

Distributed Control for Identical Dynamically Coupled Systems: A Decomposition Approach

Paolo Massioni and Michel Verhaegen

Abstract—We consider the problem of designing distributed controllers for a class of systems which can be obtained from the interconnection of a number of identical subsystems. If the state space matrices of these systems satisfy a certain structural property, then it is possible to derive a procedure for designing a distributed controller which has the same interconnection pattern as the plant. This procedure is basically a multiobjective optimization under Linear Matrix Inequality constraints, with system norms as performance indices. The explicit expressions for computing these controllers are given for both \mathcal{H}_∞ or \mathcal{H}_2 performance, and both for static state feedback and dynamic output feedback (in discrete time). At the end of the paper, two application examples illustrate the effectiveness of the approach.

Index Terms—Decomposition, distributed control, formation flying, linear matrix inequalities (LMIs), paper machines.

I. INTRODUCTION

LARGE scale systems have been object of interest in system and control theory since the seventies [20]. For this class of systems, which can have a huge amount of inputs and outputs, both the synthesis and the implementation of a centralized controller are often not feasible in practice. For this reason, techniques aimed at investigating *decentralized* or *distributed* controller architectures have been studied since then [26].

This paper focuses on linear time invariant systems composed of the interconnection of a large number of identical subsystems (Fig. 1). The interconnection can either be (I) dynamic, in the sense that the states of each subsystem influence the states of the ones to which it is connected, or (II) due to the fact that the subsystems share a common goal, or (III) both. Systems of this kind, when expressed in state space form, will have system matrices containing identical blocks on the diagonals; if no interconnections are present, actually all of the matrices are only block diagonal. Systems of type (I) instead will also have non-zero off-diagonal terms in the state matrix, while type (II) will have them in the output matrix, and type (III) in both. Also situations when the input matrix introduces coupling can be considered as possible. In this paper we will show that if these interconnections have a certain structural property, then a special control synthesis procedure can be derived. This procedure will be

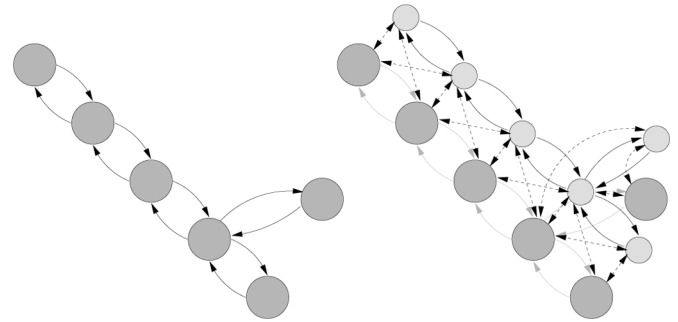


Fig. 1. On the left, a system made of the interconnection of identical subsystems. On the right, a controller (the smaller circles) that has the same spatially distributed structure of the plant. The solid lines represent interactions between members of the same system, the dashed lines are input/output flow between plant and controller.

useful for both simplifying the computational complexity of the problem as well as for finding a controller with a distributed architecture (Fig. 1, on the right).

The property that we require on the off-diagonal terms representing the interconnection is that they can 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 Laplacian matrices (for more details see [10], or [3] for an example of application in control theory). This property allows the system to be decomposed into a set of smaller subsystems. This, together with results coming from the theory of linear matrix inequalities (LMIs) [22], will allow the development of a procedure for controller synthesis with different performance criteria (e.g., \mathcal{H}_∞ or \mathcal{H}_2); moreover, these controllers will have a distributed architecture.

Examples of systems which fit into the category discussed here are found in formation flying [6], vehicle platoons [27], models coming from the discretization of partial differential equations [5], paper machines [23], segmented telescope mirrors [28] and others.

This paper is mainly inspired by the earlier work on formation stability in [13], where it is shown how the stability of a formation of vehicles with a distributed controller is related to the stability of a set of “modal” subsystems. 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 [5], symmetrically interconnected systems [17], [24] and in SVD (Singular Value Decomposition) controllers and their generalizations [2], [16], [25]; in all these cases it is shown how the global control synthesis problem can be reduced to a collection of simpler problems through a

Manuscript received March 04, 2008; revised May 14, 2008 and May 26, 2008. Current version published January 14, 2009. This work was supported by the MicroNed programme, an initiative of the Dutch Government. Recommended by Associate Editor R. D. Braatz.

The authors are with the Delft Center for Systems and Control, Delft University of Technology, Delft 2628 CD, The Netherlands (e-mail: p.massioni@tudelft.nl; m.verhaegen@moesp.org).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TAC.2008.2009574

transformation of inputs, outputs, and states (typically, a MIMO system with n inputs and n outputs is reduced to a set of n SISO systems).

The first contribution of this article is showing a general framework of these decomposition methods, in the form of a theorem that casts the possible decompositions into a single statement. A first simple version of the theorem, that is sufficient for introducing the distributed control method, is shown in Section III; for completeness, a more general version is then shown at the end of the paper in Section VI.

The second and main contribution is an LMI based method which can be used to design a controller that has the same interconnection structure as the plant. The earlier results mentioned above (e.g., [5] or [16]) allow the use of the decomposition just for simplifying the computations, but they all yield as 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; the procedure shown here instead will make sure that the controller has the same sparsity as the system, allowing a distributed implementation of the controller. The method that we propose can then be considered as an extension of the results of [13]: while this earlier article was focused on the stability of a system under a distributed controller, here instead we consider the problem of the synthesis of such controllers with performance criteria. However, we will see that the distributed structure is achieved at the cost of suboptimality with respect to a global controller.

The paper is organized as follows. Section II contains all the preliminary notions, including the formal definition of the systems which are the object of this paper and their properties. In Section III it is shown how it is possible to exploit such properties in controller synthesis, while Section IV contains the \mathcal{H}_∞ and \mathcal{H}_2 distributed controller synthesis methods which are the main result of this paper. Section V introduces a few concepts of graph theory that can be useful and Section VI reports the general decomposition theorem and its consequences. At last, Section VII contains two examples that can be quite helpful in understanding the content of the paper, and the conclusions of the article are reported in Section VIII.

II. PRELIMINARIES

We denote by \mathbb{R} the field of real numbers, by \mathbb{C} the field of complex numbers and by $\mathbb{R}^{n \times m}(\mathbb{C}^{n \times m})$ the set of real (complex) $n \times m$ matrices; \mathbb{Z} is the set of the integers. Let \otimes indicate the Kronecker product, and I_n be the identity matrix of order n . We use also the notation $A \succ 0$ to indicate that all the eigenvalues of the Hermitian matrix A are strictly positive.

We start by defining the class of matrices which are of interest in describing the dynamical systems considered in this paper.

Definition 1: Assume that $P_n \in \mathbb{R}^{n \times n}$ is diagonalizable [15]: let us define as $\mathfrak{G}^{P_n, p, q}$ the set of all matrices $\mathcal{M} \in \mathbb{R}^{n \times p \times q}$ for which there exist two matrices $M_a, M_b \in \mathbb{R}^{p \times q}$ such that

$$\mathcal{M} = I_n \otimes M_a + P_n \otimes M_b. \quad (1)$$

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

Lemma 2: Let $S_n \in \mathbb{C}^{n \times n}$ be a non singular matrix such that $\Lambda_{P_n} = S_n^{-1} P_n S_n$ is diagonal. If $\mathcal{M} \in \mathfrak{G}^{P_n, p, q}$ then

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

is block diagonal.

Proof: From Definition 1, we can write

$$\mathbf{M} = (S_n \otimes I_p)^{-1} (I_n \otimes M_a + P_n \otimes M_b) (S_n \otimes I_q)$$

then from the properties of the Kronecker product [4] we have

$$\begin{aligned} \mathbf{M} &= (S_n^{-1} I_n S_n \otimes I_p M_a I_q) + (S_n^{-1} P_n S_n \otimes I_p M_b I_q) \Leftrightarrow \\ \mathbf{M} &= I_n \otimes M_a + \Lambda_{P_n} \otimes M_b. \end{aligned} \quad (2)$$

Since I_n and Λ_{P_n} are diagonal, then we have that \mathbf{M} is block diagonal. ■

So it is immediate to find a kind of similarity transformation that renders a matrix in $\mathfrak{G}^{P_n, p, q}$ block diagonal, once the matrix that diagonalizes P_n 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). If we call \mathbf{M}_i the i th block in the diagonal of \mathbf{M} , it is easy to show that

$$\mathbf{M}_i = M_a + \lambda_i M_b \quad (3)$$

where λ_i is the i th entry in the diagonal of Λ_{P_n} , that is the i th eigenvalue of P_n . We define as $\mathfrak{D}^{P_n, p, q}$ the set of block diagonal matrices whose blocks satisfy (3), and for them we state the following Corollary that will be useful later on.

Corollary 3: Let $S_n \in \mathbb{C}^{n \times n}$ be a non singular matrix such that $\Lambda_{P_n} = S_n^{-1} P_n S_n$ is diagonal. Then if $\mathbf{N} \in \mathfrak{D}^{P_n, p, q}$ we have that

$$\mathcal{N} = (S_n \otimes I_p) \mathbf{N} (S_n \otimes I_q)^{-1} \in \mathfrak{G}^{P_n, p, q}.$$

Proof: The proof is trivial and analogous to the one of Lemma 2. ■

In this paper we focus on linear dynamical systems such that the system matrices of their state space representation are all in the set \mathfrak{G} for the same matrix P_n . We shall consider discrete time systems of the kind

$$\begin{cases} x(k+1) = \mathcal{A}x(k) + \mathcal{B}_w w(k) + \mathcal{B}_u u(k) \\ z(k) = \mathcal{C}_z x(k) + \mathcal{D}_{zw} w(k) + \mathcal{D}_{zu} u(k) \\ y(k) = \mathcal{C}_y x(k) + \mathcal{D}_{yw} w(k) \end{cases} \quad (4)$$

as well as continuous time systems

$$\begin{cases} \dot{x}(t) = \mathcal{A}x(t) + \mathcal{B}_w w(t) + \mathcal{B}_u u(t) \\ z(t) = \mathcal{C}_z x(t) + \mathcal{D}_{zw} w(t) + \mathcal{D}_{zu} u(t) \\ y(t) = \mathcal{C}_y x(t) + \mathcal{D}_{yw} w(t) \end{cases} \quad (5)$$

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 considered in this paper.

Definition 4: Let us consider the linear dynamical systems described by (4) or (5). We call such systems ‘‘decomposable systems’’ if and only if $\mathcal{A} \in \mathfrak{G}^{P_n, l, l}$, $\mathcal{B}_w \in \mathfrak{G}^{P_n, l, m_w}$, $\mathcal{B}_u \in \mathfrak{G}^{P_n, l, m_u}$, $\mathcal{C}_z \in \mathfrak{G}^{P_n, r_z, l}$, $\mathcal{C}_y \in \mathfrak{G}^{P_n, r_y, l}$, $\mathcal{D}_{zw} \in \mathfrak{G}^{P_n, r_z, m_w}$,

$\mathcal{D}_{zu} \in \mathfrak{G}^{P_n, r_z, m_u}$ and $\mathcal{D}_{yw} \in \mathfrak{G}^{P_n, r_y, m_w}$. The reason of this name will be clear later on. Notice that the order of the system is nl , and that $u \in \mathbb{R}^{m_u}$, $w \in \mathbb{R}^{m_w}$, $y \in \mathbb{R}^{r_y}$ and $z \in \mathbb{R}^{r_z}$.

In the case that P_n is symmetric, then we call the system a ‘‘symmetric decomposable system,’’ then S_n is real and orthogonal ($S_n^{-1} = S_n^T$) and Λ_n is real [15].

In this paper we focus on discrete time systems, for reasons that will be clear from Section III. We now present a Theorem which is of fundamental importance for the results shown in this paper.

Theorem 5: A decomposable system of order nl as described in Definition 4 is equivalent to n independent subsystems of order l . Each of these subsystems has only m_u inputs, m_d disturbances, r_z performance outputs and r_y control outputs.

Proof: According to Lemma 2, every matrix \mathcal{M} appearing in the state-space description of the system can be rewritten as

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

with $\mathbf{M} \in \mathfrak{D}^{P_n, p, q}$, block diagonal, with p, q assuming appropriate values. We can then rewrite the system equations (e.g., (4), those of the discrete time case) as

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

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

$$\begin{aligned} x &= (S_n \otimes I_l) \hat{x} \\ w &= (S_n \otimes I_{m_w}) \hat{w} \\ u &= (S_n \otimes I_{m_u}) \hat{u} \\ z &= (S_n \otimes I_{r_z}) \hat{z} \\ y &= (S_n \otimes I_{r_y}) \hat{y} \end{aligned} \quad (6)$$

the system finally becomes

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

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:

$$\begin{cases} \hat{x}_i(k+1) = \mathbf{A}_i \hat{x}_i(k) + \mathbf{B}_{w,i} \hat{w}_i(k) + \mathbf{B}_{u,i} \hat{u}_i(k) \\ \hat{z}_i(k) = \mathbf{C}_{z,i} \hat{x}_i(k) + \mathbf{D}_{zw,i} \hat{w}_i(k) + \mathbf{D}_{zu,i} \hat{u}_i(k) \\ \hat{y}_i(k) = \mathbf{C}_{y,i} \hat{x}_i(k) + \mathbf{D}_{yw,i} \hat{w}_i(k) \end{cases} \quad \text{for } i = 1, \dots, n \quad (7)$$

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 \mathcal{A}); for this reason, we will sometimes call them ‘‘modal subsystems’’ to emphasize this fact. ■

Also notice that according to (2), these systems can be written as

$$\begin{cases} \hat{x}_i(k+1) = (A_a + \lambda_i A_b) \hat{x}_i(k) + \\ \quad + (B_{w,a} + \lambda_i B_{w,b}) \hat{w}_i(k) + (B_{u,a} + \lambda_i B_{u,b}) \hat{u}_i(k) \\ \hat{z}_i(k) = (C_{z,a} + \lambda_i C_{z,b}) \hat{x}_i(k) + \\ \quad + (D_{zw,a} + \lambda_i D_{zw,b}) \hat{w}_i(k) + (D_{zu,a} + \lambda_i D_{zu,b}) \hat{u}_i(k) \\ \hat{y}_i(k) = (C_{y,a} + \lambda_i C_{y,b}) \hat{x}_i(k) + \\ \quad + (D_{yw,a} + \lambda_i D_{yw,b}) \hat{w}_i(k) \end{cases} \quad \text{for } i = 1, \dots, n. \quad (8)$$

Remark 6: The main result on formation stability of [13] can be considered a special case of Theorem 5.

This property means that many control design problems for the class of systems we are considering can be solved by looking at the decomposed problem. This idea of decomposing the problem into a set of smaller problems is at the base of the methods shown in Section III. But before going into these, we first highlight an additional observation that will be of use.

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:

$$\hat{u}_i = \mathbf{K}_i \hat{x}_i \quad \text{for } i = 1, \dots, n \quad (9)$$

or dynamic output feedback controllers:

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

(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’’:

$$u = \mathcal{K}x \quad \text{(static state feedback)} \quad (10)$$

or

$$\begin{cases} x_c(k+1) = \mathcal{A}_c x_c(k) + \mathcal{B}_c y(k) \\ u(k) = \mathcal{C}_c x_c(k) + \mathcal{D}_c y(k) \end{cases} \quad \text{(dynamic output feedback)}. \quad (11)$$

For example, it will be

$$\mathcal{K} = (S_n \otimes I_{m_u}) \mathbf{K} (S_n \otimes I_l)^{-1} \notin \mathfrak{G}^{P_n, m_u, l}.$$

But if the controller matrices are chosen such as they can be parameterized according to (2), then the matrices

$$\begin{aligned} \mathcal{K} &= (S_n \otimes I_{m_u}) \mathbf{K} (S_n \otimes I_l)^{-1} \\ \mathcal{A}_c &= (S_n \otimes I_l) \mathbf{A}_c (S_n \otimes I_l)^{-1} \\ \mathcal{B}_c &= (S_n \otimes I_l) \mathbf{B}_c (S_n \otimes I_{r_y})^{-1} \\ \mathcal{C}_c &= (S_n \otimes I_{m_u}) \mathbf{C}_c (S_n \otimes I_l)^{-1} \\ \mathcal{D}_c &= (S_n \otimes I_{m_u}) \mathbf{D}_c (S_n \otimes I_{r_y})^{-1} \end{aligned} \quad (12)$$

which represent the possible controllers for the untransformed system, will have the same structure as the matrices of the system: thanks to Corollary 3, $\mathcal{K}, \mathcal{A}_c, \mathcal{B}_c, \mathcal{C}_c, \mathcal{D}_c \in \mathfrak{G}^{P_n, \bullet, \bullet}$, 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.

III. DECOMPOSITION APPROACH

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 5. 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, an additional care will be needed, which we are going to show shortly.

For example, let us now consider the problem of finding a stabilizing static state feedback (as in (10)) for the system in (4). The basic LMI approach for solving the problem is to find a feasible solution to the following inequality [22]:

$$\begin{bmatrix} \mathcal{X} & \mathcal{A}\mathcal{X} + \mathcal{B}_u\mathcal{L} \\ * & \mathcal{X} \end{bmatrix} \succ 0 \quad (13)$$

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:

$$\begin{bmatrix} X_i & (A_a + \lambda_i A_b)X_i + (B_{u,a} + \lambda_i B_{u,b})L_i \\ * & X_i \end{bmatrix} \succ 0 \quad (14)$$

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 (12) to get \mathcal{K} , then this \mathcal{K} in general will not be in the set $\mathfrak{G}^{P_n, m_u, l}$.

If we instead want to have $\mathcal{K} \in \mathfrak{G}^{P_n, m_u, l}$, we can solve the set of LMIs in (14) with the following coupling constraints:

$$\begin{aligned} X_i &= X & \text{for } i = 1, \dots, n \\ L_i &= L_a + \lambda_i L_b \end{aligned}$$

and thus, the gains K_i will be parameterized according to (3)

$$K_i = (L_a + \lambda_i L_b)X^{-1} = K_a + \lambda_i K_b$$

thus yielding a $\mathcal{K} \in \mathfrak{G}^{P_n, m_u, l}$. This approach is similar to the so-called *multiobjective optimization* [21]; 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 new result has appeared that allows more generality to these multiobjective optimization problems in discrete time. In [8] it is shown that (13) can be replaced by the equivalent

$$\begin{bmatrix} \mathcal{X} & \mathcal{A}\mathcal{G} + \mathcal{B}_u\mathcal{L} \\ * & \mathcal{G} + \mathcal{G}^T - \mathcal{X} \end{bmatrix} \succ 0 \quad (15)$$

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 (14) is

$$\begin{bmatrix} X_i & (A_a + \lambda_i A_b)G_i + (B_{u,a} + \lambda_i B_{u,b})L_i \\ * & G_i + G_i^T - X_i \end{bmatrix} \succ 0 \quad (16)$$

for $i = 1, \dots, n$

on which we can put the following constraints:

$$\begin{aligned} G_i &= G & \text{for } i = 1, \dots, n \\ L_i &= L_a + \lambda_i L_b. \end{aligned} \quad (17)$$

These constraints still introduce conservatism due to the single matrix G for all the LMIs, but leaves 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 of 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 in the Section III-A-I

Remark 7: There are some cases when the set of n LMIs in (16) coupled by (17) 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 which contain the extreme (maximum and minimum) values of λ_i . Since LMIs are a convex optimization problem, the feasibility of just these two inequalities will automatically grant the feasibility of all the others.

Remark 8: Let us evaluate the reduction in complexity of the problem, by going from its general formulation of (15) to the approach proposed here ((16) together with (17)). 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 (15) 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 $\mathcal{O}(n^2 l^2)$. The number of constraints (the size of the LMI) is $\mathcal{O}(n^2 l^2)$ as well. For (16) and (17), 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. The reduction of complexity is then of the order of the number n , which can be quite big for massively distributed systems.

Moreover, if Remark 7 holds, then the X_i variables become 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 which are discussed in Section IV

IV. CONTROLLER SYNTHESIS

A. A Preliminary Note on System Norms

In this section we will explore the possibility of designing controllers with performance criteria based on their disturbance rejection ability. If we call T_{wz} the transfer function of the system in closed loop, from the disturbance w to the performance output z , then that means that we are trying to minimize a system norm of this transfer function with an appropriate choice of the controller. As the approach shown in this article

is to solve the synthesis problems for the system in its decomposed form, then it is important to understand the relation between the norms of the system in its original form and in the transformed one. It is important to notice that the decomposing transformation (6) 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 9: Let T_{wz} be the transfer function of the system in (4) or (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 (6), from the new disturbance \hat{w} to the new output \hat{z} . Then it holds (for both discrete and continuous time systems)

$$\frac{\underline{\sigma}(S_n)}{\bar{\sigma}(S_n)} \|\hat{T}_{\hat{w}\hat{z}}\|_{\mathcal{H}_2} \leq \|T_{wz}\|_{\mathcal{H}_2} \leq \frac{\bar{\sigma}(S_n)}{\underline{\sigma}(S_n)} \|\hat{T}_{\hat{w}\hat{z}}\|_{\mathcal{H}_2} \quad (18)$$

$$\frac{\underline{\sigma}(S_n)}{\bar{\sigma}(S_n)} \|\hat{T}_{\hat{w}\hat{z}}\|_{\mathcal{H}_\infty} \leq \|T_{wz}\|_{\mathcal{H}_\infty} \leq \frac{\bar{\sigma}(S_n)}{\underline{\sigma}(S_n)} \|\hat{T}_{\hat{w}\hat{z}}\|_{\mathcal{H}_\infty} \quad (19)$$

where $\bar{\sigma}(S_n)$ and $\underline{\sigma}(S_n)$ are respectively the maximum and minimum singular value of S_n .

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

$$\|\hat{T}_{\hat{w}\hat{z}}\|_{\mathcal{H}_2}^2 = \sum_{i=1}^n \|\hat{T}_{\hat{w}_i\hat{z}_i}\|_{\mathcal{H}_2}^2$$

$$\|\hat{T}_{\hat{w}\hat{z}}\|_{\mathcal{H}_\infty} = \max_i \|\hat{T}_{\hat{w}_i\hat{z}_i}\|_{\mathcal{H}_\infty}.$$

Proof: These expressions can be easily obtained from the definitions of the \mathcal{H}_2 and \mathcal{H}_∞ norms [14], using the properties of the Kronecker product [4] and the following bounds for the trace and the maximum singular value of the product of two matrices (see for example [11] and [15]):

$$\underline{\lambda}(A)\text{trace}(B) \leq \text{trace}(AB) \leq \bar{\lambda}(A)\text{trace}(B)$$

(A, B semipositive definite)

$$\bar{\sigma}(AB) \leq \bar{\sigma}(A)\bar{\sigma}(B)$$

where $\bar{\lambda}(A)$ and $\underline{\lambda}(A)$ are respectively the maximum and minimum eigenvalue of A . We omit the complete derivation for the sake of brevity. ■

Remark 10: If we have a symmetric decomposable system, then S_n is orthogonal: this means $\bar{\sigma}(S_n)/\underline{\sigma}(S_n) = 1$. So, according to (18) and (19), the system norms are the same for $\hat{T}_{\hat{w}\hat{z}}$ and T_{wz} .

In the remainder of the paper we will consider only symmetric decomposable systems. These systems are easier to treat for two reasons: first, as seen in this 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 rest of the Section can be easily generalized to any decomposable system; extra care will only be needed to handle LMIs with complex parameters (that still have to yield real solutions) as well as the fact that the bounds we can impose to the norms have to be scaled by the factor $\bar{\sigma}(S_n)/\underline{\sigma}(S_n)$.

B. Controller Synthesis: Static State Feedback

The method shown in Section III 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. We first look for static state feedback controllers which have the same structure as the system; they can be expressed by (10), with

$$\mathcal{K} = I_n \otimes K_a + P_n \otimes K_b. \quad (20)$$

We adapt the results of [8] 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 the previous Section is understood.

Theorem 11: Consider a discrete time symmetric decomposable system (Definition 4). A sufficient condition for the existence of a static state feedback controller described by (9) and (20) that yields a $\|T_{wz}\|_{\mathcal{H}_\infty} < \gamma$ is that the following set of LMIs has a feasible solution:

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

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 that the matrices of index i which are parameterized according to (3) have been written in bold, like for example \mathbf{L}_i , \mathbf{A}_i , $\mathbf{B}_{u,i}$ in the last equation. We will use this convention in the remainder of the paper.

Theorem 12: Consider a discrete time symmetric decomposable system (Definition 4), with $\mathbf{D}_{zw} = 0$. A sufficient condition for the existence of a static state feedback controller described by (9) and (20) that yields a $\|T_{wz}\|_{\mathcal{H}_2} < \gamma$ is that the following set of LMIs has a feasible solution:

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

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

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 13: As for Remark 7, there are situations when the sets of LMIs in Theorem 11 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 12, by replacing the bound on the sum of the traces of W_i with a bound on the sum of the traces of only the two W_i matrices involved.

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

$$\begin{cases} x(k+1) = (A + \Delta)x(k) + B_w w(k) + B_u u(k) \\ y(k) = Cx(k) + D_w w(k) + D_u u(k) \end{cases} \quad (21)$$

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

$$\Delta = I_n \otimes (V_a \delta W_a) + P_n \otimes (V_b \delta W_b) = \Delta_0 \delta$$

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

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

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

$$\mathbf{\Delta}_i = \delta \mathbf{V}_i \mathbf{W}_i$$

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}_∞ performances with robust stability, by solving the following optimization problem:

$$\begin{aligned} & \text{minimize } \gamma^2 \text{ over :} \\ & \left[\begin{array}{cccc} P_i & \mathbf{A}_i X + \mathbf{B}_{u,i} \mathbf{L} & \mathbf{B}_{w,i} & 0 \\ * & X + X^T - P_i & 0 & X^T \mathbf{C}_i^T + \mathbf{L}^T \mathbf{D}_{u,i}^T \\ * & * & I_{m_w} & \mathbf{D}_{w,i}^T \\ * & * & * & \gamma^2 I_{r_z} \end{array} \right] \succ 0 \\ & \left[\begin{array}{ccc} Q_i & \mathbf{A}_i X + \mathbf{B}_{u,i} \mathbf{L} & \mathbf{W}_i \\ * & X + X^T - Q_i & 0 \\ * & * & 1 \\ * & * & * \end{array} \right] \succ 0 \quad (22) \\ & \text{for } i = 1, \dots, n \end{aligned}$$

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 performances, while the second imposes that the \mathcal{H}_∞ of the transfer function from the output of the uncertainty (which has been ‘‘pulled out’’) to its input is smaller than 1. This grants the robust stability as a consequence of the small gain theorem [29].

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, 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.

C. Controller Synthesis: Dynamic Output Feedback

The method can be used also for dynamic output feedback. Let us first start by reporting the general result as in [8], for \mathcal{H}_∞ . Let us assume we have a generic system of the kind

$$\begin{cases} x(k+1) = Ax(k) + B_w w(k) + B_u u(k) \\ z(k) = C_z x(k) + D_{zw} w(k) + D_{zu} u(k) \\ y(k) = C_y x(k) + D_{yw} w(k) \end{cases} \quad (23)$$

for which we want to create a controller

$$\begin{cases} x_c(k+1) = A_c x_c(k) + B_c y(k) \\ u(k) = C_c x_c(k) + D_c y(k) \end{cases} \quad (24)$$

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

$$\begin{cases} D_c = R \\ C_c = (L - RC_y X) U^{-1} \\ B_c = V^{-1} (F - Y B_u R) \\ A_c = V^{-1} (Q - Y (A + B_u D_c C_y) X - V B_c C_y X) U^{-1} + \\ \quad - V^{-1} Y B_u C_c \end{cases} \quad (25)$$

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, and solve the n independent LMIs, each one with its own decision variables. These LMIs are shown in (26) at the top of next page [(27) shows the \mathcal{H}_2 case].

It is quite easy to see that under certain assumptions there is a parameterization of the decision variables such that $\mathbf{A}_c, \mathbf{B}_c, \mathbf{C}_c, \mathbf{D}_c \in \mathfrak{D}^{P_n, \bullet, \bullet}$, yielding a controller in the untransformed domain that will be of the same structure as the plant. The parameterization is the following:

$$\begin{cases} X_i = X \\ Y_i = Y \\ S_i = S \\ \mathbf{L}_i = L_a + \lambda_i L_b \\ \mathbf{F}_i = F_a + \lambda_i F_b \\ \mathbf{Q}_i = Q_a + \lambda_i Q_b \\ \mathbf{R}_i = R_a + \lambda_i R_b \\ P_i = P_i^T, H_i = H_i^T, J_i \text{ unconstrained} \end{cases} \quad \text{for } i = 1, \dots, n \quad (28)$$

that together with

$$\begin{cases} VU = S - YX \text{ (with } U, V \text{ non singular)} \\ \mathbf{D}_{c,i} = \mathbf{R}_i \\ \mathbf{C}_{c,i} = (\mathbf{L}_i - \mathbf{R}_i \mathbf{C}_{y,i} X) U^{-1} \\ \mathbf{B}_{c,i} = V^{-1} (\mathbf{F}_i - Y \mathbf{B}_{u,i} \mathbf{R}_i) \\ \mathbf{A}_{c,i} = V^{-1} (\mathbf{Q}_i - Y (\mathbf{A}_i + \mathbf{B}_{u,i} \mathbf{R}_i \mathbf{C}_{y,i}) X + \\ \quad - V \mathbf{B}_{c,i} \mathbf{C}_{y,i} X) U^{-1} - V^{-1} Y \mathbf{B}_{u,i} \mathbf{C}_{c,i} \end{cases} \quad (29)$$

and (12) will yield $\mathbf{A}_c, \mathbf{B}_c, \mathbf{C}_c, \mathbf{D}_c \in \mathfrak{G}^{P_n, \bullet, \bullet}$; 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

$$\begin{aligned} \mathbf{C}_{c,i} &= (\mathbf{L}_i - \mathbf{R}_i \mathbf{C}_{y,i} X) U^{-1} \\ &= (L_{c,a} + \lambda_i L_{c,b} - (R_a + \lambda_i R_b) \\ &\quad \times (C_{y,a} + \lambda_i C_{y,b}) X) U^{-1}. \end{aligned}$$

So if we want $\mathbf{C}_{c,i}$ to be parameterized as $\mathbf{C}_{c,i} = C_{c,a} + \lambda_i C_{c,b}$, then we either need to have $\mathbf{C}_{y,i}$ constant ($C_{y,b} = 0$) or to set

$$\begin{bmatrix} P_i & J_i & \mathbf{A}_i X_i + \mathbf{B}_{u,i} \mathbf{L}_i & \mathbf{A}_i + \mathbf{B}_{u,i} \mathbf{R}_i \mathbf{C}_{y,i} & \mathbf{B}_{w,i} + \mathbf{B}_{u,i} \mathbf{R}_i \mathbf{D}_{yw,i} & 0 & 0 \\ * & H_i & \mathbf{Q}_i & Y_i \mathbf{A}_i + \mathbf{F}_i \mathbf{C}_{y,i} & Y_i \mathbf{B}_{w,i} + \mathbf{F}_i \mathbf{D}_{yw,i} & 0 & 0 \\ * & * & X_i + X_i^T - P_i & I_l + S_i^T - J_i & 0 & X_i^T \mathbf{C}_{z,i}^T + \mathbf{L}_i^T \mathbf{D}_{zu,i}^T & \\ * & * & * & Y_i + Y_i^T - H_i & 0 & \mathbf{C}_{z,i}^T + \mathbf{C}_{y,i}^T \mathbf{R}_i^T \mathbf{D}_{zu,i}^T & \\ * & * & * & * & I_{m_w} & \mathbf{D}_{zw,i}^T + \mathbf{D}_{yw,i}^T \mathbf{R}_i^T \mathbf{D}_{zu,i}^T & \\ * & * & * & * & * & \gamma^2 I_{r_z} & \end{bmatrix} \succ 0 \quad (26)$$

for $i = 1, \dots, n$

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

$$\begin{bmatrix} P_i & J_i & \mathbf{A}_i X_i + \mathbf{B}_{u,i} \mathbf{L}_i & \mathbf{A}_i + \mathbf{B}_{u,i} \mathbf{R}_i \mathbf{C}_{y,i} & \mathbf{B}_{w,i} + \mathbf{B}_{u,i} \mathbf{R}_i \mathbf{D}_{yw,i} \\ * & H_i & \mathbf{Q}_i & Y_i \mathbf{A}_i + \mathbf{F}_i \mathbf{C}_{y,i} & Y_i \mathbf{B}_{w,i} + \mathbf{F}_i \mathbf{D}_{yw,i} \\ * & * & X_i + X_i^T - P_i & I_l + S_i^T - J_i & 0 \\ * & * & * & Y_i + Y_i^T - H_i & 0 \\ * & * & * & * & I_{m_w} \end{bmatrix} \succ 0 \quad (27)$$

$\mathbf{D}_{zw,i} + \mathbf{D}_{zu,i} \mathbf{R}_i \mathbf{D}_{yw,i} = 0$
for $i = 1, \dots, n$, and :
 $\sum_{i=1}^n \text{trace}(W_i) < \gamma^2$

TABLE I
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	$C_{y,b} = 0$	$R_b = 0$

\mathbf{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 I.

We summarize this result in the following Theorem.

Theorem 14: Consider a discrete time symmetric decomposable system (Definition 4), in one of the cases of Table I. A sufficient condition for the existence of a decomposable dynamic output feedback controller described by (24) that yields a $\|T_{wz}\|_{\mathcal{H}_\infty} < \gamma$ is that the set of LMI constraints in (26) has a feasible solution. The decision variables and their parametrization are shown in (28), while Table I shows the additional constraints which might be needed. The state space matrices of the controller can be retrieved through (29).

With analogous reasoning, it is possible to adapt the results of [8] for \mathcal{H}_2 performance; we state this result in this last Theorem.

Theorem 15: Consider a discrete time symmetric decomposable system (Definition 4), in one of the cases of Table I. A sufficient condition for the existence of a decomposable dynamic output feedback controller described by (24) that yields a $\|T_{wz}\|_{\mathcal{H}_2} < \gamma$ is that the set of LMI constraints in (27) has a feasible solution. The decision variables and their parametrization are the ones shown in (28), with the addition of $W_i = W_i^T$; Table I shows the additional constraints which might be needed. The state space matrices of the controller can be retrieved through (29).

Remark 16: As for Remark 7, also for the LMIs in Theorem 14 and Theorem 15 a reduction can be done in the case of $B_{u,b} = 0$, $D_{zu,b} = 0$, $D_{yw,b} = 0$.

V. GRAPH THEORY

In the Introduction we mentioned graph theory as a way of interpreting the systems which are object of this paper. 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 (1) it is apparent that the same matrix \mathcal{M} can be obtained with different P_n '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 P_n which has bounded eigenvalues. Following the same reasoning as in [13], then it can be convenient to choose it as a graph Laplacian matrix. It is out of the scope of this article to give a complete account of graph theory and of the properties of Laplacian matrices; we invite the reader to refer to [13] for the notions which are relevant for our purpose, and the next paragraph is a brief summary of what is reported there.

For our scope it can be sufficient for the reader to know that Laplacian matrices are an algebraic way of describing a graph. A Laplacian matrix \mathcal{L}_n of order n is a square sparse matrix for which the entry in the i th row and k th column satisfies

$$\mathcal{L}_n(i, k) = \begin{cases} 1 & \text{if } i = k \\ \text{either } 0 \text{ or } -\frac{1}{J_i} & \text{if } i \neq k \end{cases} \quad (30)$$

where J_i is the number of non-zero off-diagonal elements in the i th row. These properties imply that the columns of a Laplacian sum up to 0. For symmetric Laplacian matrices it holds that all the eigenvalues λ_i are real and situated between 0 and 2

$$0 \leq \lambda_i \leq 2.$$

This then explains why it can be useful to choose (if possible) a symmetric Laplacian matrix as pattern matrix: for a symmetric Laplacian, 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 13 or 16 holds, we can assume directly 0 as minimum eigenvalue and 2 as the maximum one. Secondly, as these boundaries hold for all Laplacians, then the result of this controller synthesis will be valid regardless of the number of interconnected elements n , as long as the pattern is a symmetric Laplacian matrix. These statements will be made clearer by the first example in Section VII.

VI. A GENERAL DECOMPOSITION APPROACH

A question that comes natural by looking at Theorem 5 is whether such a decomposition property is possible for other kind of systems as well. For example, we know from literature (see [5]) that any circulant system can be decomposed, but Theorem 5 in its form does not accommodate all the possible circulant systems (although P_n can be circulant). Another question would be whether systems that have state space matrices constructed with more than one pattern matrix, e.g.

$$\mathcal{M} = I_n \otimes M_a + P_{n_1} \otimes M_{b_1} + P_{n_2} \otimes M_{b_2}$$

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. We stay in a discrete time setting to keep the style of the rest of the paper, but of course an equivalent version for continuous time can be given.

Theorem 17 (General Decomposition Theorem): Consider a state space system of equations as in (4). Assume that all the state space matrices (\mathcal{A} , \mathcal{B}_w , etc.) can be expressed as:

$$\mathcal{M} = \sum_{k=1}^{\mu} P_n^{(k)} \otimes M^{(k)} \quad (31)$$

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

Then the system is equivalent to a set of n systems of equations as in (7), where all the state space matrices (\mathbf{A}_i , $\mathbf{B}_{w,i}$, etc.) are

$$\mathbf{M}_i = \sum_{k=1}^{\mu} \lambda_i^{(k)} M^{(k)} \quad (32)$$

where the matrices $M^{(k)}$ are the same which appear in (31), and $\lambda_i^{(k)}$ is the i th eigenvalue of $P_n^{(k)}$ (i.e., the i th entry of the diagonal matrix $S_n^{-1}P_n^{(k)}S_n$).

Conversely, a set of n systems as in (7) is equivalent to a system as in (4) with matrices as in (31), if all the matrices in the n systems are parameterized according to (32).

Proof: The proof follows the same reasoning as the one for Theorem 5. ■

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 [15], so a simple rule for having multiple patterns is to choose $P_n^{(k)}$ matrices which commute. Theorem 5 is the special case of this last Theorem for $P_n^{(1)} = I_n$ and $P_n^{(2)} = P_n$; as I_n commutes with any matrix, then any diagonalizable P_n is acceptable.

The LMI procedures shown in Section IV can then be naturally extended to accommodate any number of patterns; simply by redefining “bold” decision variables as in (32), then formally the same LMIs can be used to search for controllers. Remarks 7, 13 and 16 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 $\lambda_i^{(k)}$ which are at the vertices of the convex hull of all the possible μ -uples of them. This last sentence will be made clearer in the first example of Section VII, where we will briefly show the synthesis of a controller with multiple pattern matrices.

Also the theory of SVD controllers [16] partially fits into the framework of Theorem 17; in general, an SVD controller can be designed for plants whose transfer function T can be factorized as $T = V\Sigma U^T$, where Σ is a diagonal matrix of transfer functions, and V and U are real (static) unitary matrices. It can be easily shown that Theorem 17 can be applied in the case of $V = U$.

At last, we briefly discuss two very special cases of systems which have been described in literature, and for which the general decomposition theorem applies. We have also to stress that these are not the only possibilities, in fact the second example of Section VII will show a system which is neither of the two.

A. Symmetrically Interconnected Systems

Symmetrically interconnected systems [17] can be represented as having a state space realization with matrices of the kind

$$\mathcal{M} = I_n \otimes M_a + \mathcal{L}_n^{(c)} \otimes M_b$$

where $\mathcal{L}_n^{(c)}$ is the Laplacian of the complete graph (i.e., the graph with all the possible interconnections)

$$\mathcal{L}_n^{(c)} = \begin{bmatrix} 1 & -\frac{1}{n-1} & -\frac{1}{n-1} & \cdots & -\frac{1}{n-1} \\ -\frac{1}{n-1} & 1 & -\frac{1}{n-1} & \cdots & -\frac{1}{n-1} \\ -\frac{1}{n-1} & -\frac{1}{n-1} & 1 & \cdots & -\frac{1}{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{n-1} & -\frac{1}{n-1} & -\frac{1}{n-1} & \cdots & 1 \end{bmatrix}.$$

This kind of matrix is always diagonalizable (as it is circulant, [7]); however, it has only two distinct eigenvalues, 0 with multiplicity 1 and $n/n - 1$ with multiplicity $n - 1$ [13]. This means that decomposing this system will result in n modal subsystems, $n - 1$ of which are all the same (this is not a consequence of using a Laplacian matrix to represent the system, but it holds in general for all symmetrically interconnected systems, as it was already pointed out in [17]). So symmetrically interconnected systems are inherently described by only two subsystems.

It is also true that optimal controllers for symmetrically interconnected systems are inherently symmetrically interconnected too; in fact, the synthesis method shown in this paper is not necessary for this kind of systems. In Section IV 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 make an example to clarify. 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

$$\begin{aligned} K_1 &= K_a \\ K_{2\dots n} &= K_a + \frac{n}{n-1} K_b. \end{aligned}$$

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 [17]).

B. Circulant Systems

Circulant systems [5] can be represented as having state space realization with block circulant matrices [7]. Such matrices fit the description of (31), as they can be expressed as

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

where Π_n is the permutation matrix

$$\Pi_n = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} = \left[\begin{array}{c|c} 0 & I_{n-1} \\ \hline 1 & 0 \end{array} \right].$$

Permutation matrices (Π_n and its powers) are all diagonalizable and all commute with one another, so they can be simultaneously diagonalized by a matrix F_n . This matrix is well known in literature and it is called Fourier matrix [7].

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 been already explored [5], [9]. However, it can be useful to use the method shown in this paper 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

$$\mathcal{M} = \sum_{k=-b}^b \Pi_n^k \otimes M^{(k)}$$

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

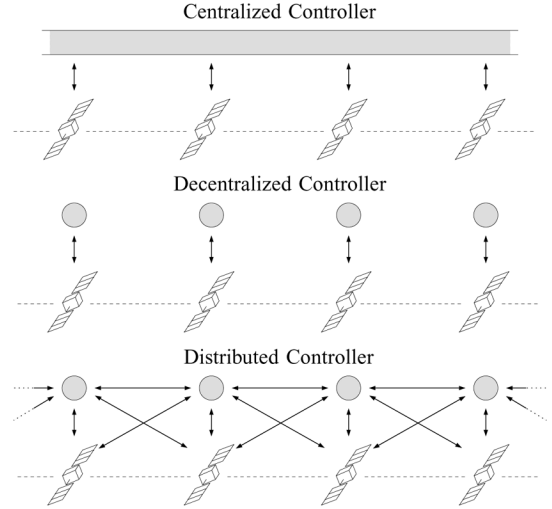


Fig. 2. Three different types of controllers. The arrows represent information flow.

VII. EXAMPLES

A. Satellite Formation Flying

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. This is typically a formation flying problem. Let us consider a swarm of satellites orbiting around a planet on a circular orbit (a similar example is shown in [12]). The small perturbations of their motion with respect to the nominal circular trajectory are described by the so-called Clohessy–Wiltshire equations [19]

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

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 which is 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; 3) a distributed controller made of the interaction of local controllers which can communicate with the nearest neighbors. Fig. 2 visually shows the difference between these types of controllers.

The last of the three controllers can be designed with the method shown in this article. We said that the goal is designing a controller for minimizing the relative errors on the 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 \mathcal{C}_z . As we need to

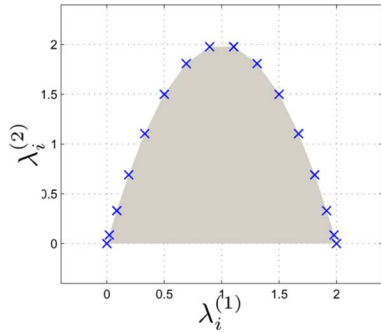


Fig. 3. Eigenvalues of $L_n^{(1)}$ versus those of $L_n^{(2)}$ ($n = 30$). The shaded area is the convex hull of these points: the vertices of the convex hull are all the points themselves in this case.

put a penalty on the difference of the position, we can choose as performance output the following:

$$z_{x_k,i} = -\frac{1}{2}x_{k,i-1} + x_{k,i} - \frac{1}{2}x_{k,i+1} \quad \text{for } k = 1, 2, 3.$$

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 $x_{k,i}(t) - x_{k,i+1}(t)$ 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 V, Laplacian matrices are a better choice, and in this case it is possible to use a symmetric Laplacian matrix which we call $\mathcal{L}_n^{(1)}$:

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

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

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

$$z_{a_k,i} = wa_k \quad \text{for } k = 1, 2, 3$$

where w is a weighting parameter.

We turned the problem into a discrete time problem: we use Theorem 14, and as we are in the case of Remark 16, the output feedback synthesis problem can be solved with a reduced set of only two LMIs. Moreover, as $\mathcal{L}_n^{(1)}$ is a symmetric 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

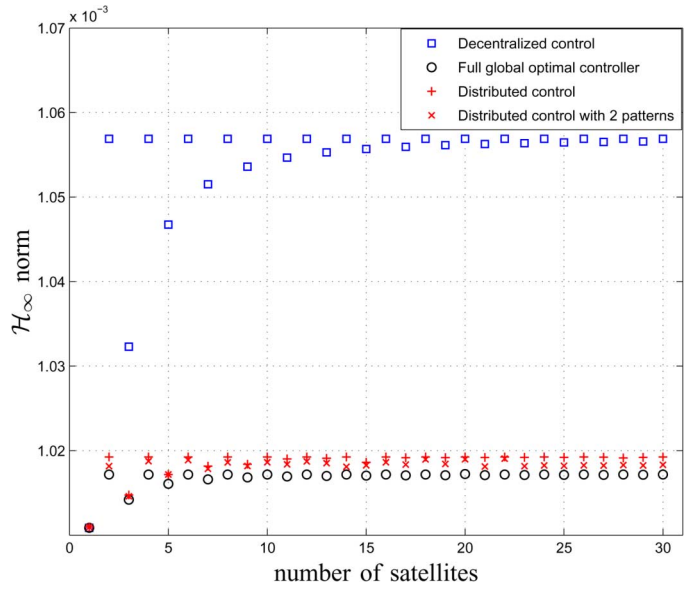


Fig. 4. \mathcal{H}_∞ norm with the four different controllers, for different numbers n of satellites.

with those of of index $i + 2$ and $i - 2$. This can be done by introducing a second pattern matrix as in Section VI:

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

This is again a symmetric Laplacian, and it is a valid choice for a second pattern matrix¹ as it is circulant too, and all circulant matrices commute [7]. However, as explained in Section VI, Remark 16 cannot be applied in its form anymore, as the eigenvalues $\lambda_i^{(1)}$ and $\lambda_i^{(2)}$ of $L_n^{(1)}$ and $L_n^{(2)}$ respectively must not be considered on their own but as couples $(\lambda_i^{(1)}, \lambda_i^{(2)})$. The set of LMIs can be 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 Fig. 3, no reduction is possible in this case and all the n LMIs have to be considered.

We solved the computations for the synthesis of the two distributed controllers described above using Matlab and SeDuMi as solver [1]; we computed also an optimal centralized controllers and a decentralized one for comparison. The results of the computations, for different numbers of satellites, are shown in Fig. 4. As it was expected, the centralized optimal control offers the best performances, 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.

Fig. 5 shows the sparsity of the four different controllers of Fig. 4.

B. Paper Machines

Another example of a system that can be analyzed with the methods shown in this article comes from the cross-directional

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

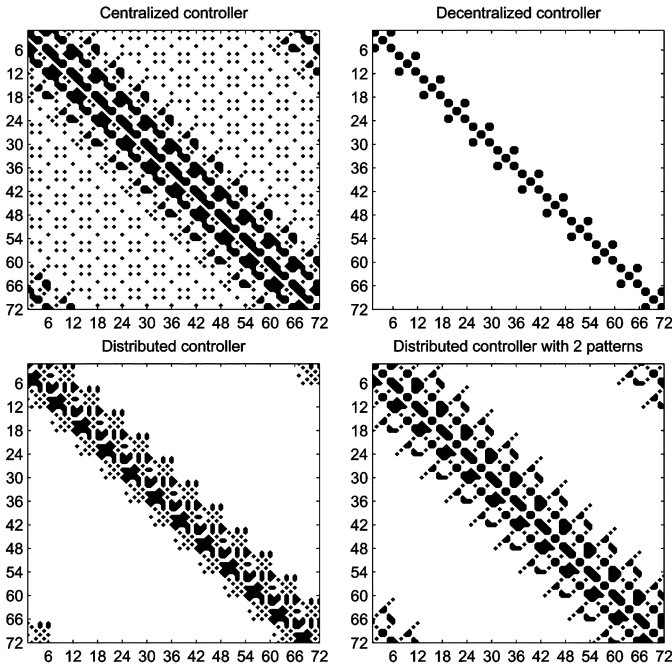


Fig. 5. Sparsity of the state matrix of the four different controllers, for $n = 12$. A black dot indicates a non-zero entry.

control of paper machines [23]. 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 $y(k) \in \mathbb{R}^n$, then a model for the system can be

$$y(k) = \mathcal{P}u(k-d) + ay(k-1) + w(k) \quad (33)$$

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 [25], with some adaptations) assume \mathcal{P} as a generic band matrix, while others (like [23]) 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 [18], [23], or to use a centralized optimal SVD controller which uses the decomposition to simplify the complexity of the computations [25]. With our approach, we will find a suboptimal controller that can be implemented as distributed; Fig. 6 shows the difference between the structure of this controller and of an SVD one.

For using the methods of this article, we need to turn the model of the machine into a state space formulation as in (4), with the result that \mathcal{P} becomes the pattern matrix. We prefer

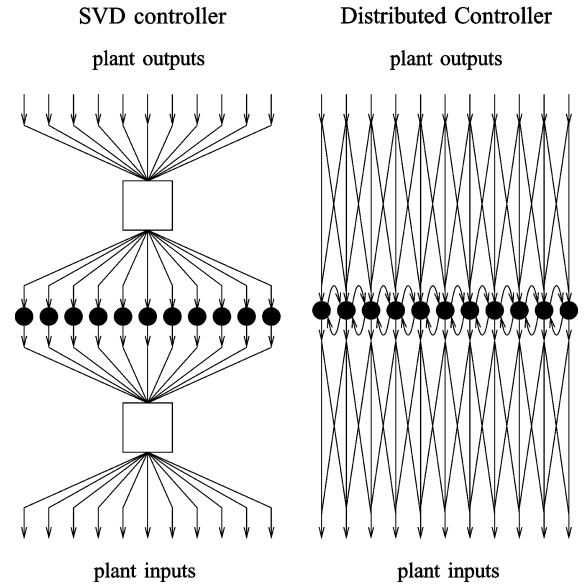


Fig. 6. Difference between an SVD controller and a distributed controller. The boxes represent static transformations, while the circles are dynamic controllers. The SVD controller still needs to handle all the inputs and outputs in the same processing units, so it can be considered as a kind of centralized controller.

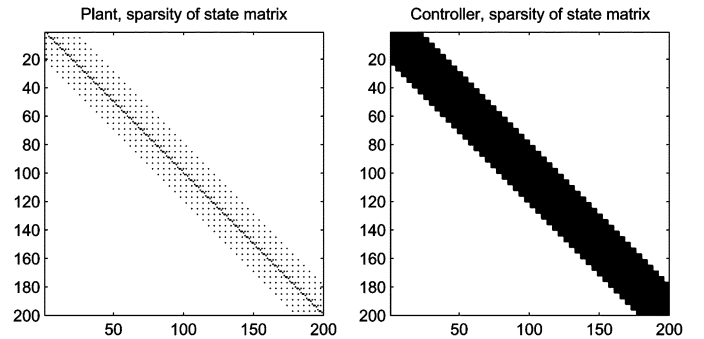


Fig. 7. Sparsity of the state matrix of the plant (\mathcal{A}) and of the controller, for $n = 50$. A black dot indicates a non-zero entry.

working with a symmetric \mathcal{P} , so we chose the numerical model from [23]; more generic models could still be handled, but they would require the use of LMIs with complex values, as explained in Section IV.

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 less restrictions in the synthesis problem (see Table I), and we will also be able to use the reduced method (Remark 16).

This time we have used \mathcal{H}_2 synthesis for different values of n . The results are shown in Table II. Fig. 7 gives an idea of the structure of the systems, showing the sparsity pattern of the state matrix of the plant and the controller.

VIII. CONCLUSION

In this article, we have presented new methods for designing distributed controllers for a special class of systems, which can be considered as the interconnection of n identical subsystems. Thanks to the properties of the matrices in the state space formulation of these systems, it is possible to decompose synthesis

TABLE II
CONTROLLER PERFORMANCES 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

problems into a set of smaller ones, thus reducing the computation time. Moreover, by constraining the possible results of the synthesis for the smaller problems, it is possible to keep the distributed structure of the plant in the controller. These constraints can be easily introduced if the synthesis is expressed in terms of LMIs and a parameterization of the decision variables is introduced. We have shown that the complexity of the design problems, with respect to a centralized controller, is reduced by a factor n or even n^2 in certain cases. The methods have then been applied to two examples, one in satellite formation flying and the other in paper machine control.

ACKNOWLEDGMENT

The authors would like to thank Dr. T. Keviczky for his very constructive comments and suggestions.

REFERENCES

- [1] SeDuMi Home Page [Online]. Available: <http://sedumi.mcmaster.ca/>
- [2] J. Anthonis and H. Ramon, "Linear mechanical systems and dyadic transfer function matrices," *Automatica*, vol. 8, pp. 1353–1363, 2003.
- [3] F. Borrelli and T. Keviczky, "Distributed LQR design for dynamically decoupled systems," in *Proc. 45th IEEE Conf. Decision Control*, San Diego, CA, Dec. 2006, pp. 5639–5644.
- [4] J. W. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Trans. Circuits Syst.*, vol. CS-25, no. 9, pp. 772–781, Sep. 1978.
- [5] R. W. Brockett and J. L. Willems, "Discretized partial differential equations: Examples of control systems defined on modules," *Automatica*, vol. 10, no. 4, pp. 507–515, 1974.
- [6] J. R. Carpenter, "A preliminary investigation of decentralized control for satellite formations," in *Proc. IEEE Aerosp. Conf.*, Mar. 2000, vol. 7, pp. 63–74.
- [7] P. J. Davis, *Circulant Matrices*. New York: Wiley-Interscience, 1979.
- [8] M. C. de Oliveira, J. C. Geromel, and J. Bernoussou, "An LMI optimization approach to multiobjective controller design for discrete-time systems," in *Proc. 38th Conf. Decision Control*, Phoenix, AZ, Dec. 1999, pp. 3611–3616.
- [9] N. Denis and D. P. Looze, " H_∞ controller design for systems with circulant symmetry," in *Proc. 38th Conf. Decision Control*, Phoenix, AZ, Dec. 1999, pp. 3144–3149.
- [10] R. Diestel, *Graph Theory*. Berlin, Germany: Springer, 1996.
- [11] Y. Fang, K. A. Loparo, and X. Feng, "Inequalities for the trace of matrix product," *IEEE Trans. Automat. Control*, vol. 39, no. 12, pp. 2489–2490, Dec. 1994.
- [12] J. A. Fax, "Optimal and Cooperative Control of Vehicle Formations," Ph.D. dissertation, California Institute of Technology, Pasadena, 2002.
- [13] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Trans. Automat. Control*, vol. 49, no. 9, pp. 1465–1476, Sep. 2004.
- [14] W. K. Gawronski, *Advanced Structural Dynamics and Active Control of Structures*. Berlin, Germany: Springer, 1998.
- [15] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985.
- [16] M. Hovd, R. D. Braatz, and S. Skogestad, "SVD controllers for \mathcal{H}_2 -, \mathcal{H}_∞ -, and μ -optimal control," *Automatica*, vol. 33, no. 3, pp. 433–439, 1994.
- [17] M. Hovd and S. Skogestad, "Control of symmetrically interconnected plants," *Automatica*, vol. 30, no. 6, pp. 957–973, 1994.
- [18] D. L. Laughlin, M. Morari, and R. D. Braatz, "Robust performance of cross-directional basis-weight control in paper machines," *Automatica*, vol. 29, no. 6, pp. 1395–1410, 1993.

- [19] B. V. Rauschenbakh, M. Y. Ovchinnikov, and S. M. P. McKenna-Lawlor, *Essential Spaceflight Dynamics and Magnetospherics*. Norwell, MA: Kluwer, 2003.
- [20] N. Sandell, P. Varaiya, M. Athans, and M. Safonov, "Survey of decentralized control methods for large scale systems," *IEEE Trans. Automat. Control*, vol. AC-23, no. 2, pp. 108–128, Apr. 1978.
- [21] C. Scherer, P. Gahinet, and M. Chilali, "Multiobjective output-feedback control via LMI optimization," *IEEE Trans. Automat. Control*, vol. 42, no. 7, pp. 896–911, Jul. 1997.
- [22] C. Scherer and S. Weiland, "Linear Matrix Inequalities in Control," [Online]. Available: <http://www.dcsce.tudelft.nl/~cscherer/2416/lmi05.pdf>
- [23] G. E. Stewart, D. M. Gorinevsky, and G. A. Dumont, "Feedback controller design for a spatially distributed system: The paper machine problem," *IEEE Trans. Control Syst. Technol.*, vol. 11, no. 5, pp. 612–628, Sep. 2003.
- [24] M. K. Sundareshan and R. M. Elbanna, "Large-scale systems with symmetrically interconnected subsystems: Analysis and synthesis of decentralized controllers," in *Proc. 29th Conf. Decision Control*, Honolulu, HI, Dec. 1990, pp. 1137–1142.
- [25] J. G. VanAntwerp, A. P. Featherstone, and R. D. Braatz, "Robust cross-directional control of large scale sheet and film processes," in *J. Process Control*, 2001, vol. 11, pp. 149–177.
- [26] S. Wang and E. J. Davison, "On the stabilization of decentralized control systems," *IEEE Trans. Automat. Control*, vol. AC-18, no. 5, pp. 473–478, Oct. 1973.
- [27] M. J. Woo and J. W. Choi, "A relative navigation system for vehicle platooning," in *Proc. 40th SICE Annu. Conf.*, Jul. 2001, pp. 28–31.
- [28] Y. Yam, J. H. Lang, D. H. Staelin, and T. L. Johnson, "The experimental computer control of a two-dimensional hyperbolic system," *IEEE Trans. Automat. Control*, vol. AC-33, no. 1, pp. 79–87, Jan. 1988.
- [29] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Englewood Cliffs, NJ: Prentice Hall, 1996.



Paolo Massioni was born in Milan, Italy, in 1980. He received the M.Sc. degree (with honors) in aerospace engineering from the Politecnico di Milano, Milan, Italy, in 2005 and is currently pursuing the Ph.D. degree at the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands.

In 2005, he had a short traineeship period in the European Space Research and Technology Centre (ESTEC), Noordwijk, The Netherlands. His main research interests are control and identification of

distributed or large scale systems, subspace identification, satellite attitude control, and formation flying.



Michel Verhaegen received the M.Sc. degree (with honors) in aeronautics from the Delft University of Technology, Delft, The Netherlands, in 1982 and the Ph.D. degree in applied sciences from the Catholic University, Leuven, Belgium, in 1985.

During his graduate study, he held a Research Assistantship sponsored by the Flemish Institute for scientific research (IWT). From 1985 to 1994 he was a two-year Research Fellow of the U.S. National Research Council (NRC), affiliated with the NASA Ames Research Center in California, and a five-year

Research Fellow of the Dutch Academy of Arts and Sciences, affiliated with the Network Theory Group of the Delft University of Technology. From 1994 to 1999, he was an Associate Professor of the Control Laboratory, Delft University of Technology and became appointed as Full Professor at the Faculty of Applied Physics, University of Twente, Twente, The Netherlands, in 1999. In 2001, he moved back to the University of Delft and is now a member of the Delft Center for Systems and Control. He has held short sabbatical leaves at the University of Uppsala, McGill, Lund and the German Aerospace Research Center (DLR), Munich, Germany, and is participating in several National and European Research Networks. In 2007, he was appointed Program Leader of the Dutch National Science Foundation "Perspective" program on Smart Optics. His main research interest is in the interdisciplinary domain of numerical linear algebra and system theory. In this field he has published over 100 papers. Current activities focus on the transfer of knowledge about new identification, fault tolerant control and data driven controller design methodologies to research laboratories and industry. Application areas include smart structures, adaptive optics, wind energy, and vehicle mechatronics.