The \mathcal{H}_∞ norm for the centralized controller is $5.28 \cdot 10^{-2}$, which is only 3.4% lower: this means that the distributed approach does not result in a significant loss of performance. The consumption of propellant is high but acceptable for both thruster options.

In order to evaluate a possible means of reducing the propellant consumption, we have carried out other simulations where each satellite uses its thrusters only if its relative position with respect to its neighbors has an error that exceeds a certain threshold. Keeping the propellant consumption low is a critical issue in space flight, and this method is a common practice that allows the controller to use a smaller effort, resulting though in bigger average error.

We simulated the formation for two different values of the threshold; the results are shown in Table 5.2 and Table 5.3; for the second simulation, the error and cumulative Δv for each satellite are shown (only for the first 100 orbits) in Figure 5.2 and 5.3 respectively.

As could be predicted, the introduction of the threshold dramatically decreases the propellant costs; the higher the threshold, the higher the savings, at the cost of higher inaccuracy. It is also possible to notice some chattering in Figure 5.2 due to the on/off nature of the controller, but this does not appear to compromise stability. This simulation shows the feasibility of the mission concept and the possibility of keeping the formation operational for one or more years.

Table 5.2: Distributed control of the formation: results of the first simulation with on/off control.

Threshold	5 m
Δv needed for 1 year	24.5 m/s
Fraction of propellant (cold gas thruster, $I_{sp} = 70$ s)	3.51 %
Fraction of propellant (hydrazine thruster, $I_{sp} = 230$ s)	1.08 %
Average error in relative positions	1.18 m
Maximum error in relative positions	5.66 m

Table 5.3: Distributed control of the formation: results of the second simulation with on/off control.

Threshold	10 m
Δv needed for 1 year	15.3 m/s
Fraction of propellant (cold gas thruster, $I_{sp} = 70$ s)	2.20 %
Fraction of propellant (hydrazine thruster, $I_{sp} = 230$ s)	0.67 %
Average error in relative positions	4.31 m
Maximum error in relative positions	14.32 m

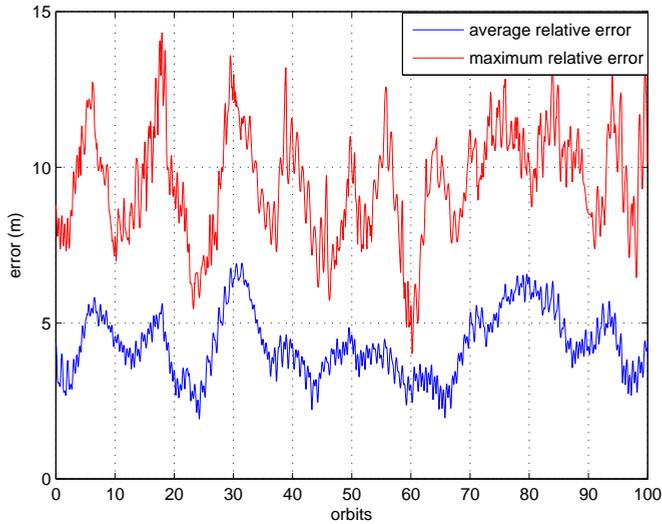


Figure 5.2: Errors in the relative positions of the satellites in the formation, for the second simulation with on/off control. The controller is turned on only if the relative position error exceeds 10 m.

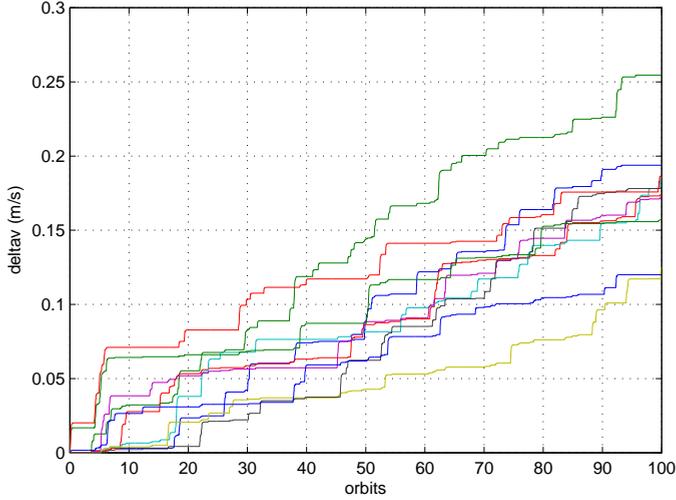


Figure 5.3: Cumulative Δv employed during the course of 100 orbits, for the second simulation with on/off control. Every line represents the consumption of one of the ten satellites; the functions increase in steps due to the on/off nature of the controller, caused by the use of the threshold (10 m).

5.4.2 Formation in a halo orbit

The Lagrangian points are the five equilibrium points for a point mass moving in the rotating frame of two celestial bodies revolving around their mutual center of mass. These points feature stable or unstable orbits which are called “halo orbits”; the use of satellites in these orbits has been of subject of interest for a considerable time [24]. Future missions also envision the possibility of formations of spacecraft in proximities of Lagrangian points in order to perform far-range astronomy [28]. For this reason we show the applicability of the control method also to formations located in halo orbits. We consider the Circular Restricted Three Body Problem for an object moving in the Sun-Earth system. The motion of a body in such a situation is described by the following differential equations:

$$\begin{cases} \ddot{X} = 2n\dot{Y} + \frac{\partial U}{\partial X} \\ \ddot{Y} = -2n\dot{X} + \frac{\partial U}{\partial Y} \\ \ddot{Z} = \frac{\partial U}{\partial Z} \end{cases} \quad (5.21)$$

where X , Y and Z are the coordinates of the spacecraft, n is the angular velocity of the revolution of the Sun-Earth system and U is the combined gravitational and centrifugal potential, defined as:

$$U = \frac{1}{2}n^2(X^2 + Y^2) + \frac{Gm_1}{r_1} + \frac{Gm_2}{r_2} \quad (5.22)$$

with G the universal gravitational constant, m_1 and m_2 the masses of the Sun and the Earth, and r_1 and r_2 the distances of the spacecraft with respect to these two celestial bodies (see [93] for further details). We considered the equations in their non-dimensional form, and a direct shooting method [9] was used for finding numerically a periodic orbit around L_1 , that is shown in Figure 5.4. Orbits around such points are unstable and need to be maintained with active control.

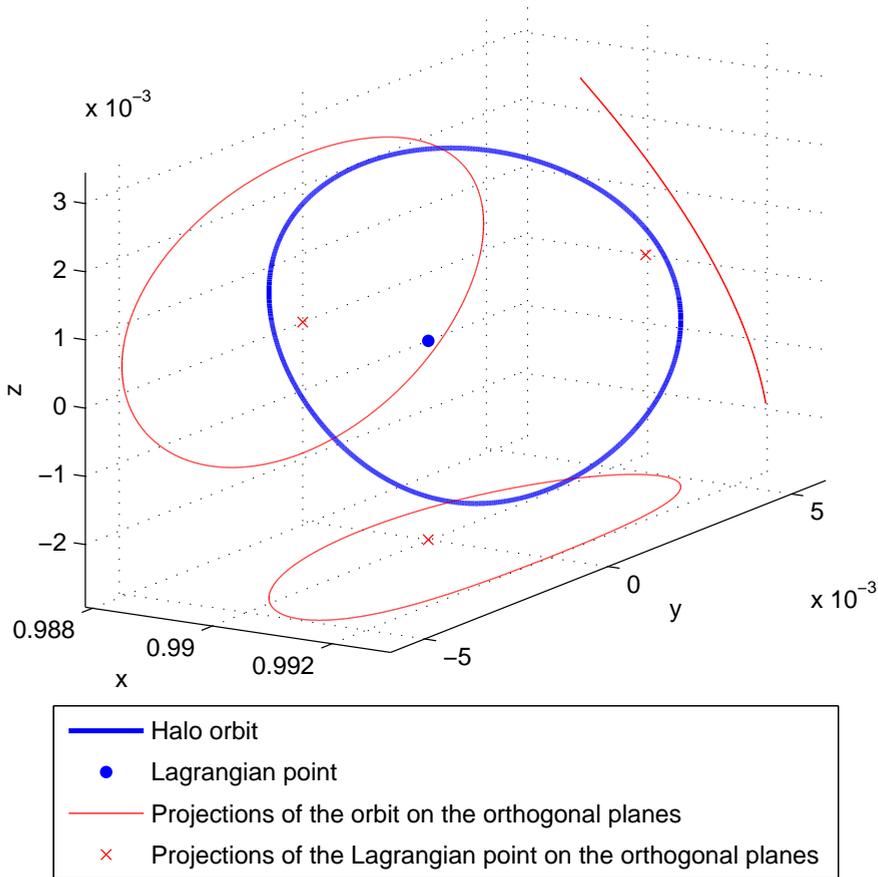


Figure 5.4: Halo orbit around the L_1 point of the Sun-Earth system (axes in non-dimensional units).

We discretized the orbit in 200 different points in order to obtain numerically an LTP model. We consider a formation of spacecraft in a 9-element threefold symmetric Golay array [32] that might be a reasonable choice for a synthetic aperture space telescope. The shape of the formation is shown in Figure 5.5, where the dashed lines indicate the allowed communication links. Constructing the pattern matrix \mathcal{P} related to this graph as a weighted normalized graph Laplacian, we have again that its real eigenvalues λ_i are such that $0 \leq \lambda_i \leq 2$.

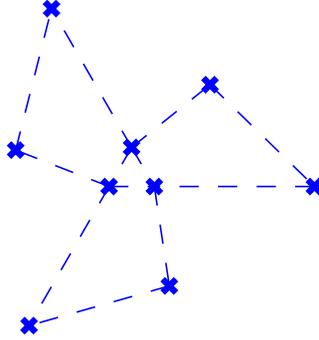


Figure 5.5: Communication structure of the graph representing a 9-element Golay formation.

We also assume that the spacecraft are all following the same orbit close to each other, so the same dynamic law holds for all of them.

Figure 5.6 shows the results of the simulation, where it can be seen how the spacecraft from a random initial position eventually get to recover the correct shape. We stress again that communication is needed only among nearest neighbors in the formation: the controllers exchange their states and measurements only with the closest spacecraft. Figure 5.7 shows the errors in the relative distances between the spacecraft. The results of this simulation are to be taken mainly qualitatively and they show the efficacy of the control in stabilizing the formation in a complex situation like an unstable orbit around a Lagrangian point.

5.5 Conclusions

In this chapter we have shown that the decomposition approach to control that is the object of this thesis can be effectively applied to spacecraft formations in periodic orbits. We have shown by means of examples that the new method makes it possible to have controllers with a performance that is quite close to the centralized theoretical \mathcal{H}_∞ optimum and at the same time to allow flexibility and a distributed architecture.

The method is based on LMI tools for control synthesis, so it can be extended to accommodate other requirements that are managed by such methods, for example robustness issues with respect to model uncertainties. Another possibility is to use multiobjective optimization [80], that would allow minimizing the propellant consumption while keeping the positioning accuracy within certain bounds.

In this work we have shown only applications to position control of the spacecraft in the formation, but it is possible to extend the use of the controller synthesis method shown here also to the station keeping of the relative attitude, if a linearized dynamics around the nominal position is used.

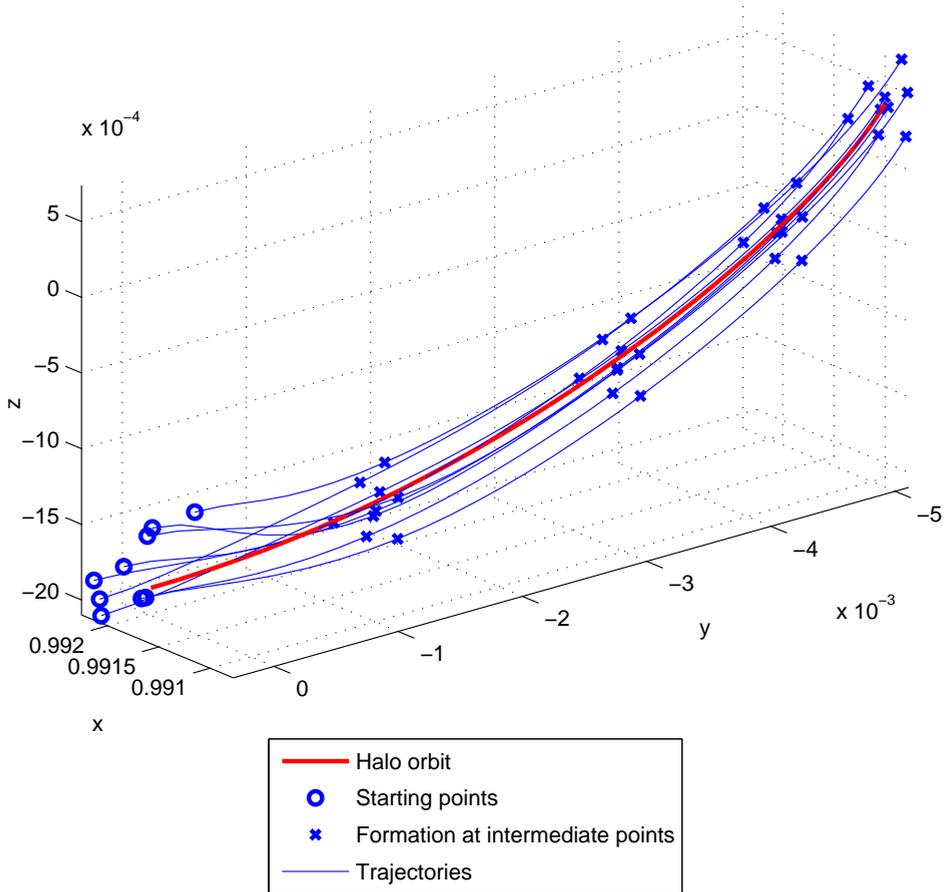


Figure 5.6: Motion of a controlled formation of spacecraft on an L_1 halo orbit (only a part of it is shown). The relative distances between the spacecraft have been exaggerated in order to make them visible (figure is not to scale).

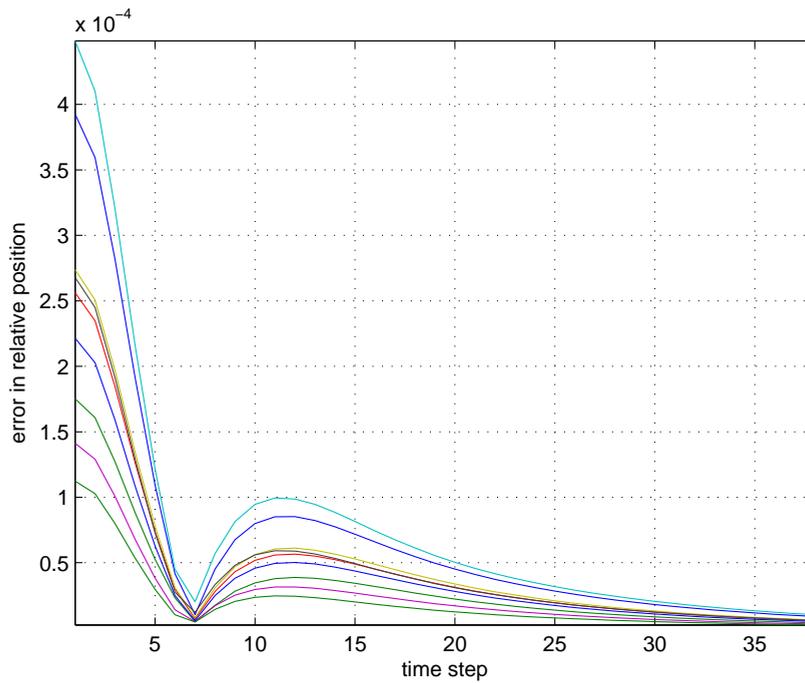


Figure 5.7: Errors in the relative distances between the spacecraft in the halo orbit formation. The shape of the curve is explained by the presence of an overshoot.

Identification

After considering the problem of distributed control, we now move to the complementary problem: identification of “distributed” models. The controller synthesis methods shown in Chapter 3 require the models to have the structure of a decomposable system. If such models are not available from first principles, is it still possible to identify them from input/output data?

We first approach the problem for a different kind of class, namely for the class of “circulant systems” that was briefly introduced in Section 2.6.1. Afterwards, we will look at the problem of identifying decomposable system models, and we will show that the solution is not as simple as for the circulant case, and it will require an additional step featuring a nonconvex optimization.

6.1 Introduction

The system and control theory community has dedicated recently a lot of attention to the study of distributed or large scale systems and their properties; this thesis is one of the results of such effort. The literature focuses mainly on the problem of distributed control design, as we too have done so far. From now on, in this last chapter, we will instead consider a different problem: identification. The distributed control techniques offer a wealth of methods that can be applied to special classes of large scale systems, but is there a way to get models fitting in these classes out of data?

In this chapter we are going to look for answers to this last question. It is clear that when “distributed” models of systems are not available from first principles, it is important to have a way to find such models from measured data, in order to be able to employ the distributed control techniques. To our knowledge the problem has not been fully explored in the literature yet. Of course, in order to be able to identify a system as a decomposable system, we will need to have the prior

knowledge of the fact that the system can be described with such a model, and we will also need to postulate a pattern matrix in advance. This prior knowledge can typically derive from the inspection of the physical system that we want to model; look again at Figure 2.2, decomposable systems can be imagined as the result of the interconnection of a set of identical systems, which interact with each other following a pattern. If the signals that the subsystems use to interact are measurable, then it is possible to identify each single subsystem including such signals into the input/output set; but if these signals cannot be measured, obtaining a structured model from the global input/output data alone is not a trivial task. Also notice that if we can measure the signals that the subsystems exchange, but it is not possible to separate a single subsystem from the others, then we would need to use identification methods that are suitable for closed-loop identification, as each element can be considered to be in closed-loop with the rest of the system.

Before analyzing the general problem of identifying decomposable systems, we first focus on the identification of a special subclass of them (in the wide sense): circulant systems [19]. Circulant systems represent a very special situation, for which the identification algorithm will turn out to be simpler with respect to the general case: for this reason, we consider the circulant case as a starting point. Later on, we will move to the identification of decomposable systems in general, which will need an additional step involving a nonlinear optimization. In the end, we will make some considerations on when this step might not be necessary, also for systems other than circulant.

This chapter is structured as follows. Section 6.2 concerns the identification of circulant systems whereas Section 6.3 concerns the identification of decomposable systems in general. Both these two sections contain identification algorithms and simulated examples of the use of such algorithms in practice. The conclusions are in Section 6.5.

The results contained in this chapter have been published in [58; 60].

6.2 Identification of circulant systems

Circulant systems, introduced briefly in Section 2.6.1, represent the best example of systems for which a customized identification algorithm based on decomposition can be developed. We start by presenting more in detail the definition and the properties of such systems, and subsequently we will present a possible identification algorithm based on subspace identification.

6.2.1 Circulant systems and their properties

We now show the basic concepts that are needed for introducing the notion of a circulant system. We first recall the concept of permutation matrix (introduced in Section 2.6.1).

Definition 6.1 (permutation matrix) *The circulant shift permutation matrix of order $N > 1$ is defined as:*

$$\Pi_N = \left[\begin{array}{c|c} 0 & I_{N-1} \\ \hline 1 & 0 \end{array} \right] \quad (6.1)$$

Notice that Π_N is orthogonal ($\Pi_N^{-1} = \Pi_N^T$). Right-multiplying an $N \times N$ matrix by Π_N is equivalent to cyclically shifting all its columns of one position to the right. Left-multiplication instead cyclically shifts the rows up.

Definition 6.2 (circulant matrix) *A square matrix E of size $N \times N$ is called ‘‘circulant’’ if and only if it has the following structure:*

$$E = \begin{bmatrix} e_1 & e_2 & e_3 & e_4 & \cdots & e_N \\ e_N & e_1 & e_2 & e_3 & \cdots & e_{N-1} \\ e_{N-1} & e_N & e_1 & e_2 & \cdots & e_{N-2} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ e_2 & e_3 & e_4 & e_5 & \cdots & e_1 \end{bmatrix} \quad (6.2)$$

where $e_i \in \mathbb{R}$ or $e_i \in \mathbb{C}$. This is the same as saying, a square matrix is circulant if and only if each row is obtained from the preceding one by a cyclic shift of one position to the right.

This definition is equivalent to saying that a circulant matrix is invariant under a similarity transformation with respect to Π_N : $E = \Pi_N^{-1} E \Pi_N = \Pi_N^T E \Pi_N$.

Definition 6.3 (block circulant matrix) *A block circulant matrix E of order N is a (not necessarily square) matrix with the following block structure:*

$$E = \begin{bmatrix} E_1 & E_2 & E_3 & E_4 & \cdots & E_N \\ E_N & E_1 & E_2 & E_3 & \cdots & E_{N-1} \\ E_{N-1} & E_N & E_1 & E_2 & \cdots & E_{N-2} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ E_2 & E_3 & E_4 & E_5 & \cdots & E_1 \end{bmatrix} \quad (6.3)$$

where $E_i \in \mathbb{R}^{p \times q}$ or $E_i \in \mathbb{C}^{p \times q}$, with p, q positive integers.

It is immediate to see [17] that such a matrix can also be written as:

$$E = \sum_{i=1}^N (\Pi_N^{i-1} \otimes E_i). \quad (6.4)$$

Let us now introduce some new notation. We will denote the set of block circulant matrices of order N , with blocks of size $p \times q$, as $\mathcal{C}_{N,p,q}$; we will use the symbol $\mathcal{C}_{N,p,q}^{\mathbb{R}}$ or $\mathcal{C}_{N,p,q}^{\mathbb{C}}$ if we want to specify that the entries of such matrices are respectively real or complex. Let $\mathcal{D}_{N,p,q}$ instead denote the set of block diagonal matrices

with N block rows and block columns, and blocks of size $p \times q$. Again, we will use either $\mathcal{D}_{N,p,q}^{\mathbb{R}}$ or $\mathcal{D}_{N,p,q}^{\mathbb{C}}$ if we want to emphasize the nature of the entries of such matrices. For a matrix $E \in \mathcal{D}_{N,p,q}$, E_i will indicate the i^{th} block on the diagonal; for a matrix $E \in \mathcal{C}_{N,p,q}$, E_i indicates the i^{th} block in the first row (as shown in Definition 6.3).

Remark 6.1 *The sums and products of block circulant matrices of the same order are still block circulant. The inverse of a square invertible block circulant matrix is block circulant [17].*

Lemma 6.1 (block-permutation) *A block circulant matrix $E \in \mathcal{C}_{N,p,q}$ is invariant under a block-permutation transformation, which means:*

$$(\Pi_N \otimes I_p)^{-1} E (\Pi_N \otimes I_q) = (\Pi_N^{-1} \otimes I_p) E (\Pi_N \otimes I_q) = E. \quad (6.5)$$

Proof: From (6.4), we have:

$$(\Pi_N \otimes I_p)^{-1} E (\Pi_N \otimes I_q) = \sum_{i=1}^N (\Pi_N \otimes I_p)^{-1} (\Pi_N^{i-1} \otimes E_i) (\Pi_N \otimes I_q). \quad (6.6)$$

From the properties of the Kronecker product [12] it follows:

$$(\Pi_N \otimes I_p)^{-1} E (\Pi_N \otimes I_q) = \sum_{i=1}^N (\Pi_N^{-1} \Pi_N^{i-1} \Pi_N \otimes I_p E_i I_q) = \sum_{i=1}^N (\Pi_N^{i-1} \otimes E_i) = E. \quad (6.7)$$

□

Definition 6.4 (Fourier matrix) *We define the Fourier matrix of order N as:*

$$F_N = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & w_N & w_N^2 & \cdots & w_N^{(N-1)} \\ 1 & w_N^2 & w_N^4 & \cdots & w_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w_N^{(N-1)} & w_N^{2(N-1)} & \cdots & w_N^{(N-1)(N-1)} \end{bmatrix} \quad (6.8)$$

with $w_N = e^{-\frac{2\pi j}{N}} = \cos \frac{2\pi}{N} - j \sin \frac{2\pi}{N}$.

The matrix F_N is unitary and symmetric: $F_N^H F_N = F_N F_N^H = I_N$, $F_N^T = F_N$. Left-multiplying a column vector with a Fourier matrix is equivalent to computing its Discrete Fourier Transform (DFT); for large values of N , it is convenient to use the Fast Fourier Transform (FFT) algorithm instead of computing the matrix product [17].

We call f_i the i^{th} row of F_N . We will show now that all the rows but the first of the Fourier matrix are complex conjugate between each other; if N is even, then

f_1 and $f_{N/2}$ are real, while the other rows form complex conjugate pairs; if N is odd, then f_1 alone is real with the other rows forming complex conjugate pairs.

Lemma 6.2 *The rows of F_N are either real or in complex conjugate pairs according to the relation: $f_{N+2-i} = \bar{f}_i$ for $i = 2, \dots, N$.*

Proof: *The first row f_1 is trivially always real. For what concerns the other rows, we can see that the k^{th} element of f_{N+2-i} is: $f_{N+2-i,k} = e^{-\frac{2\pi j}{N}(N+1-i)(k-1)}$, while the k^{th} element of f_i is: $f_{i,k} = e^{-\frac{2\pi j}{N}(i-1)(k-1)}$. From the properties of the complex exponential ($e^{\bar{z}} = \overline{e^z}$, $e^{z+2\pi jk} = e^z$ for $k \in \mathbb{Z}$) then we have:*

$$f_{N+2-i,k} = e^{-\frac{2\pi j}{N}(-i+1)(k-1)} = \overline{e^{-\frac{2\pi j}{N}(i-1)(k-1)}} = \bar{f}_{i,k}. \quad (6.9)$$

This implies $f_{N+2-i} = \bar{f}_i$. □

Fourier matrices have the remarkable property of diagonalizing any circulant matrix. This property is crucial because it allows decomposing circulant systems to smaller independent ones, thus reducing the complexity of the identification problem. The property is stated in the theorem that follows, and then generalized to block circulant matrices.

Theorem 6.1 (diagonalization property) *For any matrix $E \in \mathbb{C}^{N \times N}$, it holds that $F_N E F_N^H$ is a diagonal matrix if and only if E is circulant.*

Proof: *The proof can be found in [17].* □

Corollary 6.1 *Consider a matrix $E \in \mathbb{C}^{Np \times Nq}$. Then $\mathbf{E} = (F_N \otimes I_p) E (F_N \otimes I_q)^H \in \mathcal{D}_{N,p,q}^{\mathbb{C}}$ if and only if $E \in \mathcal{C}_{N,p,q}^{\mathbb{C}}$.*

Proof: *We start with the “if” part. If E is block circulant, then from (6.4) we have:*

$$\mathbf{E} = (F_N \otimes I_r) \left(\sum_{i=1}^N \Pi_N^{i-1} \otimes E_i \right) (F_N \otimes I_m)^H. \quad (6.10)$$

From the properties of the Kronecker product (see Section 2.3) then we have:

$$\mathbf{E} = \sum_{i=1}^N (F_N \Pi_N^{i-1} F_N^H) \otimes E_i. \quad (6.11)$$

Notice that Π_N^{i-1} is circulant, so $F_N \Pi_N^{i-1} F_N^H$ is diagonal (Theorem 6.1). Then we have that $(F_N \Pi_N^{i-1} F_N^H) \otimes E_i$ is block diagonal, and \mathbf{E} is a sum of block diagonal matrices, so it is block diagonal too. This proves the “if” part. The “only if” part is the equivalent of saying that for a matrix $G \in \mathbb{C}^{Np \times Nq}$, $(F_N \otimes I_r)^H G (F_N \otimes I_r) \in \mathcal{C}_{N,p,q}^{\mathbb{C}}$ if $G \in \mathcal{D}_{N,p,q}^{\mathbb{C}}$ (just assume that $G = (F_N \otimes I_p) E (F_N \otimes I_q)^H \in \mathcal{D}_{N,p,q}^{\mathbb{C}}$). We can see that being G block diagonal, it holds that:

$$(F_N \otimes I_p)^H G (F_N \otimes I_q) = \sum_{i=1}^N (f_i^H \otimes I_p) G_i (f_i \otimes I_q). \quad (6.12)$$

Then thanks to (2.15) (consider that $G_i = 1 \otimes G_i$), we have:

$$(F_N \otimes I_p)^H G (F_N \otimes I_q) = \sum_{i=1}^N (f_i^H f_i \otimes G_i). \quad (6.13)$$

Notice that $f_i^H f_i = F_N^H H_i F_N$, where H_i is a matrix with all entries equal to 0 but the i^{th} entry on the diagonal that is 1. So H_i is diagonal and thanks to Theorem 6.1, $f_i^H f_i$ is circulant. Then $(f_i^H f_i \otimes G_i)$ is block circulant, and $(F_N \otimes I_p)^H G (F_N \otimes I_q)$ being a sum of block circulant matrices is block circulant as well. \square

It is possible to show that the complex block diagonal matrices obtained through the transformation via Fourier matrices from real block circulant matrices have some special features; and all of the block diagonal matrices of such kind can be transformed into real block circulant ones with the inverse transformation.

Corollary 6.2 For a matrix $E \in \mathcal{C}_{N,p,q}^{\mathbb{R}}$, then for $\mathbf{E} = (F_N \otimes I_p) E (F_N \otimes I_q)^H \in \mathcal{D}_{N,p,q}^{\mathbb{C}}$ it holds that $\mathbf{E}_1 \in \mathbb{R}^{p \times q}$ and $\mathbf{E}_{N+2-i} = \bar{\mathbf{E}}_i$ for $i = 2, \dots, N$.

Conversely, for a matrix $G \in \mathcal{D}_{N,p,q}^{\mathbb{C}}$ for which $G_1 \in \mathbb{R}^{p \times q}$ and $G_{N+2-i} = \bar{G}_i$ for $i = 2, \dots, N$, we have that $(F_N \otimes I_p)^H G (F_N \otimes I_q) \in \mathcal{C}_{N,p,q}^{\mathbb{R}}$.

Proof: It is a consequence of Lemma 6.2. \square

We are now ready to introduce the notion of circulant system and show its key features. After the definition, we will first state a property that characterizes such kind of systems, and then we will show how they can be decomposed into smaller independent systems, thus enabling efficient solutions to the identification problem.

Definition 6.5 (circulant systems) Consider a discrete-time MIMO system with Nm inputs and Nr outputs, which can be described by state-space equations of the kind:

$$\begin{cases} x(k+1) = \mathcal{A}x(k) + \mathcal{B}u(k) \\ y(k) = \mathcal{C}x(k) + \mathcal{D}u(k) \end{cases} \quad (6.14)$$

with $\mathcal{A} \in \mathbb{R}^{Nl \times Nl}$, $\mathcal{B} \in \mathbb{R}^{Nl \times Nm}$, $\mathcal{C} \in \mathbb{R}^{Nr \times Nl}$, $\mathcal{D} \in \mathbb{R}^{Nr \times Nm}$. The vector $x \in \mathbb{R}^{Nl \times 1}$ is the state, $u \in \mathbb{R}^{Nm \times 1}$ is the input signal and $y \in \mathbb{R}^{Nr \times 1}$ is the output signal. We call the system "circulant" (or block circulant) if and only if it has a representation with $\mathcal{A} \in \mathcal{C}_{N,l,l}^{\mathbb{R}}$, $\mathcal{B} \in \mathcal{C}_{N,l,m}^{\mathbb{R}}$, $\mathcal{C} \in \mathcal{C}_{N,r,l}^{\mathbb{R}}$ and $\mathcal{D} \in \mathcal{C}_{N,r,m}^{\mathbb{R}}$. When we will refer to the matrices of a circulant system, we will consider only this realization with circulant matrices.

We consider also the input u to be made of N blocks of size $m \times 1$, which we denote as u_i , and the output y to be made of N blocks of size $r \times 1$, which we denote as y_i ($i = 1, \dots, N$). We call these blocks "local inputs" and "local outputs".

If a state-space realization of a system is given, such a system is a circulant system if and only if there exists a similarity transformation that turns the state-space matrices into block circulant ones.

Each of these subsystems has only m inputs and r outputs.

Proof: According to Corollary 6.1, it holds that:

$$\begin{aligned}\mathcal{A} &= (F_N \otimes I_l)^H \mathbf{A} (F_N \otimes I_l) \\ \mathcal{B} &= (F_N \otimes I_l)^H \mathbf{B} (F_N \otimes I_m) \\ \mathcal{C} &= (F_N \otimes I_r)^H \mathbf{C} (F_N \otimes I_l) \\ \mathcal{D} &= (F_N \otimes I_r)^H \mathbf{D} (F_N \otimes I_m)\end{aligned}\quad (6.17)$$

with $\mathbf{A} \in \mathcal{D}_{N,l,l}^{\mathbb{C}}$, $\mathbf{B} \in \mathcal{D}_{N,l,m}^{\mathbb{C}}$, $\mathbf{C} \in \mathcal{D}_{N,r,l}^{\mathbb{C}}$, $\mathbf{D} \in \mathcal{D}_{N,r,m}^{\mathbb{C}}$. So we can rewrite equation (6.14) as:

$$\begin{cases} x(k+1) = (F_N \otimes I_l)^H \mathbf{A} (F_N \otimes I_l) x(k) + (F_N \otimes I_l)^H \mathbf{B} (F_N \otimes I_m) u(k) \\ y(k) = (F_N \otimes I_r)^H \mathbf{C} (F_N \otimes I_l) x(k) + (F_N \otimes I_r)^H \mathbf{D} (F_N \otimes I_m) u(k) \end{cases}\quad (6.18)$$

\Leftrightarrow

$$\begin{cases} (F_N \otimes I_l) x(k+1) = \mathbf{A} (F_N \otimes I_l) x(k) + \mathbf{B} (F_N \otimes I_m) u(k) \\ (F_N \otimes I_r) y(k) = \mathbf{C} (F_N \otimes I_l) x(k) + \mathbf{D} (F_N \otimes I_m) u(k) \end{cases}\quad (6.19)$$

If we apply the following invertible transformations for state, input and output:

$$\begin{aligned}\hat{x}(k) &= (F_N \otimes I_l) x(k) \\ \hat{u}(k) &= (F_N \otimes I_m) u(k) \\ \hat{y}(k) &= (F_N \otimes I_r) y(k)\end{aligned}\quad (6.20)$$

then the system turns into:

$$\begin{cases} \hat{x}(k+1) = \mathbf{A} \hat{x}(k) + \mathbf{B} \hat{u}(k) \\ \hat{y}(k) = \mathbf{C} \hat{x}(k) + \mathbf{D} \hat{u}(k) \end{cases}\quad (6.21)$$

All the matrices involved in this systems are block diagonal, so this system is equivalent to the following N independent l^{th} order subsystems (of complex variables), each of them with m inputs and r outputs:

$$\begin{cases} \hat{x}_i(k+1) = \mathbf{A}_i \hat{x}_i(k) + \mathbf{B}_i \hat{u}_i(k) \\ \hat{y}_i(k) = \mathbf{C}_i \hat{x}_i(k) + \mathbf{D}_i \hat{u}_i(k) \end{cases} \quad \text{for } i = 1, \dots, N \quad (6.22)$$

where \mathbf{A}_i , \mathbf{B}_i , \mathbf{C}_i and \mathbf{D}_i are respectively the blocks in the diagonal of \mathbf{A} , \mathbf{B} , \mathbf{C} and \mathbf{D} , and $\hat{x}_i(k)$, $\hat{u}_i(k)$ and $\hat{y}_i(k)$ are the blocks of the column vectors $\hat{x}(k)$, $\hat{u}(k)$ and $\hat{y}(k)$; $\mathbf{A}_i \in \mathbb{C}^{l \times l}$, $\mathbf{B}_i \in \mathbb{C}^{l \times m}$, $\mathbf{C}_i \in \mathbb{C}^{r \times l}$, $\mathbf{D}_i \in \mathbb{C}^{r \times m}$, $\hat{x}_i(k) \in \mathbb{C}^{l \times 1}$, $\hat{u}_i(k) \in \mathbb{C}^{m \times 1}$ and $\hat{y}_i(k) \in \mathbb{C}^{r \times 1}$. \square

It is important to point out that not all the N modal subsystems of (6.22) are independent; actually, as a direct consequence of Corollary 6.2, the systems of index $N+2-i$ are the complex conjugate version of the systems of index i , for $i = 2, \dots, N$. So there are only $N/2 + 1$ independent systems if N is even and $(N+1)/2$ independent systems if N is odd.

Corollary 6.3 With respect to (6.22), let P_i indicate any among the following: \mathbf{A}_i , \mathbf{B}_i ,

$C_i, D_i, \hat{x}_i(k), \hat{u}_i(k)$ and $\hat{y}_i(k)$. It holds that:

$$\begin{aligned} P_1 \text{ is real} \\ P_{N+2-i} = \bar{P}_i \text{ for } i = 2, \dots, N \end{aligned} \quad (6.23)$$

Proof: It is a consequence of Lemma 6.2. □

6.2.2 The identification algorithm

Motivation and rationale

As a consequence of Lemma 6.3, we have seen that there exist categories of systems which can be identified as circulant just from physical insight, as a result of the invariance of their input and output pairs to shifts. As an example, consider the system shown in Figure 6.1: the global system is made of four smaller identical subsystems, each with its input and output, connected in a circular way. The interconnections between neighboring systems are all the same, and actually it is impossible to distinguish one system from the other.

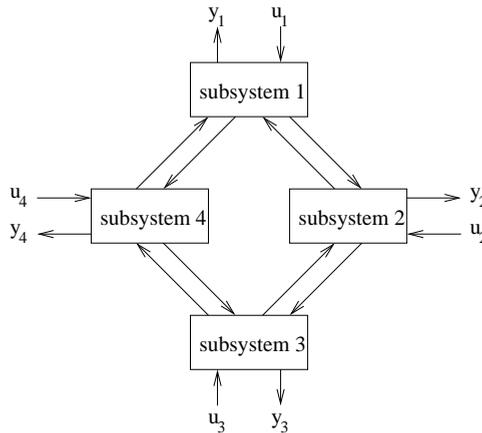


Figure 6.1: Example of circulant system made of 4 identical subsystems.

In such a situation, putting Subsystem 1 in the place of Subsystem 2, Subsystem 2 in the place of Subsystem 3, Subsystem 3 in the place of Subsystem 4, and Subsystem 4 in the place of Subsystem 1 would still yield the same global system. Then we know that the invariance to shift of input/output pairs of Lemma 6.3 must hold, as it is impossible to know whether we are looking at the original system or its shifted (or “rotated”) version. An example of systems of this kind can be found in adaptive optics; we can consider again for example the picture of a deformable mirror in Figure 4.1, under a different point of view. The mirror is made of a set of actuators placed on a regular grid which can displace the reflecting surface. Such device is invariant under rotations of multiples of 60 degrees, and it

can be seen as the interconnection of six identical sectors, which influence each other strongly. Another kind of circulant system could be a system made of two identical interconnected parts, which could simply be a physical object with one plane of symmetry.

So there might be the necessity of identifying such kind of systems from data. Subspace methods [88; 91] are the most common choice for linear MIMO systems, and they could be used in a situation as this to identify a discrete-time state-space model of the global system, from the set of all outputs and all inputs. The problem of this approach is in the fact that subspace methods return state-space matrices up to an arbitrary similarity transformation, that disrupts any structure the system may have. Moreover, we will also demonstrate that it can be useful to force the circulant structure to the model in order to improve the accuracy of the estimation: the knowledge of the symmetries of the system can be used as an a priori information on the MIMO model.

We will shortly show that it is indeed possible to exploit the structure of circulant systems for identification; in fact, we will illustrate an identification algorithm that:

1. allows using the prior knowledge of the system as circulant;
2. reduces the computational complexity of the problem;
3. preserves the circulant structure, that is, the identified model is again a circulant system.

The algorithm is a direct consequence of the diagonalization property of circulant systems (Theorem 6.2) and it can be outlined as follows. As the system can be turned into N independent subsystems, and as for each of these subsystem we can find a priori which are the inputs and outputs, then it is possible to identify each of these modal subsystems separately from each other. For this purpose, it is sufficient to transform the inputs and the outputs as in (6.20), and use them with any method (subspace identification, prediction error, etc. [51]) to identify the state-space matrices of the modal systems; the only additional care we will need to take is that we should extend the method to models with complex values. Actually not all the N subsystems have to be identified, but only the free ones, while the others are just the complex conjugates as explained in Corollary 6.3. Then, once these systems have been identified, the global model can be retrieved with the use of (6.17). Corollary 6.1 will grant that the global matrices obtained are block circulant, while Corollary 6.2 will grant that such matrices have real entries.

We said in the previous paragraph that any method can be used for identifying the modal subsystems; actually, subspace methods seem to be the best choice at this point, as they are inherently fit to deal with state-space models (instead of transfer functions) and they can naturally be extended to the complex domain (as it was done in e.g. [62], for a different purpose). The subspace identification process is a “numerical recipe” that yields four matrices as result of an input/output couple; all the algebraic operations used in subspace identification (matrix sum, matrix product, singular value decomposition or QR factorization) can be extended to

complex numbers. Moreover, subspace methods will offer insight into the order l of the subsystems from the singular values of the extended observability matrices (see [92] for details). This will make it possible to choose a good value of the order: although the different subsystems may yield different results, it is necessary to choose the same order l for all of them. For these reasons, in the sequel we will use a subspace algorithm, specifically the MOESP (Multi-variable Output-Error State sPace) algorithm [91]. MOESP is fit for systems with white measurement noise only, and in the examples here we will restrict to them, but of course the idea of circulant identification can be extended to more sophisticated subspace methods (e.g. PI-MOESP, PO-MOESP [92], N4SID [88]) that take into account different models of noise. The characteristics of the noise are not changed by the decoupling transformation, so MOESP will give an unbiased estimate if the noise in the untransformed signals is white.

Now we are almost ready to write the algorithm explicitly in its steps. But first, we discuss the condition of “persistence of excitation” that is necessary for the identification process.

Persistence of excitation

The persistence of excitation is a requirement that is put on input signals in order to make system identification possible.

Definition 6.6 (persistence of excitation [92]) *The signal $u(k)$, for $k = 0, 1, \dots$ is persistently exciting of order N if and only if there exists an integer M such that the Hankel matrix of the input:*

$$U_{0,N,M} = \begin{bmatrix} u(0) & u(1) & \cdots & u(M-1) \\ u(1) & u(2) & \cdots & u(M) \\ \vdots & & \ddots & \vdots \\ u(N-1) & u(N) & \cdots & u(M+N-2) \end{bmatrix} \quad (6.24)$$

has full row rank.

If we use a subspace method for identifying an N^{th} order model, then it is necessary to have an input which is persistently exciting of at least order N [94]. In case we want to identify a circulant system of order Nl , then we would expect that the input has to be at least persistently exciting of order Nl . Actually, this is not really necessary, as we identify the modal subsystems and not the full model itself. So for the subsystems we only need that the \hat{u}_i which is involved is persistently exciting of order l , which is less restrictive. The following lemma and an example will give more insight into the issue of persistence of excitation for circulant systems.

Lemma 6.4 *If the full input signal $u(k)$ of a circulant system (equation (6.14)) is persistently exciting of order s , then each one of the “modal” input signals $\hat{u}_i(k)$ obtained through (6.20) is persistently exciting of order s .*

Proof: If the input is persistently exciting of order s , then there exists an M for which $U_{0,s,M}$ has full row rank. If we consider the Hankel matrix $\hat{U}_{0,s,M}$ of the transformed input \hat{u} , it is easy to see that it is related to $U_{0,s,M}$ by the relation:

$$\hat{U}_{0,s,M} = (I_s \otimes F_N \otimes I_m)U_{0,s,M}. \quad (6.25)$$

As $I_s \otimes F_N \otimes I_m$ is full rank, then, thanks to Sylvester's inequality [92], if $U_{0,s,M}$ is full row rank then $\hat{U}_{0,s,M}$ is full row rank as well. We call $\hat{U}_{0,s,M}^i$ for $i = 1, \dots, N$ the Hankel matrices obtained from the single "modal" inputs \hat{u}_i . All of these matrices are submatrices of $\hat{U}_{0,s,M}$ that is full rank, and so they are all full rank ($\hat{U}_{0,s,M}^i$ is made of the rows of $\hat{U}_{0,s,M}$ containing \hat{u}_i). So each signal $\hat{u}_i(k)$ is persistently exciting of order s . \square

The relevant consequence of this lemma is that we do not actually need a signal with persistency of excitation of order Nl in order to identify a circulant system of the same order, but l is enough, as with it we can identify the modal systems which are of order l . And besides this, there could be signals which are not even persistently exciting of order l for the full system, but are so for the modal systems once transformed.

Consider for example a situation where all the N local inputs are equal to zero, but one (let us assume it is u_1). Also assume that u_1 alone would be persistently exciting of order l , meaning that the Hankel matrix $U_{0,s,M}^1$ built with u_1 is full rank. Instead $U_{0,s,M}$ will never be full rank as it contains some null rows, making the full signal u persistently exciting of order 0. So it will not be possible to identify the full system without making any assumptions on its structure. But the modal systems can be identified, as the matrices $\hat{U}_{0,s,M}^i$ will be all full rank; in fact $\hat{U}_{0,s,M}^i = U_{0,s,M}^1 / \sqrt{N}$. This example shows that the knowledge on the structure of the system makes the identification possible for a much wider class of input signals, and that it is not even necessary to put an input in all the physical subsystems, but a single input channel can be enough.

The novel algorithm

We are then ready to propose a solution for the following problem:

Problem Description 6.1 (Circulant system identification)

Consider a given set of N input signals $u_i(k) \in \mathbb{R}^{m \times 1}$ and N output signals $y_i(k) \in \mathbb{R}^{m \times 1}$, for $i = 1, \dots, N$ and $k = 1, \dots, k_{\max}$. This set of data is associated to a dynamical system; we know, thanks to considerations stemming from Lemma 6.3, that this system has a circulant structure and that we can use a model of circulant system according to Definition 6.5 to describe it, where N , m and r are already known and l is unknown. Goal: identify an lN^{th} order state-space circulant model from input/output data.

The solution is reported in the form of an algorithm.

Algorithm 6.1 (subspace identification of circulant systems)

Problem 6.1 is solved in the following steps:

1. Compute the Fourier matrix F_N of order N .
2. Transform input and output signals, by computing:

$$\hat{u}(k) = (F_N \otimes I_m)u(k), \quad \hat{y}(k) = (F_N \otimes I_r)y(k). \quad (6.30)$$

3. Verify that each signal $\hat{u}(k)$ is persistently exciting of at least order l .
4. Use MOESP to identify independent state-space models of order l from each \hat{u}_i/\hat{y}_i pair:

$$\begin{cases} \hat{x}_i(k+1) = \hat{\mathbf{A}}_i \hat{x}_i(k) + \hat{\mathbf{B}}_i \hat{u}_i(k) \\ \hat{y}_i(k) = \hat{\mathbf{C}}_i \hat{x}_i(k) + \hat{\mathbf{D}}_i \hat{u}_i(k) \end{cases} \quad \text{for } i = 1, \dots, N \quad (6.31)$$

If N is even, then identify the systems for $i = 1, \dots, N/2$; if N is odd instead, identify the systems for $i = 1, \dots, (N+1)/2$ (the other values of i correspond to signals which are just the complex conjugates, so they do not contain further information): the method will yield as results the identified (complex) matrices $\hat{\mathbf{A}}_i$, $\hat{\mathbf{B}}_i$, $\hat{\mathbf{C}}_i$ and $\hat{\mathbf{D}}_i$. Then use Corollary 6.3 to get the matrices of the other (dependent) systems:

$$\begin{aligned} \hat{\mathbf{A}}_i &= \overline{\hat{\mathbf{A}}_{N+2-i}} \\ \hat{\mathbf{B}}_i &= \overline{\hat{\mathbf{B}}_{N+2-i}} \\ \hat{\mathbf{C}}_i &= \overline{\hat{\mathbf{C}}_{N+2-i}} \\ \hat{\mathbf{D}}_i &= \overline{\hat{\mathbf{D}}_{N+2-i}} \end{aligned} \quad \text{for } i = \begin{cases} \frac{N}{2} + 1, \dots, N & \text{if } N \text{ even} \\ \frac{N+1}{2} + 1, \dots, N & \text{if } N \text{ odd} \end{cases} \quad (6.32)$$

5. Construct the block diagonal matrices: $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, $\hat{\mathbf{C}}$ and $\hat{\mathbf{D}}$ putting the identified blocks together.
6. Retrieve the global system matrices with the following formulas:

$$\begin{aligned} \hat{\mathbf{A}} &= (F_N \otimes I_l)^H \hat{\mathbf{A}} (F_N \otimes I_l) \\ \hat{\mathbf{B}} &= (F_N \otimes I_l)^H \hat{\mathbf{B}} (F_N \otimes I_m) \\ \hat{\mathbf{C}} &= (F_N \otimes I_r)^H \hat{\mathbf{C}} (F_N \otimes I_l) \\ \hat{\mathbf{D}} &= (F_N \otimes I_r)^H \hat{\mathbf{D}} (F_N \otimes I_m) \end{aligned} \quad (6.33)$$

$\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, $\hat{\mathbf{C}}$ and $\hat{\mathbf{D}}$ are real and block circulant, thanks to Corollaries 6.3, 6.1 and 6.2. Notice that it is not necessary to compute the multiplications above fully, it is just enough to get the first block row of these matrices and then the others are known thanks to the circulant structure.

In addition, if desired, in most cases it is possible to find a transformation that block-diagonalizes either \hat{B} or \hat{C} (it could be useful to get more physical insight into the system). For example, we can diagonalize \hat{C} by finding the circulant matrix T :

$$T = \begin{bmatrix} I_l & T_2 & T_3 & \cdots & T_{N-1} \\ T_{N-1} & I_l & T_2 & \cdots & T_{N-2} \\ \vdots & \vdots & \vdots & & \vdots \\ T_2 & T_3 & T_4 & \cdots & I_l \end{bmatrix} \quad (6.34)$$

for which it holds:

$$\underbrace{\begin{bmatrix} C_1 & C_2 & C_3 & \cdots & C_{N-1} \\ C_{N-1} & C_1 & C_2 & \cdots & C_{N-2} \\ \vdots & \vdots & \vdots & & \vdots \\ C_2 & C_3 & C_4 & \cdots & C_1 \end{bmatrix}}_{\hat{c}} \begin{bmatrix} I_l & T_2 & T_3 & \cdots & T_{N-1} \\ T_{N-1} & I_l & T_2 & \cdots & T_{N-2} \\ \vdots & \vdots & \vdots & & \vdots \\ T_2 & T_3 & T_4 & \cdots & I_l \end{bmatrix} = \begin{bmatrix} C & 0 & \cdots & 0 \\ 0 & C & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C \end{bmatrix} \quad (6.35)$$

that is obtained by setting:

$$\underbrace{\begin{bmatrix} C_{N-1} & C_1 & C_2 & \cdots & C_{N-2} \\ \vdots & \vdots & \vdots & & \vdots \\ C_2 & C_3 & C_4 & \cdots & C_1 \end{bmatrix}}_{\hat{c} \text{ but the first row}} \begin{bmatrix} I_l \\ T_{N-1} \\ \vdots \end{bmatrix} = 0 \quad (6.36)$$

This allows finding the desired transformation:

$$\begin{bmatrix} T_{N-1} \\ \vdots \\ T_2 \end{bmatrix} = - \underbrace{\begin{bmatrix} C_1 & C_2 & \cdots & C_{N-2} \\ \vdots & \vdots & & \vdots \\ C_3 & C_4 & \cdots & C_1 \end{bmatrix}}_{\hat{c} \text{ but the first row and column}}^\dagger \begin{bmatrix} C_{N-1} \\ \vdots \\ C_2 \end{bmatrix} \quad (6.37)$$

where \dagger indicates the pseudoinverse, if the matrix subject to pseudoinversion has a left inverse.

Subspace algorithms deliver the system matrices up to a similarity transformation. We show in the next theorem that a set of independent similarity transformations of the N independent subsystems are of no concern for the final result. In fact, all the possible similarity transformations of the subsystems are always equivalent to a global similarity transformation for the complete system.

Theorem 6.3 *Let us assume that each of the subsystems described by (6.22) is known up to a similarity transformation with a nonsingular matrix T_i :*

$$\begin{cases} \hat{x}_i(k+1) = T_i^{-1} \mathbf{A}_i T_i \hat{x}_i(k) + T_i^{-1} \mathbf{B}_i \hat{u}_i(k) \\ \hat{y}_i(k) = \mathbf{C}_i T_i \hat{x}_i(k) + \mathbf{D}_i \hat{u}_i(k) \end{cases} \quad \text{for } i = 1, \dots, N. \quad (6.38)$$

Then this is equivalent to knowing the global circulant system up to a global similarity transformation.

Proof: If we use (6.33) to recover the global matrices from the ones of the subsystems, transformed by T_i , then we have (we neglect \hat{D} as it is not influenced by similarity transformations):

$$\begin{aligned} \hat{\mathbf{A}}_T &= (F_N \otimes I_l)^H T^{-1} \hat{\mathbf{A}} T (F_N \otimes I_l) \\ \hat{\mathbf{B}}_T &= (F_N \otimes I_l)^H T^{-1} \hat{\mathbf{B}} (F_N \otimes I_m) \\ \hat{\mathbf{C}}_T &= (F_N \otimes I_r)^H \hat{\mathbf{C}} T (F_N \otimes I_l) \end{aligned} \quad (6.39)$$

where $T \in \mathcal{D}_{N,l,l}$ is the block diagonal matrix containing all the T_i blocks; T^{-1} is block diagonal as well. We can rewrite the equations inserting some identity matrices in key points; for the first one, we have:

$$\hat{\mathbf{A}}_T = (F_N \otimes I_l)^H T^{-1} \underbrace{(F_N \otimes I_l)(F_N \otimes I_l)^H}_{I_{Nl}} \hat{\mathbf{A}} \underbrace{(F_N \otimes I_l)(F_N \otimes I_l)^H}_{I_{Nl}} T (F_N \otimes I_l). \quad (6.40)$$

Being T block diagonal, then $\mathcal{T} = (F_N \otimes I_l)^H T^{-1} (F_N \otimes I_l)$ is block-circulant (Corollary 6.1). So we have:

$$\hat{\mathbf{A}}_T = \mathcal{T} (F_N \otimes I_l)^H \hat{\mathbf{A}} (F_N \otimes I_l) T^{-1} = \mathcal{T} \hat{\mathbf{A}} T^{-1}. \quad (6.41)$$

In a similar way, we can show that:

$$\hat{\mathbf{B}}_T = \mathcal{T} \hat{\mathbf{B}}, \quad \hat{\mathbf{C}}_T = \hat{\mathbf{C}} T^{-1} \quad (6.42)$$

that is the same as saying, we know the system matrices up to a similarity transformation. \square

Remark 6.2 *As a last consideration, let us evaluate the reduction in complexity that is obtained by using the proposed algorithm instead of a global MOESP. The complexity of MOESP is determined by its most costly operation, that is the QR factorization; in this analysis, we limit ourselves to looking at this step. For a matrix in $\mathbb{R}^{j \times k}$, the cost of the QR factorization is $\mathcal{O}(jk^2)$ [33]. Application of MOESP to the complete Nm input and Nr output system, for M time steps, requires the QR factorization of a matrix with M rows and $sN(m+r)$ columns, where s is a number of choice that is bigger than the order (so $s = \mathcal{O}(Nl)$). This means that the cost of the QR for the global MOESP is $\mathcal{O}(Ms^2N^2(m+r)^2) \approx \mathcal{O}(M(m+r)^2N^4l^2)$. If we use the circulant MOESP, we need to perform N times the QR decomposition of $N \times s(m+r)$ matrices, where now $s = \mathcal{O}(l)$. This means that the global cost of the QRs for the circulant MOESP is $\mathcal{O}(M(m+r)^2Nl^2)$, a factor N^3 less. Of course the circulant MOESP would require also the signal transformations and the*

construction of the global matrices (steps 2 and 6 of Algorithm 6.1); these operations can be done with the FFT algorithm, and they would have a computational cost of $\mathcal{O}(M(r + m)N \log N)$ and $\mathcal{O}(l^2 N \log N)$ respectively, which are anyway quite smaller compared to the QR factorization.

6.2.3 Some simulation results

Measurement noise

For demonstrating the use of the algorithm, a stable circulant system of 12th order, with $N = 4$, $l = 3$, $m = 1$ and $r = 1$ was randomly generated. The four input signals are made of 200 random samples each; white measurement noise has been added to all the four outputs.

In the test, we generated 250 different input/output pairs, and used them to identify the system. Algorithm 6.1 (from now on, we will call it “circulant MOESP”) was used and compared to a standard MOESP that assumes no structure at all for the system. In Figure 6.2 the poles of the true system are shown, together with the poles identified with the two different methods in 50 of the 250 runs; the poles identified with standard MOESP are indicated by a cross, while those found with the algorithm which assumes a circulant structure are indicated by a circle. At a glance it is possible to see that the circles are in general closer to the true poles if compared to the crosses (Figure 6.3 shows a magnification around one of the poles). Table 6.1 shows this observation in a more rigorous way, by comparing the mean square of the error in identifying each of the poles of the system.

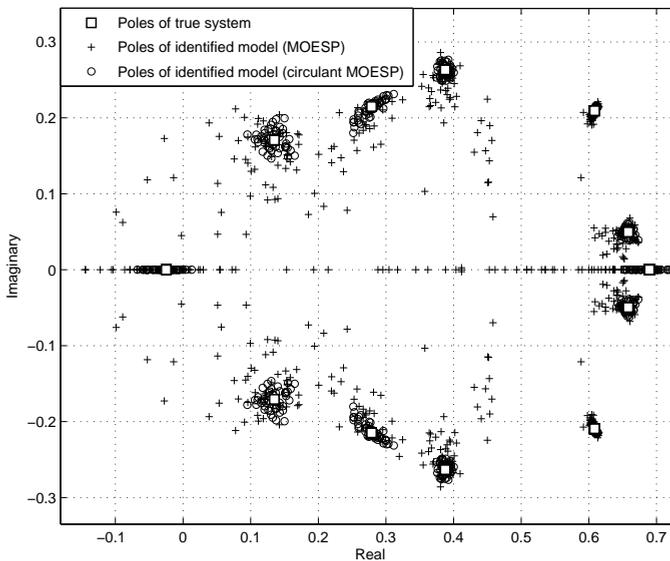


Figure 6.2: Poles of the identified model in a set of 50 different experiments.

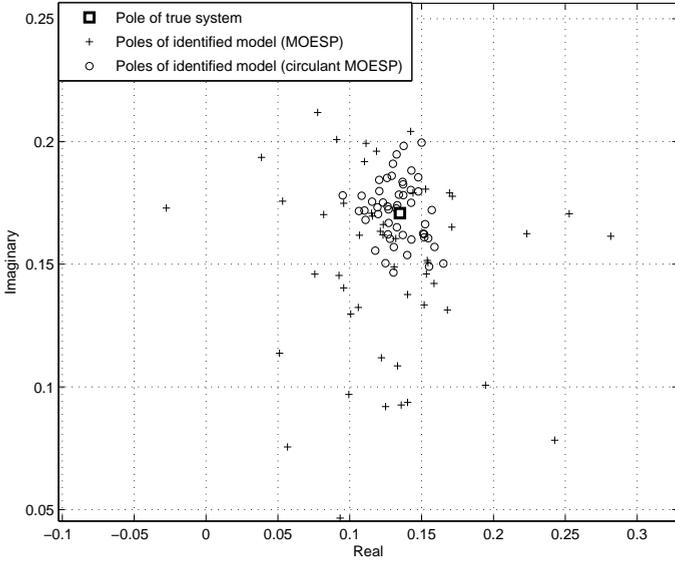


Figure 6.3: Detail of Figure 6.2 around one of the poles.

Table 6.1: Comparison of the performance of the two different methods in identifying the poles with measurement noise.

Pole	Root mean square error	
	standard MOESP	circulant MOESP
-0.02486	0.04641	0.01958
$0.13497 \pm 0.17077j$	0.06768	0.01879
$0.27881 \pm 0.21487j$	0.08064	0.02038
$0.38761 \pm 0.26329j$	0.02801	0.00877
$0.60841 \pm 0.20941j$	0.00783	0.00394
$0.65795 \pm 0.04966j$	0.02973	0.00870
0.68937	0.05027	0.01679

So this example suggests that if we have a system with circulant structure, the novel method performs better than standard MOESP.

Not perfectly circulant systems

Another test has been done adding a “random perturbation” to the \mathcal{A} matrix as well. This causes the system to be not perfectly circulant (that is most likely in real-life situations), but it has been verified that the method is still applicable; the idea is to show that small perturbations in the circulant structure do not cause completely wrong results. Again, we generated 250 different input/output pairs (with measurement noise), and used them to identify the system. For each pair, the \mathcal{A} matrix has been perturbed with a different random perturbation matrix, each element of which was smaller than $1/1000$ in modulus. For small perturbations as these, there is still an advantage in the accuracy of the method with respect to standard MOESP, as shown in Figure 6.4 and in Table 6.2.

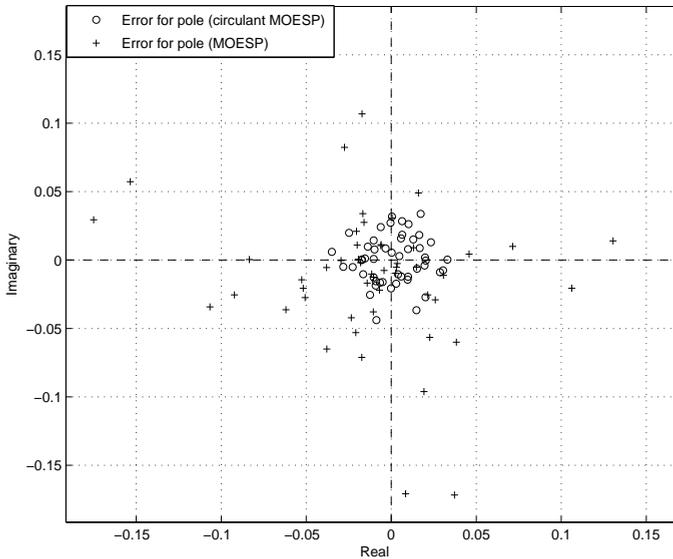


Figure 6.4: Error in identifying one of the poles (the second in Table 6.2) in 50 experiments with perturbation on the \mathcal{A} matrix.

Vibrating plate

A last test has been executed with the help of a finite element simulation. A kind of test bed for a vibration control experiment was designed using a finite element software (ABAQUS); the experiment consists of a metallic plate clamped at the corners, with four co-located [70] actuator and sensor pairs. The overall set-up satisfies the circulant symmetry and it is shown in Figure 6.5.

Table 6.2: Comparison of performance of the different methods in identifying the poles, with measurement noise and a perturbation on \mathcal{A} .

Pole (if no noise)	Root mean square error	
	standard MOESP	circulant MOESP
-0.02486	0.05287	0.02033
$0.13497 \pm 0.17077j$	0.06198	0.02002
$0.27881 \pm 0.21487j$	0.07897	0.02169
$0.38761 \pm 0.26329j$	0.02744	0.00926
$0.60841 \pm 0.20941j$	0.00771	0.00426
$0.65795 \pm 0.04966j$	0.02874	0.00922
0.68937	0.05199	0.01857

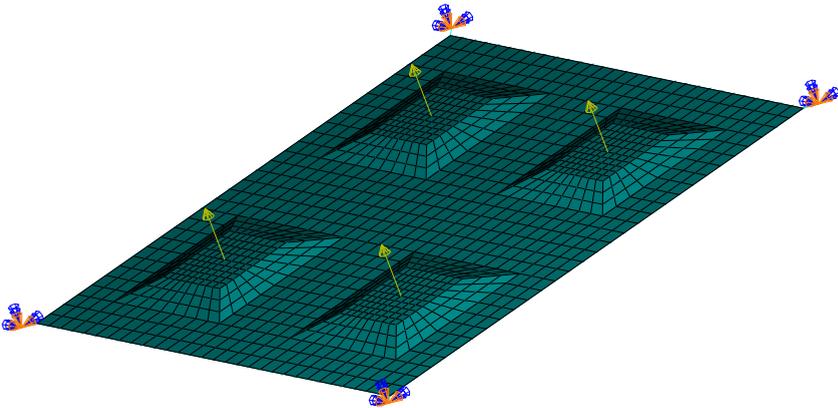


Figure 6.5: Finite element model of a vibrating plate. The plate is clamped at the corners, the arrows indicate the location of the input forces. The measured output is the displacement of the application points of the forces.

An experiment was executed with frequency sweeps as inputs. The inputs were made of 512 samples each, and a time step of 0.02 s was used. Algorithm 6.1 has been used to identify a 40th order model for the plate, in a noiseless case as well as with white measurement noise (with a signal to noise ratio of about 30 dB). The Bode plot of the transfer function from one actuator to its co-located sensor is shown in Figure 6.6. In the picture we can see that the peaks of the transfer function match the eigenfrequencies of the system as they can be computed by the software, and that the presence of noise does not change the result significantly. It is interesting to compare this to the results that are obtained with standard MOESP in both cases, which are shown in Figure 6.7. We can see that the noiseless case yields almost the same result, while the noise effects a much bigger distortion of the transfer function when regular MOESP is used.

6.3 Identification of decomposable systems in general

If the pattern matrix is somehow known a priori, the problem of identifying decomposable systems in general seems quite similar to the problem of identifying circulant systems. But this is not exactly true: in this section we are going to use the same ideas as before to develop algorithms for decomposable system identification, which will be significantly more complicated than the ones shown for circulant systems.

6.3.1 Differences with respect to circulant system identification

As for the circulant system identification, the basic idea is the following: once it has been assumed that the input/output data comes from a decomposable system whose structure we know, we can perform the identification on the transformed data \hat{u} and \hat{y} :

$$\begin{aligned}\hat{u} &= (\mathcal{S} \otimes I_m)^{-1}u \\ \hat{y} &= (\mathcal{S} \otimes I_r)^{-1}y\end{aligned}\tag{6.43}$$

similarly to what is done in (6.20). The only knowledge that is required is the matrix \mathcal{S} that diagonalizes \mathcal{P} , with which \hat{y} and \hat{u} can be computed. As we can then apply the identification to the system in the decomposed form, we know that instead of identifying the whole system with the full input \hat{u} and the full output \hat{y} we can identify each modal subsystem independently; each one of these subsystems is related to only a part of the signal, i.e. the i^{th} subsystem has only \hat{u}_i as input and \hat{y}_i as output. This procedure has again reduced the problem of identifying an model of order Nl with Nm inputs and Nr outputs into the problem of identifying N systems of order l with m inputs and r outputs. Notice that in this way the computational complexity of the subspace identification algorithm is reduced. Up to here, this is a very general idea and basically any identification method for MIMO systems can be used; later on we will again focus on the application of MOESP.

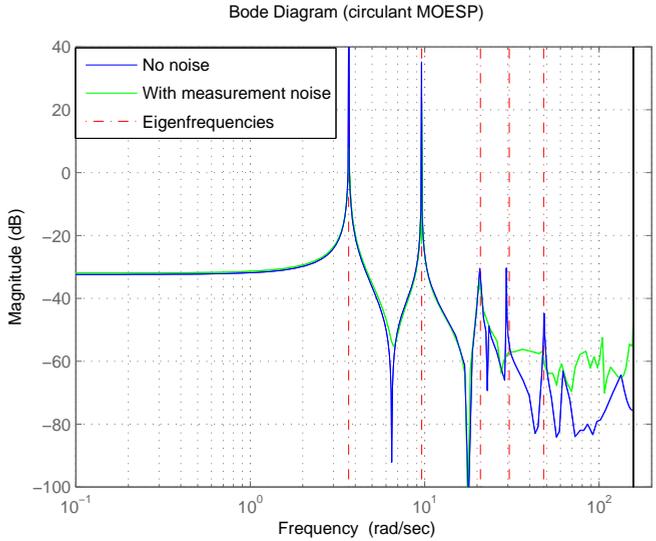


Figure 6.6: Bode plot of the transfer from one actuator to its co-located sensor, for the model identified with circulant MOESP. The dash-dotted lines indicate the frequencies of the first five eigenmodes as they were computed by the finite element software.

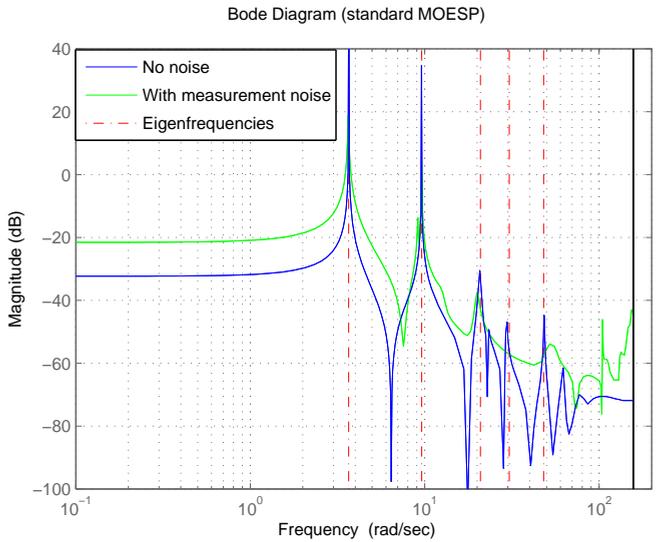


Figure 6.7: Bode plot of the transfer from one actuator to its co-located sensor, for the model identified with regular MOESP.

However, the procedure as described so far is not enough if we want to have a decomposable realization of the state-space system as the final result. In fact, applying any identification method to the modal subsystem will yield N sets of state-space matrices, which we call $\bar{A}_i, \bar{B}_i, \bar{C}_i$ and \bar{D}_i , which are completely independent from each other. If we stack these matrices into block diagonal matrices $\bar{A}, \bar{B}, \bar{C}$ and \bar{D} , and then we transform them back with (2.35) (such that they describe the dynamics between the original input and outputs u and y instead of the transformed \hat{u} and \hat{y}), what we get is not a decomposable system. In fact, as stated in Corollary 2.2, only if these matrices $\bar{A}_i, \bar{B}_i, \bar{C}_i$ and \bar{D}_i are “bold” (parameterized according to (2.34)), i.e. in the set $\mathcal{B}_{P, \bullet, \bullet}$, then the untransformed version will be in $\mathcal{G}_{P, \bullet, \bullet}$. For example, this means that for the “A” matrix we will need:

$$\begin{aligned}\bar{A}_1 &= \bar{A}_a + \lambda_1 \bar{A}_b \\ \bar{A}_2 &= \bar{A}_a + \lambda_2 \bar{A}_b \\ &\vdots \\ \bar{A}_N &= \bar{A}_a + \lambda_N \bar{A}_b\end{aligned}\tag{6.44}$$

but in general these \bar{A}_a and \bar{A}_b do not exist, even if we know that there exist a realization of the system that is decomposable; in fact if we use subspace identification methods we know that the state-space matrices of the system are obtained up to a similarity transformation; so we can see each of these \bar{A}_i as the arbitrarily transformed version of an ideal, underlying matrix that satisfies the constraint. So there is the need of finding a way to “force” the identified matrices into the constraint. In the case of circulant systems, this problem does not exist, as any block diagonal matrix can be transformed into a circulant matrix by the inverse transformation. Figures 6.8 and 6.9 graphically explain the differences between circulant and decomposable systems.

6.3.2 The novel algorithm

What we can try to do is to find a procedure that approximates the system $\bar{A}, \bar{B}, \bar{C}$ and \bar{D} with a system of “bold” matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and \mathbf{D} . As it was done for the design of controllers in [59], we can try to use Linear Matrix Inequalities (LMIs) where constraints can easily be introduced.

A possible way of approximating a system with state-space matrices (A, B, C, D) with another state-space realization $\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}$ (which might have some constraints) is to compute the difference between the systems and minimize the \mathcal{H}_∞ norm of this difference. The difference has the following realization:

$$\begin{aligned}\hat{A} &= \begin{bmatrix} A & 0 \\ 0 & \tilde{A} \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} B \\ \tilde{B} \end{bmatrix}, \\ \hat{C} &= [C \quad -\tilde{C}], \quad \hat{D} = [D - \tilde{D}].\end{aligned}\tag{6.45}$$

Computing the \mathcal{H}_∞ norm of this difference with the methods shown in [18] leads

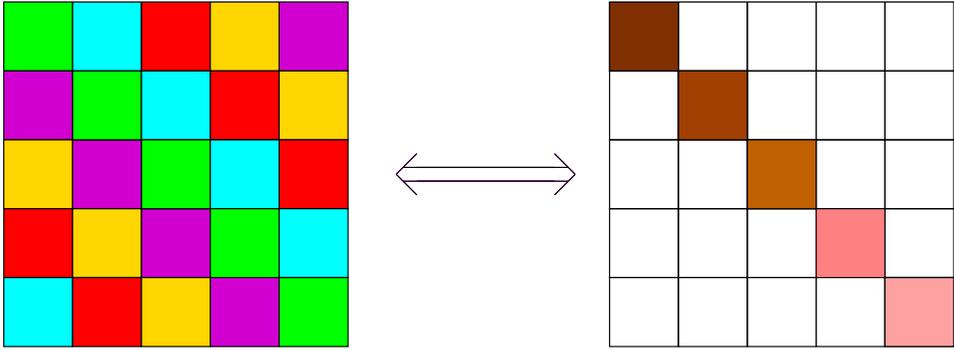


Figure 6.8: The decomposition transformation turns every circulant matrix into a block diagonal matrix, and viceversa; to every matrix in $\mathcal{C}_{N,p,q}$ corresponds one and only one matrix in $\mathcal{D}_{N,p,q}$.

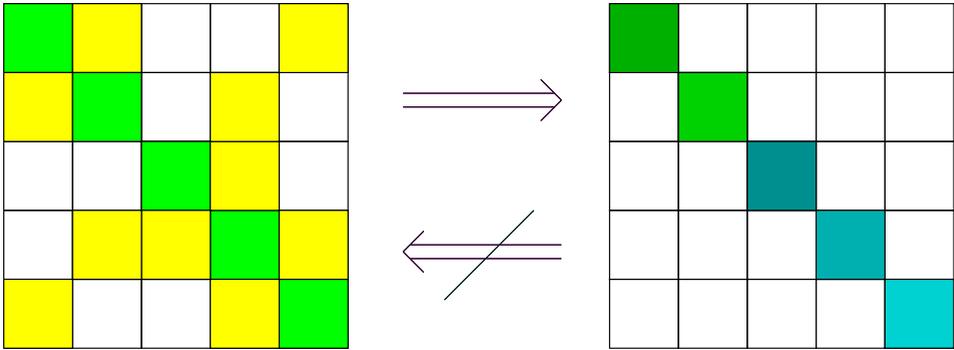


Figure 6.9: The decomposition transformation turns every matrix of a decomposable system into a block diagonal matrix, but the converse is not true; to every matrix in $\mathcal{G}_{N,p,q}$ corresponds one and only one matrix in $\mathcal{B}_{\mathcal{P},p,q}$, but $\mathcal{B}_{\mathcal{P},p,q}$ is a subset of $\mathcal{D}_{N,p,q}$.

to the following LMI:

$$\begin{bmatrix}
 P_{11} & P_{12} & AG_{11} & AG_{12} & B & 0 \\
 * & P_{22} & \tilde{A}G_{12}^T & \tilde{A}G_{22} & \tilde{B} & 0 \\
 * & * & G_{11} + G_{11}^T - P_{11} & G_{12} + G_{12}^T - P_{12} & 0 & G_{11}^T C^T - G_{12}^T \tilde{C}^T \\
 * & * & * & G_{22} + G_{22}^T - P_{22} & 0 & G_{12}^T C^T - G_{22}^T \tilde{C}^T \\
 * & * & * & * & I & D^T - \tilde{D}^T \\
 * & * & * & * & * & \gamma^2 I
 \end{bmatrix} \succ 0 \tag{6.46}$$

The \mathcal{H}_∞ norm is the minimum value of γ for which (6.46) has a feasible solution, where $P_{11} = P_{11}^T, P_{22} = P_{22}^T, P_{12}, G_{11}, G_{22}$ and G_{12} are optimization variables.

The details of this derivation are routine and not reported here for brevity.

If we want to use (6.46) for synthesis, i.e. for finding the \tilde{A} , \tilde{B} , \tilde{C} , \tilde{D} which minimize γ , then (6.46) is no longer an LMI due to the presence of products between unknowns (e.g. $\tilde{A}G_{12}^T$), but a Bilinear Matrix Inequality (BMI). So the optimization problem cannot be solved directly by LMI solvers, but it can be solved with an iterative procedure: one of the unknowns in the product is kept constant (e.g. \tilde{A}), and then the problem is again an LMI that can be solved; the result for the other unknown in the product (e.g. G_{12}^T) can then be assumed as constant in order to solve for the other one (\tilde{A}) again, and so on. Though this procedure does not guarantee the global optimum, we will see from the examples that it proves to be effective. A similar procedure is described in [35] for the purpose of model order reduction of linear systems (notice that A, B, C, D and $\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}$ might represent systems of different order), and also in this reference, the same iterative method is proposed in order to solve the BMI.

At this point, all the elements needed to complete the identification algorithm have been given; (6.46) can be used to cast the identified model into a decomposable system model: a set of N coupled BMIs is needed, namely:

$$\left[\begin{array}{cccccc} P_{11,i} & P_{12,i} & \bar{A}_i G_{11,i} & \bar{A}_i G_{12,i} & \bar{B}_i & 0 \\ * & P_{22,i} & \mathbf{A}_i G_{12,i}^T & \mathbf{A}_i G_{22,i} & \mathbf{B}_i & 0 \\ * & * & G_{11,i} + G_{11,i}^T - P_{11,i} & G_{12,i} + G_{12,i}^T - P_{12,i} & 0 & G_{11,i}^T \bar{C}_i^T - G_{12,i}^T \mathbf{C}_i^T \\ * & * & * & G_{22,i} + G_{22,i}^T - P_{22,i} & 0 & G_{12,i}^T \bar{C}_i^T - G_{22,i}^T \mathbf{C}_i^T \\ * & * & * & * & I_m & \bar{D}_i^T - \mathbf{D}_i^T \\ * & * & * & * & * & \gamma^2 I_r \end{array} \right] \succ 0$$

for $i = 1 \dots N$

(6.47)

where A, B, C, D are replaced by $\bar{A}_i, \bar{B}_i, \bar{C}_i, \bar{D}_i$ and $\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}$ are replaced by $\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i, \mathbf{D}_i$; remember that these last “bold variables” are constrained to be function of only two matrices each (e.g., $\mathbf{A}_i = A_a + \lambda_i A_b$, etc.). We are ready now to formulate the algorithm, which we provide in two different versions. As an identification method we propose MOESP, which can deal only with white measurement noise, but the algorithms can easily be extended to other methods which assume different noise properties.

Problem Description 6.2 (Decomposable system identification)

A set of N input signals $u_i(k) \in \mathbb{R}^{m \times 1}$ and N output signals $y_i(k) \in \mathbb{R}^{r \times 1}$ is given, for $i = 1, \dots, N$ and $k = 1, \dots, k_{\max}$. This set of data is associated with a dynamical system; we assume that this system can be modelled as a decomposable system (2.5) whose pattern matrix \mathcal{P} we know. Moreover, N , m and r are known. Goal: identify an lN^{th} order decomposable system model from input/output data.

Algorithm 6.2 (Subspace Identification of decomposable systems I)

Consider Problem 6.2. The problem is solved in the following steps:

1. Compute the matrix S that diagonalizes \mathcal{P} .
2. Transform input and output signals, by computing:

$$\begin{aligned}\hat{u} &= (S \otimes I_m)^{-1}u \\ \hat{y} &= (S \otimes I_r)^{-1}y\end{aligned}\quad (6.51)$$

3. Verify that each signal $\hat{u}(k)$ is persistently exciting ([92]) of at least order l .
4. Use MOESP to identify the independent state-space models of order l from each \hat{u}_i/\hat{y}_i pair:

$$\begin{cases} \hat{x}_i(k+1) = \overline{A}_i \hat{x}_i(k) + \overline{B}_i \hat{u}_i(k) \\ \hat{y}_i(k) = \overline{C}_i \hat{x}_i(k) + \overline{D}_i \hat{u}_i(k) \end{cases} \quad \text{for } i = 1, \dots, N \quad (6.52)$$

5. Then use (6.47) to find the matrices of a decomposable system ($\mathbf{A}_i = A_a + \lambda_i A_b$, $\mathbf{B}_i = B_a + \lambda_i B_b$, $\mathbf{C}_i = C_a + \lambda_i C_b$, $\mathbf{D}_i = D_a + \lambda_i D_b$) that approximate the systems made of \overline{A}_i , \overline{B}_i , \overline{C}_i , \overline{D}_i . Use the following procedure:

- (a) Choose initial values of A_a , A_b , C_a , C_b , (e.g., zero matrices);
- (b) Assume A_a , A_b , C_a and C_b as constant using the values previously found, and solve (6.47) with B_a , B_b , D_a , D_b , $P_{11,i} = P_{11,i}^T$, $P_{22,i} = P_{22,i}^T$, $P_{12,i}$, $G_{11,i}$, $G_{22,i}$ and $G_{12,i}$ as decision variables, minimizing γ ;
- (c) Assume $G_{11,i}$, $G_{22,i}$ and $G_{12,i}$ as constant using the values previously found, and solve (6.47) with A_a , A_b , C_a , C_b , B_a , B_b , D_a , D_b , $P_{11,i} = P_{11,i}^T$, $P_{22,i} = P_{22,i}^T$ and $P_{12,i}$ as decision variables, minimizing γ ;
- (d) If γ is smaller than a desired value or after a predetermined number of iterations, end the loop, otherwise go to b).

6. The identified global system matrices are then given by:

$$\begin{aligned}\mathbf{A} &= I_N \otimes A_a + \mathcal{P} \otimes A_b \\ \mathbf{B} &= I_N \otimes B_a + \mathcal{P} \otimes B_b \\ \mathbf{C} &= I_N \otimes C_a + \mathcal{P} \otimes C_b \\ \mathbf{D} &= I_N \otimes D_a + \mathcal{P} \otimes D_b\end{aligned}\quad (6.53)$$

Notice again that we have assumed that the model order l is chosen a priori, even if MOESP can give hints on the model order.

Another more computationally complex identification procedure is possible. Subspace identification methods like MOESP can be considered as a procedure in two steps, where the first step is basically the identification of a model of high order (the parameter that usually is called s in the literature) and the second step (which involves the truncation of the singular value decomposition of the extended observability matrix) can be interpreted as a model order reduction (see [92] for details). As (6.47) can be used also as a model order reduction tool (similarly to what done in [35]), we can think of performing this model order reduction part using the iterative procedure instead of using the truncation. This would allow us to both reduce the model order and cast the model into a decomposable system at the same time, instead of doing one first and then the other (as in Algorithm 6.2). This leads us the following Algorithm:

Algorithm 6.3 (Identification of decomposable systems II)

Consider again Problem 6.2. The problem is solved in the following steps:

1. 2. 3. *Same as in Algorithm 6.2.*
4. *Use MOESP to identify a model $(\bar{A}_i, \bar{B}_i, \bar{C}_i, \bar{D}_i)$ of order $a > l$ from each \hat{u}_i/\hat{y}_i pair. A choice for a can be $a \simeq 2l$; s can be $a + 1$.*
5. *Use (6.47) to find the matrices of a decomposable system $(\mathbf{A}_i = A_a + \lambda_i A_b, \mathbf{B}_i = B_a + \lambda_i B_b, \mathbf{C}_i = C_a + \lambda_i C_b, \mathbf{D}_i = D_a + \lambda_i D_b)$ that approximate the systems made of $\bar{A}_i, \bar{B}_i, \bar{C}_i, \bar{D}_i$. Use the same procedure as in 5. in Algorithm 6.2.*
6. *Same as in Algorithm 6.2.*

Notice that this last algorithm has a higher computational cost due to the presence of bigger matrices in the LMI (e.g. $\bar{A}_i \in \mathbb{R}^{s \times s}$ instead of $\in \mathbb{R}^{l \times l}$). Also notice that, for both algorithms, if \mathcal{P} is not symmetric, then there may be complex eigenvalues that will appear in the LMIs; also \hat{u}_i and \hat{y}_i might be complex. In any case this does not affect the computations (the same as for the circulant case), as LMIs can be solved also for the complex case, and subspace identification methods are “numerical recipes” involving algebraic operations (matrix sum, matrix product, singular value decomposition or QR factorization) that can be extended to complex numbers.

6.3.3 Numerical results

In this section we present the results of the application of the identification algorithms to an academic example. We randomly generated four different stable systems of order 24, made of the interconnection of 8 identical SISO subsystems

of order 3, with:

$$\mathcal{P} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (6.54)$$

According to the notation, we have $l = 3$, $N = 8$, $r = 1$ and $m = 1$; the pattern matrix is symmetric so we will have only real numbers in our computations.

We simulate the system putting random signals to the 8 input channels, and subsequently white measurement noise is added to the outputs. We use data batches of 300 samples to perform the identification. We compare the results of the identification using standard MOESP for MIMO systems, Algorithm 6.2 and Algorithm 6.3; the index that we use for comparing is the Variance Accounted For, or VAF [92], defined as:

$$\text{VAF} = 100\% \cdot \max \left(0, 1 - \frac{\frac{1}{N} \sum_{k=1}^N \|y(k) - y_e(k)\|_2^2}{\frac{1}{N} \sum_{k=1}^N \|y(k)\|_2^2} \right) \quad (6.55)$$

where N is the number of samples in the batch and y_e is the output that can be estimated simulating the identified system; the higher the value, the better the model approximates the original input/output data. The parameter a is chosen as 6 (twice the order of the modal subsystems) for Algorithm 6.3, and the maximum number of iterations for solving the BMIs is set to 60. The results of the tests are shown in Figure 6.10 and Figure 6.11; the former shows the VAF with respect to the identification data set, while the latter shows the VAF for a validation set.

From the plots it appears that Algorithm 6.3 always provides a better VAF with respect to the unstructured approach. This confirms that using the prior knowledge might, in most cases, improve the quality of the estimate. Algorithm 6.2 sometimes fails to find a good estimate, and this justifies the choice that was made in upgrading it into Algorithm 6.3, by clumping together the model order reduction and the “structuring” part.

Figure 6.12 shows the sparsity of the state matrix of the models identified with the novel algorithms, to be compared with the standard MOESP which would deliver a full, unstructured matrix. The LMIs have been solved using the solver DSDP [8] and Yalmip [52] as interface. Figure 6.13 shows the typical convergence behavior for the two iterative algorithms.

6.4 Other special cases

As we have just seen, the solution that we have proposed for the general problem of identifying a decomposable system involves the solution of a BMI. The solution

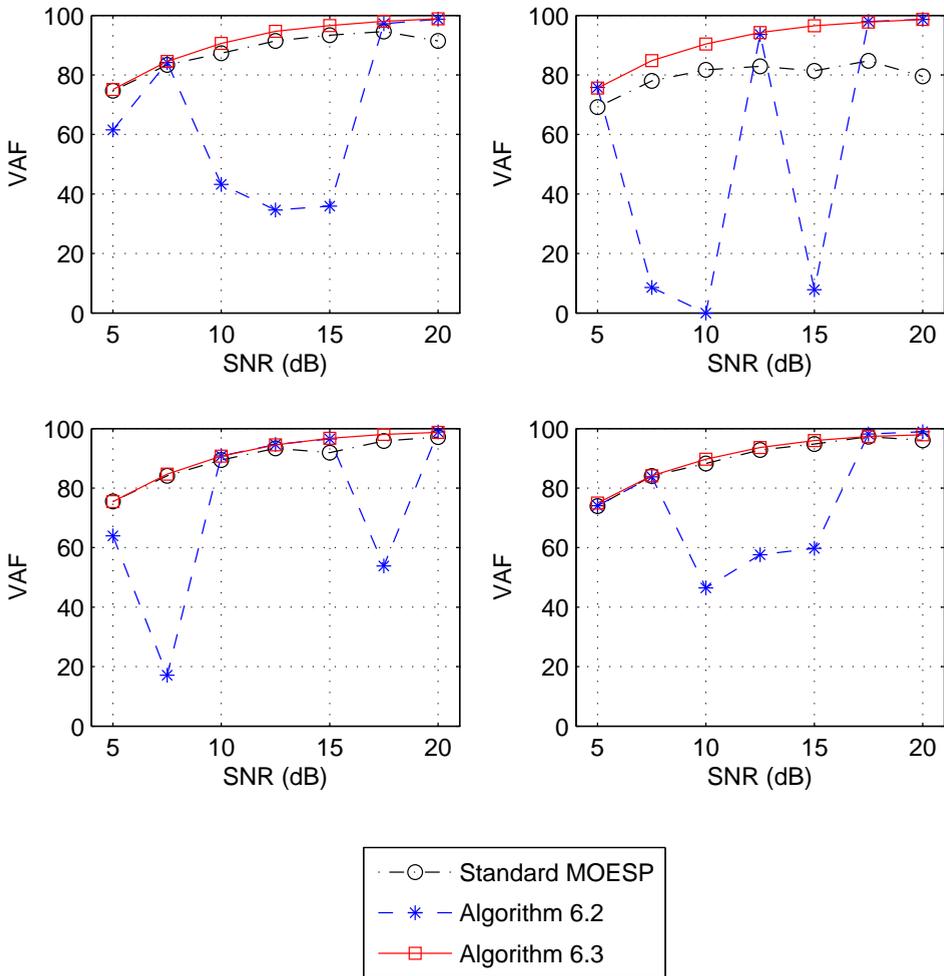


Figure 6.10: Comparison of the VAF (the average of the 8 channels) obtained with the different methods for 4 randomly generated systems, for various levels of signal to noise ratio (SNR); each graph corresponds to a different system. The VAF has been computed on the same data set that was used for system identification. It is possible to see that Algorithm 6.3 almost always scores the best, with a few exceptions where it is slightly outperformed by Algorithm 6.2. Algorithm 6.2 does not always prove reliable.

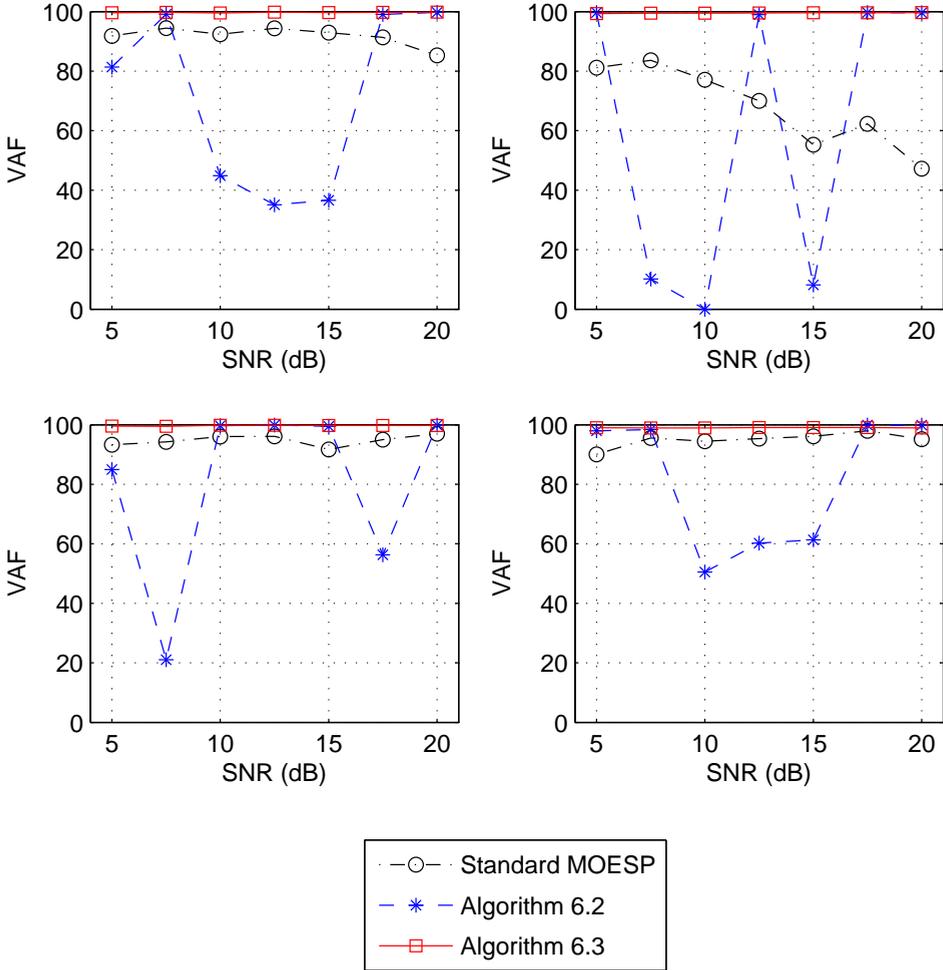


Figure 6.11: Comparison of the VAF for the identified models, using a noiseless validation data set that is different from the set used for identification. Algorithm 6.3 scores the best in all cases but one, where it is outperformed by Algorithm 6.2.

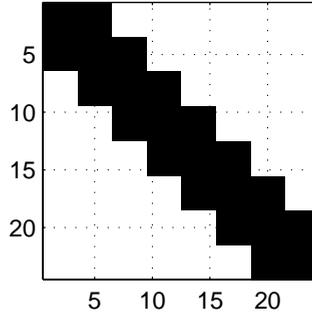


Figure 6.12: Sparsity of the “A” matrices identified with Algorithm 6.2 or Algorithm 6.3. The black colour indicates a non-zero entry.

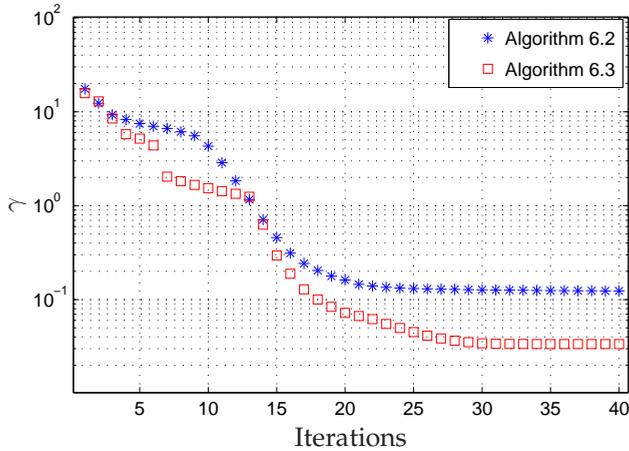


Figure 6.13: Convergence of the two algorithms (γ is the error in the approximation of the \mathcal{H}_∞ norm).

of a BMI cannot be guaranteed to be the global optimum, so this step makes the general procedure somehow weak, while the circulant identification algorithm is guaranteed to work all the times. Actually, there are other situations where this BMI step is not needed.

Let us look again at (6.44), which was used to justify the necessity of the nonlinear optimization step. Ideally, out of (6.44) we would like to get the \bar{A}_a and \bar{A}_b from the \bar{A}_i , for $i = 1 \dots N$. In general these \bar{A}_a and \bar{A}_b do not exist, simply because (6.44) contains N (matrix) equations and the (matrix) unknowns are only 2. This might not be true if the \bar{A}_i 's have a very special structure which would make the N equations linearly dependent with each other, a thing that alas we cannot guarantee as the \bar{A}_i 's are the result of a subspace identification algorithm, and so they are "corrupted" by random similarity transformations. It is easy to see that for the circulant case instead, we would have as many unknowns as the number of \bar{A}_i 's, but this is not the only case: the same thing would happen, for example, for a wide-sense decomposable system that has N pattern matrices; in fact, a circulant system is a wide-sense decomposable system with the powers of Π_N as pattern matrices, as shown in (2.51).

There might be also cases which can be solved without BMI where the unknowns are fewer than N , if the equations (6.44) are not linearly independent. Consider for example the class of symmetrically interconnected systems, introduced in Section 2.6.2. We have seen that a symmetrically interconnected system is actually equivalent to only two different systems. This means that we can put the signals corresponding to the $N - 1$ identical subsystems together, and identify for it a single set of state-space matrices, which we call $A_{2\dots N}$, $B_{2\dots N}$, etc. We have that (6.44) becomes:

$$\begin{aligned} \bar{A}_1 &= \bar{A}_a + \lambda_1 \bar{A}_b \\ \bar{A}_{2\dots N} &= \bar{A}_a + \lambda_{2\dots N} \bar{A}_b \end{aligned} \tag{6.56}$$

In this case, the two unknowns in the two equations can always be found by solving the linear system (excluding the "pathological" cases of course).

6.5 Conclusions

In this chapter, we have proposed a number of procedures that allow the identification of structured models of decomposable systems, thus enabling their use in distributed control design methods that are specifically targeted to this class of systems. These procedures are based on a transformation of the input/output signals and on an optimization procedure that solves an LMI recursively. This last step is not necessary in certain cases, for example for circulant systems and for symmetrically interconnected systems.

These procedures would allow a complete data-driven approach to the class of large scale systems that is the subject of this thesis. It is though interesting to point out that the examples of systems that we have given for the control part of this work (Chapters 3, 4 and 5) and for the identification part are quite different.

In fact, the control part is mainly focused on examples of multi-vehicle systems, for which the distributed identification procedure would not make much sense as we can easily identify each vehicle as a single unit. So, even if theoretically the identification and the control part would seem to fit together perfectly with each other, it might be true that for each specific application, only a part of the methods would apply.

Conclusions

With this thesis work we have extended the wealth of results in the field of distributed control and identification. We provided a set of controller synthesis methods applicable to the class of “decomposable systems”, and a complementary result concerning their identification. These method can be readily applicable, as shown in the examples contained in this thesis, but they can be also considered as a starting point for more research. A few hints on future directions are given here.

7.1 Summary and considerations

We focused on a class of distributed /large scale systems, which we called “decomposable systems”, that can mirror a wide set of relevant systems in terms of application. Decomposable systems portray systems made of identical subunits with an arbitrary interconnection pattern, so they are specifically suitable to describe vehicle formations or (micro)mechanical systems made of identical subparts. The reason for focusing on this class of system is that they have a remarkable decomposition property, that allows reducing any problem of analysis or synthesis regarding them to a problem concerning a parameter-varying system of the order of a subunit. From this consideration, it is possible to develop a variety of methods.

The main contribution of this thesis is in the controller synthesis methods, which we proved to be readily applicable to cases of interest in state-of-the-art technological applications, as satellite formation flying or control of deformable mirrors for adaptive optics. Moreover, these methods are based on LMIs and they can easily be combined with other robust control results to obtain extensions, as we have shown for the formation flying example, where a specific LTP extension was developed in order to cope with periodic non-circular orbits. As an additional result, we have exploited the same ideas that are at the base of the controller synthesis methods in order to develop system identification methods for the same class of systems, which are useful in case a first principle model of the plant is lacking.

7.2 Further research directions of this work

The research in this thesis, as any research, is of course not the end of the story. The field of distributed control can be thought of as still in its first developments, and more theoretical advances are needed and expected in the coming years. For what concerns the specific topic of this work, we can highlight a few directions that could be followed to continue with this research.

- **Switching topologies**

All the work done so far has only considered topologies (pattern matrices) that do not switch. If the topologies switch “slowly”, no problem is introduced: if all the configurations are stable and if enough time is given, the system will eventually show asymptotic stability. If the switching is arbitrarily fast, then these considerations do not hold anymore. The fact that Theorem 3.1, Theorem 3.2, etc. are based on a common Lyapunov function might lead us to think that it should be possible to prove at least the stability for any switching sequences. But there might be some further complications, as the equivalence between decomposable system and LPV system might not reach this level. In fact, in a LPV system only the dynamic law changes; but for a decomposable system, a change in topology means also a change in the decomposing transformation, which implies a change in how the state itself is defined. Even if we guarantee that the system is robust with respect to time varying λ_i with a common Lyapunov function, this is not enough to prove stability, because the state itself changes when switching. These considerations show how the analysis of switching topologies can be a challenging task.

- **Heterogenous systems**

The work done considers systems made of the interconnection of identical subsystems. An obvious question that we might ask is whether it is possible to extend the methods to heterogeneous systems, or at least, heterogeneous up to a certain extent. For example, it might be possible to include only a limited set of different subsystems, or the methods might be adapted to situations where the subsystems are “not too different” from one another. The solution might be in dividing the control problem in two parts, first designing a level of controller that casts the problem into the form that we can handle. In any case, we should expect that the complexity of the computations will grow at least with the number of different subsystem classes that are involved.

- **Non-diagonalizable pattern matrices**

The decomposition theorem can be proved only if the pattern matrix is diagonalizable. But will the controller design methods work as well if the matrix is not diagonalizable? After all, a non-diagonalizable matrix is only “an ε ” different from a diagonalizable matrix. Non-diagonalizable patterns could be proved to work as a limit case of the diagonalizable ones.

- **Identification of decomposable systems**

The general method for identification of decomposable systems has an approach based on BMIs. BMIs are non-convex problems, so the solution is not guaranteed to be the global optimal. It is matter of future research whether it is possible to solve such identification problem with convex optimization techniques instead.

And these of course are only examples of what can be further investigated.

7.3 The future of distributed control

We started this thesis with a description of the future plans for the Extremely Large Telescope, which can be seen as a motivation for the research in the field of large scale systems. Later on, formation flying was shown as another main motivating example. In the year 2000, the Panel of Future Directions in Control, Dynamics and Systems was formed to provide a renewed vision of future challenges and opportunities in the control field, along with recommendations to government agencies, universities, and research organizations to ensure continued progress in areas of importance to the industries. In the report [66] which was published in 2003, the Panel pointed out five critical fields whose development would be of the highest importance in the coming years. Among these fields, distributed control is cited as one of them:

Control distributed across multiple computational units, interconnected through packet-based communications, will require new formalisms for ensuring stability, performance, and robustness. This is especially true in applications where one cannot ignore computational and communication constraints in performing control operations.

This thesis work has looked only at a part of the problem, mainly ensuring stability and performance under the computational and communication constraints. From an analysis of the literature, as seen in the introduction of this work, LMI methods seem to be the natural choice to accommodate for constraints and structured variables for linear systems. Alas, the development in LMI techniques has not been followed by a parallel development of LMI solvers, which makes the practical applicability of the methods more problematic.

Large scale systems are a reality today, not only equations: in the same report, the Internet is cited as the "largest feedback control system ever built". All these considerations show how important the research in the field of large scale systems is, which can be seen as really technologically driven. The technology, for example, is moving towards extreme levels of miniaturization, which will soon make it possible to have millions of nano-machines working together to achieve a collective task. The control theory will have to keep pace with the technology, developing the right tools for managing these new kinds of systems. The fast pace of technology might also force the adoption a more practical approach. As said

in the introduction, reaching optimality for distributed control is already being considered as too hard a task; the systems stemming from the new technologies might be too complex even for using the suboptimality results we already have, which would force the use of heuristic kinds of approaches.

Bibliography

- [1] The Herschel Science Mission, European Space Agency. Available at <http://sci.esa.int/science-e/www/object/index.cfm?fobjectid=34682>.
- [2] The Terrestrial Planet Finder Mission, Jet Propulsion Laboratory. Available at <http://planetquest.jpl.nasa.gov/TPF>.
- [3] The European extremely large telescope. ESO Information Sheet, 2009. Available at <http://www.eso.org>.
- [4] R. Aldrovandi. *Special matrices of mathematical physics: stochastic, circulant and Bell matrices*. World Scientific, 2001.
- [5] J. Anthonis and H. Ramon. Linear mechanical systems and dyadic transfer function matrices. *Automatica*, 39(8):1353–1363, 2003.
- [6] B. Bamieh, F. Paganini, and M.A. Dahleh. Distributed control of spatially invariant systems. *IEEE Transactions on Automatic Control*, 47(7):1091–1107, 2002.
- [7] R.W. Beard, J. Lawton, and F.Y. Hadaegh. A coordination architecture for spacecraft formation control. *IEEE Transactions on Control Systems Technology*, 9(6):777–790, November 2001.
- [8] S.J. Benson, Y. Ye, and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization*, 10(2):443–461, 2000.
- [9] J.T Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control and Dynamics*, 21(2):193–207, 1998.
- [10] F. Borrelli and T. Keviczky. Distributed LQR design for identical dynamically decoupled systems. *IEEE Transactions on Automatic Control*, 53(8):1901–1912, September 2008.
- [11] A. Böttcher and S.M. Grudsky. *Spectral Properties of Banded Toeplitz Matrices*. SIAM, 2005.
- [12] J.W. Brewer. Kronecker products and matrix calculus in system theory. *IEEE Transactions on Circuits and Systems*, 25(9):772–781, September 1978.

- [13] R.W. Brockett and J.L. Willems. Discretized partial differential equations: Examples of control systems defined on modules. *Automatica*, 10(4):507–515, 1974.
- [14] F.R.K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.
- [15] R. Curtain and H. Zwart. *An Introduction to Infinite-Dimensional Linear Systems Theory*. Springer, 1995.
- [16] R. D’Andrea and G.E. Dullerud. Distributed control design for spatially interconnected systems. *IEEE Transactions on Automatic Control*, 48(9):1478–1495, September 2003.
- [17] P.J. Davis. *Circulant Matrices*. Wiley-Interscience, 1979.
- [18] M.C. de Oliveira, J.C. Geromel, and J. Bernoussou. Extended \mathcal{H}_2 and \mathcal{H}_∞ norm characterizations and controller parametrizations for discrete-time systems. *International Journal of Control*, 75(9):666–679, June 2002.
- [19] N. Denis and D.P. Looze. \mathcal{H}_∞ controller design for systems with circulant symmetry. In *Proceedings of the 38th Conference on Decision & Control*, Phoenix, Arizona (USA), December 1999.
- [20] R. Diestel. *Graph Theory*. Springer, 1996.
- [21] N. Doelman, R. Fraanje, I. Houtzager, and M. Verhaegen. Real-time optimal control for adaptive optics systems. *European Journal of Control*, 15(3-4):480–488, 2009.
- [22] G.E. Dullerud and R. D’Andrea. Distributed control of heterogeneous systems. *IEEE Transactions on Automatic Control*, 49(12):2113–2128, December 2004.
- [23] G.E. Dullerud and S. Lall. A new approach for analysis and synthesis of time-varying systems. *IEEE Transactions on Automatic Control*, 44(8):1486–1497, 1999.
- [24] R.W. Farquhar. The control and use of libration point satellites. Technical Report R – 346, NASA, 1970.
- [25] J.A. Fax. *Optimal and cooperative control of vehicle formations*. PhD thesis, California Institute of Technology, Pasadena, CA, 2002.
- [26] J.A. Fax and R.M. Murray. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9):1465–1476, 2004.
- [27] G.F. Franklin, D.J. Powell, and M.L. Workman. *Digital Control of Dynamic Systems*. Prentice Hall, 1997.
- [28] C.V.M. Fridlund. Darwin, the infrared space interferometry mission. *ESA Bulletin*, (103), August 2000.

- [29] W.K. Gawronski. *Advanced Structural Dynamics and Active Control of Structures*. Springer, 1998.
- [30] E. Gill. FAST mission requirements document. Technical report, Delft University of Technology, 2008.
- [31] C. Godsil and G. Royle. *Algebraic Graph Theory*. Springer, 2001.
- [32] M. Golay. Point arrays having compact non-redundant autocorrelations. *Journal of Optical Society America*, 61:272–273, 1971.
- [33] G.H. Golub and C.F. Van Loan. *Matrix Computations*. John Hopkins University Press, 3rd edition, 1996.
- [34] R.F.M.M. Hamelinck, N. Rosielle, M. Steinbuch, and N. Doelman. Large adaptive deformable mirror: design and first prototypes. In *Proceedings of the 5th annual SPIE conference on Optics and Photonics*, San Diego, August 2005. SPIE.
- [35] A. Helmersson. Model reduction using LMIs. In *Proceedings of the 33rd IEEE Conference on Decision and Control*, volume 4, pages 3217–3222, Lake Buena Vista, FL, USA, December 1994.
- [36] K.J.G. Hinnen. *Data-Driven Optimal Control for Adaptive Optics*. PhD thesis, Delft University of Technology, January 2007.
- [37] R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.
- [38] M. Hovd, R.D. Braatz, and S. Skogestad. SVD controllers for \mathcal{H}_2 -, \mathcal{H}_∞ - and μ -optimal control. *Automatica*, 33(3):433–439, 1994.
- [39] M. Hovd and S. Skogestad. Control of symmetrically interconnected plants. *Automatica*, 30(6):957–973, 1994.
- [40] K.C. Howell and H.J. Pernicka. Stationkeeping method for libration point trajectories. *AIAA Journal of Guidance, Control, and Dynamics*, 16(1):151–159, January-February 1993.
- [41] J.K. Hunter and B. Nachtergaele. *Applied Analysis*. World Scientific, 2000.
- [42] D. Izzo and L. Pettazzi. Autonomous and distributed motion planning for satellite swarm. *AIAA Journal of Guidance, Control, and Dynamics*, 30(2):449–459, March-April 2007.
- [43] A. Jadbabaie, J. Lin, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, June 2003.
- [44] M.R. Jovanović and B. Bamieh. Lyapunov-based distributed control of systems on lattices. *IEEE Transactions on Automatic Control*, 50(4):422–433, April 2005.

- [45] V. Kapila, A.G. Sparks, J.M. Buffington, and Q. Yan. Spacecraft formation flying: Dynamics and control. *AIAA Journal of Guidance, Control, and Dynamics*, 23(3):561–564, March-April 2000.
- [46] M.H. Kaplan. *Modern Spacecraft Dynamics and Controls*. Wiley, 1976.
- [47] J.E. Kulkarni, M.E. Campbell, and G.E. Dullerud. Stabilization of spacecraft flight in halo orbits: an \mathcal{H}_∞ approach. *IEEE Transactions on Control Systems Technology*, 14(3):572–578, 2006.
- [48] C. Langbort, R.S. Chandra, and R. D’Andrea. Distributed control design for systems interconnected over an arbitrary graph. *IEEE Transactions on Automatic Control*, 49(9):1502–1519, September 2004.
- [49] D.L. Laughlin, M. Morari, and R.D. Braatz. Robust performance of cross-directional basis-weight control in paper machines. *Automatica*, 29(6):1395–1410, 1993.
- [50] J. Leitner, F. Bauer, D. Folta, M. Moreau, R. Carpenter, and J. How. Distributed spacecraft systems develop new GPS capabilities. *GPS World: Formation Flight in Space*, February 2002.
- [51] L. Ljung. *System identification: theory for the user*. Prentice-Hall, 1987.
- [52] J. Löfberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.
- [53] D.C. Maessen, E. Gill, C.J.M. Verhoeven, and G.T. Zheng. Preliminary design of the Dutch-Chinese FAST micro-satellite mission. In *Proceedings of the 4S Symposium*, Rhodes, Greece, May 2008.
- [54] B.G. Marchand and K.C. Howell. Control strategies for formation flight in the vicinity of the libration points. *AIAA Journal of Guidance, Control, and Dynamics*, 28(6):1210–1219, November-December 2005.
- [55] P. Massioni, R. Fraanje, and M. Verhaegen. Adaptive optics application of distributed control design for decomposable systems. In *Proceedings of the 48th IEEE Conference on Decision & Control*, Shanghai, China, December 2009.
- [56] P. Massioni, T. Keviczky, E. Gill, and M. Verhaegen. A decomposition based approach to linear time-periodic distributed control of satellite formations. *IEEE Transactions on Control Systems Technology*, to appear.
- [57] P. Massioni and M. Verhaegen. A full block S-procedure application to distributed control. In *Proceedings of the 2010 American Control Conference (accepted for publication)*, Baltimore, USA.
- [58] P. Massioni and M. Verhaegen. Subspace identification of circulant systems. *Automatica*, 44(11):2825–2833, November 2008.
- [59] P. Massioni and M. Verhaegen. Distributed control for identical dynamically coupled systems: a decomposition approach. *IEEE Transactions on Automatic Control*, 54(1):124–135, January 2009.

- [60] P. Massioni and M. Verhaegen. Subspace identification of distributed, decomposable systems. In *Proceedings of the 48th IEEE Conference on Decision & Control*, Shanghai, China, December 2009.
- [61] C. McInnes. Autonomous rendezvous using artificial potential functions. *AIAA Journal of Guidance, Control, and Dynamics*, 18(2):237–241, March–April 1995.
- [62] T. McKelvey. Subspace methods for frequency domain data. In *Proceedings of the 2004 American Control Conference*, Boston, Massachusetts, June - July 2004.
- [63] M. Mesbahi and F.Y. Hadaegh. Formation flying control of multiple spacecraft via graphs, matrix inequalities, and switching. *AIAA Journal of Guidance, Control, and Dynamics*, 24(2):369–377, March–April 2001.
- [64] O. Montenbruck and E. Gill. *Satellite Orbits: Models, Methods and Applications*. Springer, 2000.
- [65] N. Motee and A. Jadbabaie. Optimal control of spatially distributed systems. *IEEE Transactions on Automatic Control*, 53(7):1616–1629, August 2008.
- [66] R.M. Murray, K.J. Åström, S.P. Boyd, R.W. Brockett, and G. Stein. Future directions in control in an information-rich world. *IEEE Control Systems Magazine*, 23(2):20–33, April 2003.
- [67] B.W. Parkinson and J.J. Spilker, editors. *Global Positioning System: Theory and Application*, volume 1. American Institute of Aeronautics and Astronautics, 1996.
- [68] A.P. Popov and H. Werner. A robust control approach to formation control. In *Proceedings of the 10th European Control Conference*, Budapest, Hungary, August 2009.
- [69] V.V. Prasolov. *Problems and Theorems in Linear Algebra*. American Mathematical Society, 1994.
- [70] A. Preumont. *Vibration Control of Active Structures, an Introduction*. Kluwer Academic Publisher, 2004.
- [71] A. Rantzer. On the Kalman-Yakubovich-Popov lemma. *Systems and Control Letters*, 28(1):7–10, 1996.
- [72] B.V. Rauschenbakh, M.Yu. Ovchinnikov, and S.M.P. McKenna-Lawlor. *Essential Spaceflight Dynamics and Magnetospherics*. Kluwer Academic Publishers, 2003.
- [73] B. Recht and R. D’Andrea. Distributed control of systems over discrete groups. *IEEE Transactions on Automatic Control*, 49(9):1446–1452, September 2004.
- [74] J.K. Rice and M. Verhaegen. Distributed control: A sequentially semi-separable approach for heterogeneous linear systems. *IEEE Transactions on Automatic Control*, 54(6):1270–1283, 2009.

- [75] M.C. Roggemann, B.M. Welsh, and R.Q. Fugate. Improving the resolution of ground-based telescopes. *Rev. Modern Physics*, 69(2):437–505, 1997.
- [76] M. Rotkowitz and S. Lall. A characterization of convex problems in decentralized control. *IEEE Transactions on Automatic Control*, 51(2):274–286, February 2006.
- [77] W. Rudin. *Fourier Analysis on Groups*. Wiley-Interscience, 1962.
- [78] N. Sandell, P. Varaiya, M. Athans, and M. Safonov. Survey of decentralized control methods for large scale systems. *IEEE Transactions on Automatic Control*, 23(2):108–128, 1978.
- [79] H. Schaub, S.R. Vadali, J. L. Junkins, and K. T. Alfriend. Spacecraft formation flying control using mean orbital elements. *AAS Journal of Astronautical Science*, 48(1):69–87, 2000.
- [80] C. Scherer, P. Gahinet, and M. Chilali. Multiobjective output-feedback control via LMI optimization. *IEEE Transactions on Automatic Control*, 42(7):896–911, 1997.
- [81] C. Scherer and S. Weiland. Linear matrix inequalities in control. Online, <http://www.dscs.tudelft.nl/~cscherer/lmi/notes05.pdf>, 2005. Lecture notes.
- [82] J. Shamma and G. Arslan. A decomposition approach to distributed control of systems on lattices. *IEEE Transactions on Automatic Control*, 51(4):701–707, April 2006.
- [83] C. Simó, G. Gómez, J. Libre, and R. Martinez. Station keeping of a quasiperiodic halo orbit using invariant manifolds. In *Second International Symposium on Spacecraft Flight Dynamics*, pages 65–70, October 1986.
- [84] E. St. John-Olcayto and J. Kluza. Low-complexity spacecraft guidance using artificial potential functions. In *Proceedings of the 3rd International Conference on Recent Advances in Space Technologies*, pages 591–594, June 2007.
- [85] G.E. Stewart, D.M. Gorinevsky, and G.A. Dumont. Feedback controller design for a spatially distributed system: the paper machine problem. *IEEE Transactions on Control Systems Technology*, 11(5):612–628, 2003.
- [86] J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software, Special issue on Interior Point Methods*, 11–12:625–653, 1999.
- [87] M.K. Sundareshan and R.M. Elbanna. Large-scale systems with symmetrically interconnected subsystems: analysis and synthesis of decentralized controllers. In *Proceedings of the 29th Conference on Decision & Control*, pages 1137–1142, Honolulu, Hawaii, December 1990.
- [88] P. Van Overschee and B. De Moor. N4SID: Subspace algorithms for the identification of combined deterministic and stochastic systems. *Automatica*, 30(1):75–93, 1994.

-
- [89] J.G. VanAntwerp, A.P. Featherstone, and R.D. Braatz. Robust cross-directional control of large scale sheet and film processes. *Journal of Process Control*, 11(2):149–177, 2001.
- [90] S.M. Veres, S.B. Gabriel, D.Q. Mayne, and E. Rodgers. Analysis of formation control of a pair of nanosatellites. *AIAA Journal of Guidance, Control, and Dynamics*, 25(5):971–974, September-October 2002.
- [91] M. Verhaegen. Identification of the deterministic part of MIMO state space models given in innovation form from input-output data. *Automatica*, 30(1):61–74, 1994.
- [92] M. Verhaegen and V. Verdult. *Filtering and System Identification: A Least Squares Approach*. Cambridge University Press, 2007.
- [93] B. Wie. *Space Vehicle Dynamics and Control*. AIAA Education Series, 1998.
- [94] J.C. Willems, P. Rapisarda, I. Markovsky, and B. De Moor. A note on persistency of excitation. *Systems & Control Letters*, 54(4):325–329, 2005.
- [95] K. Zhou, J.C Doyle, and K. Glover. *Robust and optimal control*. Prentice Hall, 1996.

List of symbols

\otimes	Kronecker product
∞	infinity
\mapsto	maps to
$\langle \cdot, \cdot \rangle$	inner product
\rightarrow	going to
$\ \cdot\ _{\mathcal{H}_\infty}$	\mathcal{H}_∞ norm
$\ \cdot\ _{\mathcal{H}_2}$	\mathcal{H}_2 norm
\bar{m}	complex conjugate of a scalar (or matrix) m
\mathbf{M}_i	“bold” matrix, parameterized as $\mathbf{M}_i = M_a + \lambda_i M_b$
M_a	constant part of the “bold” matrix \mathbf{M}_i
M_b	matrix coefficient of the linear part of the “bold” matrix \mathbf{M}_i
M^H	Hermitian (complex conjugate transpose) of the matrix M
M^T	transpose of the matrix M
$M \succ 0$	the Hermitian (symmetric) matrix M is strictly positive definite
$M \prec 0$	the Hermitian (symmetric) matrix M is strictly negative definite
•	bullet, replaces a symbol that can be inferred from the context
*	star, replaces parts of Hermitian (symmetric) matrices to avoid repetitions
\mathbb{C}	field of complex numbers
e	Napier’s Constant
i	index of the modal subsystems of a decomposable system
\bar{i}	index of the eigenvalues λ_i for which $\lambda_{\bar{i}} = \bar{\lambda}$
\underline{i}	index of the eigenvalues λ_i for which $\lambda_{\underline{i}} = \underline{\lambda}$
I_n	identity matrix of order n
k	discrete time index

l	local order of an element of a decomposable system
\mathcal{L}	(normalized) Laplacian matrix
m	number of local inputs of an element of a decomposable system
m_u	number of local control inputs of an element of a decomposable system
m_w	number of local disturbance inputs of an element of a decomposable system
N	number of elements of a decomposable system (or size of the pattern matrix)
\mathcal{P}	pattern matrix
\mathbb{R}	field of real numbers
r	number of local outputs of an element of a decomposable system
r_y	number of local measured outputs of an element of a decomposable system
r_z	number of local performance outputs of an element of a decomposable system
\mathcal{S}	matrix diagonalizing the pattern matrix
t	continuous time measure
$\text{trace}(M)$	trace of the matrix M
\mathbb{Z}	ring of integer numbers
δ	uncertainty in a system (scalar)
Δ	uncertainty in a system (matrix)
κ	condition number of a matrix
λ_i	eigenvalues of the pattern matrix
$\bar{\lambda}$	maximum eigenvalue of the pattern matrix (with only real eigenvalues)
$\underline{\lambda}$	minimum eigenvalue of the pattern matrix (with only real eigenvalues)
λ_{av}	average of the eigenvalues of a pattern matrix
Λ	diagonal matrix of the eigenvalues of the pattern matrix
Π_n	circulant shift permutation matrix of order n

List of Abbreviations

BMI	Bilinear Matrix Inequality
CW	Clohessy-Wiltshire
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
GPS	Global Positioning System
L_1	Lagrangian point of the Circular Restricted Three Body Problem located between the two primaries
LMI	Linear Matrix Inequality
LPV	Linear Parameter Varying
LQR	Linear Quadratic Regulator
LTI	Linear Time-Invariant
LTP	Linear Time-Periodic
LTV	Linear Time-Varying
MEMS	Micro Electro-Mechanical Systems
MIMO	Multiple-Input Multiple-Output
MOESP	Multivariable Output-Error State sPace Identification
PI	Proportional-Integral
SISO	Single-Input Single-Output
SVD	Singular Value Decomposition
VAF	Variance Accounted For

Summary

Decomposition Methods for Distributed Control and Identification

Paolo Massioni

The recent progress in technology, as in miniaturization and microtechnologies, is now forcing control engineers to confront themselves with systems of incredibly high dimensionality, with an ever growing number of input and output channels. For such systems, which we call “large scale systems”, it is necessary to take a new approach in order to solve control problems in a reasonable time, as well as for being able to design controllers which can be realized in a physically implementable way.

This thesis concerns a class of linear time-invariant large scale systems which we call “decomposable systems”. Decomposable systems describe systems made of a set of identical subsystems (or agents) that are interacting with each other, and they can be considered as an example of homogeneous systems with arbitrary interconnections. This means that each subsystem interacts only with a limited set of the others, and the interconnection pattern does not have to stick to a special structure or lattice. This class of systems describes very well a number of physical systems of interest, such as formations of vehicles or mechanical elements made of identical subparts. Decomposable systems are interesting under the point of view of the theory as they prove to be amenable to a kind of modal decomposition that depends only on the interconnection pattern and not on the specific system; this property, or “decomposition theorem”, is at the basis of all the results shown in this thesis.

The first part of this work concerns the problem of synthesizing distributed controllers for decomposable systems. By “distributed”, we mean that the controller can be implemented as a set of simple, local controllers interacting with each other, each of them commanding a limited set of neighboring subsystems. This approach is demanded when the number of subsystems is very high: in this case it is not feasible to implement a centralized controller that reads all the outputs and decides all the control inputs. The decomposition property is exploited to convert the problem of controller synthesis for the global decomposable system into the problem of synthesizing controllers for the “modal” systems making up its decomposed version; such modal subsystems have the same order as a single agent. Then, by using techniques from robust control as well as a few results from

graph theory, it is possible to cast the distributed controller synthesis problem as an optimization problem under Linear Matrix Inequality constraints. This leads to methods allowing performance-based synthesis of controllers in a variety of cases (e.g. \mathcal{H}_2 or \mathcal{H}_∞ performance, continuous or discrete time, state or output feedback). The methods only offer suboptimality results, which can be considered as the price to be paid in exchange for the distributed structure of the controller.

The distributed controller methods are then used in simulation for two examples of relevant technological application. The first application is the distributed \mathcal{H}_2 control of a deformable mirror for adaptive optics; as future Earth-based telescopes will feature deformable mirrors with actuators and sensors in the order of the thousands, the independence of the computational complexity from the system size makes the methods of this thesis very attractive. The second application considered is satellite formation flying, for which an extension of the \mathcal{H}_∞ methods to the time-varying case is proposed. The controller is evaluated on two examples of space missions involving non-Keplerian orbits.

The last part of the thesis concerns a problem that is complementary to the one of control; namely, it investigates the possibility of identifying models of decomposable systems from data. This possibility is useful in case such models are not available from first principles. It is shown that the decomposition property can be used in this case as well. The problem is first treated for a special case, namely for the class of circulant systems, and then examined in the general case. An approach based on subspace identification is proposed.

Samenvatting

Ontbindingsmethoden voor Gedistribueerde Regeling en Identificatie

Paolo Massioni

De recente vooruitgang in de technologie, zoals miniaturisatie en microtechnologieën, dwingt nu de regeltechnici tot het aangaan van de confrontatie met systemen van ongelooflijk hoge dimensionaliteit en met een steeds groter wordend aantal van input- en outputkanalen. Voor dergelijke systemen, die we “grootschalige systemen” noemen, is het noodzakelijk om een nieuwe aanpak te ontwikkelen waarmee niet alleen binnen een redelijke tijd opgelost kunnen worden, maar waarmee ook de regelaars kunnen worden gerealiseerd op een fysiek uitvoerbare manier.

Dit proefschrift heeft betrekking op een klasse van lineaire tijdsinvariante groot-schalige systemen die wij “ontbindbare systemen” noemen. Ontbindbare systemen zijn systemen die bestaan uit een verzameling van identieke subsystemen, of agenten, die elkaar beïnvloeden. Ze kunnen worden beschouwd als een voorbeeld van homogene systemen met willekeurige interconnecties. Dit betekent dat elk subsysteem slechts communiceert met een beperkte groep van andere systemen. Het verbindingspatroon hoeft zich hierbij niet te houden aan een bijzondere structuur of raster. Deze klasse van systemen beschrijft een aantal belangrijke fysieke systemen heel goed, zoals samenstellingen van voertuigen of mechanische onderdelen gemaakt van identieke subdelen. Ontbindbare systemen zijn theoretisch interessant, aangezien zij geschikt zijn voor een soort van modale ontbinding die uitsluitend afhangt van het onderlinge verbindingspatroon en niet van het specifieke systeem. Deze eigenschap, of “ontbinding theorema”, ligt aan de basis van alle resultaten beschreven in dit proefschrift.

Het eerste deel van dit werk heeft betrekking op het probleem van de synthese van gedistribueerde regelaars voor ontbindbare systemen. Met “gedistribueerd” bedoelen we dat de regelaar kan worden geïmplementeerd als een verzameling van eenvoudige, lokale regelaars in interactie met elkaar. Elke lokale regelaar beïnvloedt een beperkt aantal aangrenzende subsystemen. Deze aanpak is vereist wanneer het aantal subsystemen zeer hoog is: in dit geval is het niet haalbaar om een centrale regelaar te implementeren die alle uitgangen leest en alle acties bepaalt. De ontbindingseigenschap wordt benut om het probleem van de regelaarssynthese voor het ontbindbare systeem als geheel te zetten in het probleem

van de regelaarssynthese voor de “modale” systemen die gezamenlijk het gescheiden systeem vormen. Dergelijke modale subsystemen hebben dezelfde orde als een enkele agent. Dan, met behulp van technieken uit de robuuste regeltechniek, evenals resultaten uit de grafentheorie, is het mogelijk om het gedistribueerde regelaarssyntheseprobleem te formuleren als een optimalisatieprobleem met beperkingen in de vorm van lineaire matrix ongelijkheden. Dit leidt tot methoden waarmee prestatiegebaseerde regelaarssynthese voor verschillende criteria mogelijk wordt (bijvoorbeeld \mathcal{H}_2 of \mathcal{H}_∞ prestaties, continue of discrete tijd, toestand- of uitgangterugkoppeling). De methoden leveren alleen suboptimale resultaten, wat kan worden beschouwd als de prijs die betaald moet worden in ruil voor de gedistribueerde structuur van de regelaar.

De gedistribueerde regelmethoden worden vervolgens gebruikt in simulatiestudies van twee relevante technologische toepassingen. De eerste toepassing is de gedistribueerde \mathcal{H}_2 regeling van een vervormbare spiegel voor adaptieve optica. Als toekomstige op aarde gestationeerde telescopen vervormbare spiegels met duizenden actuatoren en sensoren zullen hebben, maakt de onafhankelijkheid van de regelkundige complexiteit en de systeemgrootte de methoden van dit proefschrift zeer aantrekkelijk. De tweede toepassing is satelliet formatie vliegen, waarvoor een uitbreiding van de \mathcal{H}_∞ methoden voor de tijdsafhankelijke situatie wordt voorgesteld. De regelaar wordt beoordeeld aan de hand van twee voorbeelden van ruimtemissies in niet-Kepleriaanse banen.

Het laatste deel van dit proefschrift beschouwt een probleem dat complementair is aan het regelprobleem, namelijk, de mogelijkheid van het identificeren van modellen van ontbindbare systemen uit meetgegevens. Deze mogelijkheid is handig in het geval dergelijke modellen niet beschikbaar zijn vanuit eerste principes. Aangetoond wordt dat de ontbindingseigenschap ook in dit geval gebruikt kan worden. Het probleem wordt eerst behandeld voor een speciaal geval, namelijk voor de klasse van circulante systemen, en vervolgens voor het algemene geval. Een aanpak gebaseerd op subspace identificatie wordt voorgesteld.

Curriculum Vitae

Paolo Massioni was born on August 10th 1980 in Milan, Italy. After getting the diploma of high school in 1999, he enrolled in the faculty of Aerospace Engineering in Politecnico di Milano, Milan, Italy, where he received his MSc degree in the April of 2005 and he was awarded as the best graduate in Aerospace Engineering of the year. During his master thesis, which was entitled: “Design and test of control methods for a prototype of hexapod robot for planetary exploration”, he worked on the development of a six-legged walking robot and on its control algorithms.

Just after graduating, he joined the European Space Research and Technology Centre (ESTEC) in Noordwijk, the Netherlands, for a short traineeship period. The goal of the work was the preliminary design of mobility systems for manned lunar exploration, and it was carried on in the Automation and Robotics section (TEC-MMA) of ESTEC.

In February 2006, he went back to Politecnico di Milano for a short research work on the development and implementation of a fuzzy logic based fault detection module of a commercial diagnostic software.

Since August 2006 he has been a PhD candidate at the Delft Center for Systems and Control, Delft University of Technology, the Netherlands. The project, supervised by prof. Michel Verhaegen, was sponsored by MicroNed and was entitled: “Decomposition Methods for Distributed Control and Identification”. During his PhD he received his DISC certificate and he took a visiting period in the European Space Agency, in the Guidance, Navigation and Control Systems section (TEC-ECN) for a short study on a formation flying problem.

