

# Distributed Control: A Sequentially Semi-Separable Approach for Spatially Heterogeneous Linear Systems

Justin K. Rice, *Student Member, IEEE*, and Michel Verhaegen

**Abstract**—We consider the problem of designing controllers for spatially-varying interconnected systems distributed in one spatial dimension. The matrix structure of such systems can be exploited to allow fast analysis and design of centralized controllers with simple distributed implementations. Iterative algorithms are provided for stability analysis,  $H_\infty$  analysis and sub-optimal controller synthesis. For practical implementation of the algorithms, approximations can be used, and the computational efficiency and accuracy are demonstrated on an example.

**Index Terms**—Distributed control, large scale systems.

## I. INTRODUCTION

THE control of spatially distributed interconnected systems has been of great interest in practical applications involving discretized partial differential equations (PDE's), such as boundary layer and transition control in fluid mechanics [1], [2], flexible structures [3], heat conduction [4] and also in networks of spatially discrete interconnected subsystems, such as highway traffic control [5] and vehicle platooning [6], iterative circuit networks [7], building anti-earthquake systems [8], aircraft and satellite formation flight [9], [10], large adaptive telescope mirrors [11], image processing [12], paper processing [13], irrigation networks [14], etc.

The challenge has been in the computational cost of designing effective controllers and the complexity of implementing them. For the systems of PDE's, when directly solving for the optimal control is not viable, the system is often approximated as a large but finite number of coupled ODE's or interconnected subsystems. The system matrix describing the input-state-output behavior of  $N$  interconnected subsystems (ODE's), each of size (order)  $n$ , will be  $nN \times nN$ , and thus most matrix operations will be  $\mathcal{O}(n^3N^3)$  floating point operations, making traditional robust or optimal controller design prohibitively expensive for fine discretizations or large numbers of discrete subsystems. To surmount this obstacle, three main approaches have arisen.

1) *The Structured Matrix Approach*: In [15], finitely many spatially invariant subsystems connected in a certain topology are shown to generate system matrices with an easily diagonalizable structure. This allows for decoupling transformations and

corresponding control design of  $\mathcal{O}(n^3N)$  computational complexity (where  $N$  controllers, each of order  $n$  are designed). In [16] and [17] this idea is generalized to systems decouplable by SVD and to dyadic transfer matrices. When the system matrices have a special, usually spatially invariant, structure, e.g. circulant (see [18]–[20] for other structures and extensions), the decoupling transformation can be done very efficiently, and the results can be extended to many linear optimal and robust control results for such large scale systems, as in [21], where a computationally efficient robust synthesis method is developed for sheet and film processes.

Another approach to exploiting the special matrix structure in such problems is to use iterative structure preserving techniques, such as have been explored in the numerical methods and PDE community, to efficiently find centralized optimal controllers. For example, [22] has found  $\mathcal{O}(N^2)$  methods for approximate solutions of Lyapunov and Riccati equations from discretized PDE's, and [23] has decreased this to  $\mathcal{O}(N \log(N))$  for stable PDE's approximated as Hierarchical ( $\mathcal{H}$ ) matrices. However, given an approximately optimal controller, there are no guarantees on the closed loop stability and performance and the controllers have only centralized implementations.

2) *The Structured LMI Approach*: In [4], an  $\mathcal{O}(n^{2\alpha})$  method (where  $\alpha \gtrsim 3$  is the polynomial complexity of the LMI solver) was developed for  $H_\infty$  control of spatially invariant subsystems interconnected on infinite arrays or finite loops. This approach produces a centralized controller that admits a distributed implementation along the same structure as the system, with the necessary controller connectivity being less than or equal to the connectivity of the subsystems.

Recently, this technique has been extended to spatially invariant reversible systems with certain types of boundary conditions in similar complexity [24], to finitely many heterogeneous subsystems in an array with boundary conditions [25] in  $\mathcal{O}(n^{2\alpha}N^\alpha)$ , and to arbitrary interconnection topologies in [26]. There has also been a conservative extension of the results of [4] to heterogeneous systems through robust synthesis and by treating the heterogeneity as norm bounded uncertainty [27].

3) *The Spatial Fourier Transform Approach*: In [28], it is shown that for infinite arrays of spatially invariant interconnected systems, a spatial Fourier transform may be used for solving optimal control problems, and that the optimal controller may then be well approximated using spatial truncation, allowing the designer to pick the connectivity of the controller. These results have recently been extended in [29] to show that the optimal controllers for infinite dimensional exponentially

Manuscript received May 14, 2008; revised July 31, 2008, August 27, 2008, and September 16, 2008. First published May 27, 2009; current version published June 10, 2009. This work was supported by the Dutch Technology Foundation STW. Recommended by Associate Editor R. D. Braatz.

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

Digital Object Identifier 10.1109/TAC.2009.2019802

spatial decaying heterogeneous systems are also exponentially spatial decaying, but there are not yet results on efficiently calculating such controllers.

For a more thorough overview of distributed and decentralized control research, see [30], [31], and the introduction to [4].

In this paper we address the analysis and control of spatially varying systems on finite connected one-dimensional strings, such as those discussed in [25]; the purpose being to develop a method for  $\mathcal{O}(n^3N)$  computational complexity  $H_\infty$  distributed controller synthesis. We approach the problem from a structured matrix algorithmic point of view; by showing the Sequentially Semi-Separable (SSS) structure of the string connected system in Section III, we reveal that efficient ( $\mathcal{O}(n^3N)$ ) matrix operations are possible on such systems. The key to our approach is that controllers with the same SSS matrix structure admit direct distributed (localized) implementations, allowing the use of structure preserving centralized control design techniques to create distributed controllers. To this end, in Section IV we show that using structure preserving iterative methods on SSS matrices, we can efficiently check matrix stability, calculate solutions to Lyapunov and Riccati equations, and synthesize  $H_\infty$  sub-optimal controllers.

The *caveat emptor* is that for practical use, the extreme computational efficiency will come at the cost of successive matrix approximations during the iterative methods, to be discussed in Section V. However, the convergence to the solutions may be checked along the way, and the stability and performance of the synthesized controller may be verified *a posteriori* with the help of an LMI relaxation, in  $\mathcal{O}(n^3N)$  in a manner which is robust to these approximations. We also show methods for efficient controller communication order reduction (Section V-C) including spatial truncation.

To allay fears brought on by such an iterative approximation approach, in Section VI we will conduct an example of distributed control of a discretized wave equation and compare the results to the standard Q-Z based  $H_\infty$  methods used by MATLAB, and to the LMI based distributed control methods described in [26].

We note that no difficult mathematical abstractions or technical results are used in this paper; a few recent advances in numerical linear algebra are incorporated, but the concepts from linear systems and control theory are all standard. Some of the integral results of this paper would be very space consuming to prove, but are trivial to verify by the reader, which should only involve substitution and some paper.

## II. NOTATION/PRELIMINARIES

The sets of real, complex, and natural numbers will be represented by  $\mathbb{R}$ ,  $\mathbb{C}$ , and  $\mathbb{N}$  respectively, and the real and imaginary parts of a complex number will be denoted by  $\Re(\cdot)$  and  $\Im(\cdot)$ . Given some vector  $z$ , its size will be indicated by  $n_z$ : for example,  $z \in \mathbb{R}^{n_z}$ .  $\text{diag}(A, B, \dots, Z)$  will be an appropriately sized matrix with elements  $A \dots Z$ , possibly non-square matrices, running down the diagonal and 0's elsewhere.  $X \succ 0$  will denote the usual hermitian positive definiteness:  $X = X^H$  and  $\lambda(X) \in \mathbb{R}_+$ , and  $\rho(X)$  will represent the spectral radius:  $\rho(X) = \max(|\lambda(X)|)$ .  $\text{Tr}(\cdot)$  will indicate the matrix trace and  $\|A\|_F^2 = \sum_i \sum_j |A_{ij}|^2$  and  $\|A\|_2 = \max_x \|Ax\|_2/\|x\|_2$

will indicate the Frobenius norm and induced 2-norm, or spectral norm, respectively.  $\text{cond}(X)$  will indicate the condition number.

We will use  $\mathcal{F}_l(\cdot)$  and  $\mathcal{F}_u(\cdot)$  to represent lower and upper linear fractional transformations, respectively (for more information, see [32]). The time and Laplace variables will be indicated by  $t$  and  $\sigma$  respectively, while  $s$  will be a spatial index.

For transfer functions,  $RH_\infty$  will indicate the set of proper, real, rational, stable transfer functions, and for some  $G(\sigma) \in RH_\infty$ ,  $\|G(\sigma)\|_\infty$  will indicate the operator induced norm:  $\|G(\sigma)\|_\infty = \sup_{u \in \mathcal{L}_2} (\|Gu\|_2/\|u\|_2) = \sup_{\omega \in \mathbb{R}} \|G(\omega\sqrt{-1})\|_2$ , where  $\mathcal{L}_2$  is the set of Lebesgue integrable functions with norm  $\|u\|_2^2 = \int_{-\infty}^{\infty} u(t)^T u(t) dt$ .

$\hat{\cdot}$ ,  $\tilde{\cdot}$ ,  $\sim$ , and  $\wedge$  will be used throughout on a context dependent basis, usually to represent a transformation or substitution, and will always be locally defined. The super script '\*' will be used to simplify notation, where in each term it will be held constant as either  $m$  or  $p$ .

To represent computational complexity, we will use 'big O' notation,

*Definition 1:* [33] A positive function is  $f(N) \in \mathcal{O}(N^\alpha)$  if there exist finite positive constants,  $\infty > c, \kappa > 0$  such that  $f(N) < cN^\alpha, \forall N > \kappa$ .

Informally, we will say that a procedure 'is'  $\mathcal{O}(N^\alpha)$  if it can be computed in  $f(N) \in \mathcal{O}(N^\alpha)$  flops. For example, dense matrix-matrix multiplications in  $\mathbb{R}^{N \times N}$  are  $\mathcal{O}(N^3)$  while some very special structured matrix operations are of 'linear computational complexity',  $\mathcal{O}(N)$ .

Given no other choice, occasionally we will use bisection on a line to find an upper bound on some optimal value. For a function on  $x \in \mathbb{R}_+$  with a single unknown discontinuity:  $f(x) = \begin{cases} f(x) = 1 & x \geq \theta \\ f(x) = 0 & x < \theta \end{cases}$  and a test of whether or not  $f(x) = 0$ , bisection can be used to find arbitrarily narrow upper and lower bounds  $\theta_u, \theta_l$  on the location of the unknown discontinuity in an efficient manner:

*Lemma 1:* Given some desired tolerance of solution,  $\theta_{tol} > 0$ , bisection will require at most  $k = \mathcal{O}(\log_2(\theta/\theta_{tol}))$  iterations to find upper and lower bounds  $\theta_u > \theta > \theta_l$  such that  $\theta_u - \theta_l \leq \theta_{tol}$ .

*Proof:* See Appendix A. ■

Other notation will be introduced along the way.

## III. SUBSYSTEM MODEL/INTERCONNECTION STRUCTURE

The subsystem models considered will most generally consist of state space realizations of the sort:

$$\Sigma_s : \begin{bmatrix} \dot{x}_s \\ v_{s-1}^p \\ v_{s+1}^m \end{bmatrix} = \begin{bmatrix} A_s & B_s^p & B_s^m \\ C_s^p & W_s^p & Z_s^m \\ C_s^m & Z_s^p & W_s^m \end{bmatrix} \begin{bmatrix} x_s \\ v_s^p \\ v_s^m \end{bmatrix} \quad (2)$$

where  $v_s^m$  and  $v_s^p$  are interconnections to other subsystems (see Fig. 1), and the exogenous inputs and outputs are left out for now. The  $W_s$  terms represent information feedthrough between subsystems  $\Sigma_{s+1}$  and  $\Sigma_{s-1}$ . This type of subsystem has appeared in [4], [25] and associated papers, and is also similar to subsystems considered earlier in [7] and [12]. As in these works, we will assume that the subsystem interconnections  $v^m$  and  $v^p$

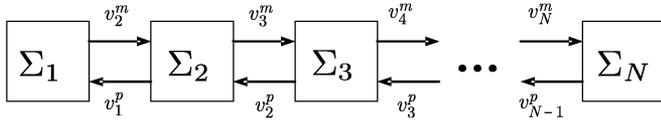


Fig. 1. Subsystem string interconnection.

are ideal, without any delay. A consideration of the effects of delays in these interconnections on system well-posedness, stability, and performance is important for the practical application of these results on real systems (see e.g. [26], [34] for a discussion within the LMI framework) but are out of our current scope.

We will generally allow each subsystem  $\Sigma_s$  to be arbitrarily different from every other subsystem, even having different state, input, and output dimensions, as long as the interconnections are of correct size. An example of such a subsystem model will be shown in Section VI, and others are available in the literature, such as multiple vehicle systems [27], flight formations [9], offshore bases [35], and discretizations of various PDE's [3], [4], [36] etc.

Due to the non-zero  $Z_s^p$  and  $Z_s^m$  terms, the interconnection between subsystems (2) might not be well-posed (see [4] for a discussion). Fortunately, there exist sufficient conditions for well-posedness of  $N$  interconnected subsystems that can be verified in  $\mathcal{O}(N)$ , for example:

*Lemma 1:* The interconnection of  $N$  subsystems  $\Sigma_s$  for  $s \in \{1, 2, \dots, N\}$  is well-posed and may be converted to *spatially strictly-proper* form:

$$\hat{\Sigma}_s : \begin{bmatrix} \dot{x}_s \\ \hat{v}_{s-1}^p \\ \hat{v}_{s+1}^m \end{bmatrix} = \begin{bmatrix} \hat{A}_s & \hat{B}_s^p & \hat{B}_s^m \\ \hat{C}_s^p & \hat{W}_s^p & 0 \\ \hat{C}_s^m & 0 & \hat{W}_s^m \end{bmatrix} \begin{bmatrix} x_s \\ \hat{v}_s^p \\ \hat{v}_s^m \end{bmatrix} \quad (3)$$

if, for the iteration:

$$\tilde{Q}_{s+1} = Z_s^p + W_s^m (I - \tilde{Q}_s Z_s^m)^{-1} \tilde{Q}_s W_s^p \quad (4)$$

with initial condition,  $\tilde{Q}_1 = 0$ , the inverse of  $(I - \tilde{Q}_s Z_s^m)$  exists  $\forall s \in \{1, 2, \dots, N-1\}$ .

*Proof:* The constructive proof is in Appendix B. ■

For structural reasons to be revealed shortly, spatially strictly proper subsystems of the form (3) (without the  $\hat{\cdot}$ 's) will be considered throughout the rest of the paper, assuming that the system is inherently in this form (as in the case of discretizations of PDE's), or has been converted to this form using a method like that in Appendix B.

If  $N$  of these subsystems (3) are connected together in a string (see Fig. 1) with zero boundary inputs ( $v_1^m = 0, v_N^p = 0$ ) and

the interconnection variables are resolved, we obtain the interconnected system:  $\bar{\Sigma} : \bar{\dot{x}} = \bar{A}\bar{x}$  where the  $\overline{\cdot}$  indicates a 'lifted' variable; for vectors:  $\bar{x} = [x_1^T \ x_2^T \ \dots \ x_N^T]^T$ , and the interconnected system matrix  $\bar{A}$  has a very special structure, called 'Sequentially Semi Separable' (SSS) [37]. For example, for  $N = 5$ , we have (1), shown at the bottom of the page. As such matrices will be used often, we will create a special notation for them:

$$\bar{A} = \mathcal{SSS}(B_s^m, W_s^m, C_s^m, A_s, B_s^p, W_s^p, C_s^p) \quad (5)$$

where the arguments of  $\mathcal{SSS}()$  are called the 'generator' matrices of  $\bar{A}$ . This type of data-sparse structured matrix has recently been studied with respect to LTV systems theory and inversion [38], scattering theory [37], and for their own sake [39], [40]. The facts in which we are interested are that SSS matrices can be stored using only  $\mathcal{O}(N)$  memory, there exist algorithms of only  $\mathcal{O}(N)$  computational complexity for SSS matrix-matrix addition and multiplication, inversion, LU, and QR factorization, and further, that the class of SSS matrices is closed under these operations, that is, they are structure preserving. These properties (many of which are similar to those possessed by  $\mathcal{H}$  matrices) are especially important, since they allow the effective use of iterative algorithms incorporating inverses, in contrast to other classes of data sparse matrices (such as banded), which are not closed under inversion.

Since we are interested in the control of interconnected systems, we will now consider the augmented subsystem model,  $\Sigma_s$ :

$$\begin{bmatrix} \dot{x}_s \\ v_{s-1}^p \\ v_{s+1}^m \\ z_s \\ y_s \end{bmatrix} = \begin{bmatrix} A_s & B_s^p & B_s^m & B_s^1 & B_s^2 \\ C_s^p & W_s^p & 0 & L_s^p & V_s^p \\ C_s^m & 0 & W_s^m & L_s^m & V_s^m \\ C_s^1 & J_s^p & J_s^m & D_s^{11} & D_s^{12} \\ C_s^2 & H_s^p & H_s^m & D_s^{21} & D_s^{22} \end{bmatrix} \begin{bmatrix} x_s \\ v_s^p \\ v_s^m \\ w_s \\ u_s \end{bmatrix} \quad (6)$$

which is the same as the subsystem in (3), but augmented to include a performance channel and disturbance input ( $z_s$  and  $w_s$ ), and a measured output and controlled input ( $y_s$  and  $u_s$ ). If we again resolve the interconnection variables, we obtain an interconnected system with SSS state space matrices:

$$\bar{\Sigma} : \begin{bmatrix} \bar{\dot{x}} \\ \bar{z} \\ \bar{y} \end{bmatrix} = \begin{bmatrix} \bar{A} & \bar{B}_1 & \bar{B}_2 \\ \bar{C}_1 & \bar{D}_{11} & \bar{D}_{12} \\ \bar{C}_2 & \bar{D}_{21} & \bar{D}_{22} \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{w} \\ \bar{u} \end{bmatrix} \quad (7)$$

where the SSS notation for each matrix is listed in Appendix D, and the lifted vectors have the same structure as before.

$$\underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix}}_{\bar{\dot{x}}} = \underbrace{\begin{bmatrix} A_1 & B_1^p C_1^p & B_1^p W_2^p C_3^p & B_1^p W_2^p W_3^p C_4^p & B_1^p W_2^p W_3^p W_4^p C_5^p \\ B_2^m C_1^m & A_2 & B_2^p C_3^p & B_2^p W_3^p C_4^p & B_2^p W_3^p W_4^p C_5^p \\ B_3^m W_2^m C_1^m & B_3^m C_2^m & A_3 & B_3^p C_4^p & B_3^p W_4^p C_5^p \\ B_4^m W_3^m W_2^m C_1^m & B_4^m W_3^m C_2^m & B_4^m C_3^m & A_4 & B_4^p C_5^p \\ B_5^m W_4^m W_3^m W_2^m C_1^m & B_5^m W_4^m W_3^m C_2^m & B_5^m W_4^m C_3^m & B_5^m C_4^m & A_5 \end{bmatrix}}_{\bar{A}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}}_{\bar{x}} \quad (1)$$

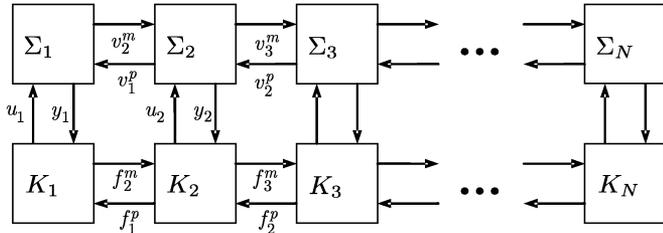


Fig. 2. Distributed controller implementation.

This of course leads to one of the fundamental problems in linear control research, extended to SSS distributed systems:

*Problem 1 ( $H_\infty$  Synthesis):* Given  $\bar{\Sigma}$ , find a stabilizing controller  $\bar{K}$  with SSS structure to solve the optimization problem:

$$\inf_{\bar{K}} \|\mathcal{F}_l(\bar{\Sigma}, \bar{K})\|_\infty = \gamma_{opt}$$

Because a distributed implementation of the controller is desired, not a centralized one, the controller  $\bar{K}$  is constrained to have a realization of SSS matrices, as we will now explain.

Suppose that some method has been used to design a controller  $\bar{K} : \begin{bmatrix} \bar{\xi} \\ \bar{u} \end{bmatrix} = \begin{bmatrix} \bar{A}_K & \bar{B}_K \\ \bar{C}_K & \bar{D}_K \end{bmatrix} \begin{bmatrix} \bar{\xi} \\ \bar{y} \end{bmatrix}$ , for the distributed system (7), where the SSS generators of each matrix are:

$$\begin{aligned} \bar{A}_K &= \text{SSS}(G_s^m, M_s^m, N_s^m, F_s, G_s^p, M_s^p, N_s^p) \\ \bar{B}_K &= \text{SSS}(K_s^m, R_s^m, Q_s^m, G_s, K_s^p, R_s^p, Q_s^p) \\ \bar{C}_K &= \text{SSS}(S_s^m, T_s^m, U_s^m, N_s, S_s^p, T_s^p, U_s^p) \\ \bar{D}_K &= \text{SSS}(X_s^m, Z_s^m, P_s^m, Y_s, X_s^p, Z_s^p, P_s^p) \end{aligned}$$

then it can be verified that such a controller can be directly distributed into subcontrollers  $K_s$ :

$$\begin{bmatrix} \dot{\xi}_s \\ f_{s-1}^p \\ f_{s+1}^m \\ u_s \end{bmatrix} = \begin{bmatrix} F_s & \hat{G}_s^p & \hat{G}_s^m & G_s \\ \hat{N}_s^p & \hat{R}_s^p & 0 & \hat{Q}_s^p \\ \hat{N}_s^m & 0 & \hat{R}_s^m & \hat{Q}_s^m \\ N_s & \hat{S}_s^p & \hat{S}_s^m & Y_s \end{bmatrix} \begin{bmatrix} \xi_s \\ f_s^p \\ f_s^m \\ y_s \end{bmatrix} \quad (8)$$

where

$$\begin{aligned} \hat{G}_s^* &= [G_s^* \quad K_s^* \quad 0 \quad 0], \quad \hat{S}_s^* = [0 \quad 0 \quad S_s^* \quad X_s^*] \\ \hat{N}_s^* &= [(N_s^*)^T \quad 0 \quad (U_s^*)^T \quad 0]^T \\ \hat{Q}_s^* &= [0 \quad (Q_s^*)^T \quad 0 \quad (P_s^*)^T]^T \\ \hat{R}_s^* &= \text{diag}(M_s^*, R_s^*, T_s^*, Z_s^*) \end{aligned}$$

where  $*$  is held constant as either  $m$  or  $p$  in each term. This is obviously the same structure as the subsystems, connected as in Fig. 2, where the  $f_s^*$  channels represent the communications between each subcontroller. This illustrates a key advantage of SSS over  $\mathcal{H}$  matrix or frequency domain controller design methods for distributed systems: SSS structured controllers admit a simple distributed controller implementation, similar in structure to those sought in [4] and [25], without any additional computation.

In the next section we will illustrate how the characteristics of SSS matrices can be exploited to allow for structure preserving  $\mathcal{O}(N)$  algorithms for some very useful control prob-

lems, building up to sub-optimal  $H_\infty$  synthesis of controllers with SSS structure, and thus a distributed implementation.

#### IV. COMPUTATIONAL METHODS

In this section we will discuss iterative methods which may be conducted very efficiently for SSS matrices. The key to the method is that there are ‘fast’ converging iterative algorithms for solving these problems, and that each iteration may be computed using SSS arithmetic algorithms in only  $\mathcal{O}(N)$ . We will begin with a discussion of some of the properties of SSS matrices under arithmetic operations, which are important to understanding our techniques.

##### A. SSS Orders

For an SSS matrix:  $\bar{Q} = \text{SSS}(C_t^c, A_t^c, B_t^c, D_t, C_t^a, A_t^a, B_t^a)$   $t \in \{1, 2, \dots, N\}$ , many matrix-matrix operations have linear computational complexity in  $N$ , but cubic in the sizes of the generator matrices. For example, if  $C_t^c, A_t^c, B_t^c, D_t, C_t^a, A_t^a, B_t^a \in \mathbb{R}^{n \times n}, \forall t \in \{1, 2, \dots, N\}$ , then computing  $\bar{Q}^2$  will take  $40n^3N$  flops or less [37].

However, under the original LTV interpretation of SSS matrices ([38]), in which an SSS matrix,  $\bar{Q}$  was used to represent the truncated input-output map:  $\bar{y} = \bar{Q}\bar{u}$  of a mixed causal Linear Time Varying (LTV) system:

$$\begin{aligned} \begin{bmatrix} x_{t+1}^c \\ x_{t-1}^a \end{bmatrix} &= \begin{bmatrix} A_t^c & 0 \\ 0 & A_t^a \end{bmatrix} \begin{bmatrix} x_t^c \\ x_t^a \end{bmatrix} + \begin{bmatrix} B_t^c \\ B_t^a \end{bmatrix} u_t \\ y_t &= [C_t^c \quad C_t^a] \begin{bmatrix} x_t^c \\ x_t^a \end{bmatrix} + D_t u_t \end{aligned}$$

where  $\bar{y}$  and  $\bar{u}$  are lifted over  $N$  time steps, SSS matrix addition has the interpretation of adding two mixed causal LTV systems in parallel, and SSS matrix multiplication has the interpretation of connecting them in series. It should thus be apparent that the sum or product of two SSS matrices will also be SSS, but of a maximum ‘order’ of the sum of the orders of the original systems.

To be explicit, we define:

*Definition 2:* The maximum upper and lower order of an SSS matrix are the largest sizes of its upper and lower multiplier terms ( $A_t^c$  and  $A_t^a$  in  $\bar{Q}$ ), respectively. The class of SSS matrices of maximum lower and upper orders  $w_l$  and  $w_u$  with  $N$  diagonal terms is denoted as  $\text{SSS}^{w_l, w_u, N}$ .

As discussed above, the growing order of SSS matrices under certain operations can be related as:

*Lemma 3:* For conformably partitioned matrices  $\bar{A} \in \text{SSS}^{a_l, a_u, N}$  and  $\bar{B} \in \text{SSS}^{b_l, b_u, N}$ , then  $\bar{A} + \bar{B} = \bar{C} \in \text{SSS}^{c_l, c_u, N}$  where  $c_l \leq a_l + b_l$ ,  $c_u \leq a_u + b_u$ , and  $\bar{A}\bar{B} = \bar{D} \in \text{SSS}^{d_l, d_u, N}$  where  $d_l \leq a_l + b_l$ ,  $d_u \leq a_u + b_u$ .

*Proof:* This can be seen from the addition and multiplication algorithms [37], [39]. ■

This is important since we intend to use iterative algorithms for controller synthesis, and evidently each iteration will cost more than the previous. In the following subsections we will derive a set of fairly weak assumptions, independent of  $N$ , that can be made on the class of interconnected systems considered to guarantee an upper bound on the number of iterations,  $k$ , necessary for convergence of each algorithm, thus yielding  $\mathcal{O}(N)$  computation, although with a very large multiplicative constant

in front. To remedy this, in Section V we will show that for practical purposes, low order approximations may be used to considerably cut down on the order growth.

### B. Matrix Sign Function

The matrix sign function [41], [42], defined for a square matrix  $X$  with no eigenvalues on the imaginary axis, will be important in our computations.

**Definition 3:** [42] Given a matrix  $X$  with Jordan decomposition  $X = PJP^{-1}$  where  $J = \begin{bmatrix} L & 0 \\ 0 & R \end{bmatrix}$  and  $\lambda(L) \in \mathbb{C}_-$ ,  $\lambda(R) \in \mathbb{C}_+$ , the matrix sign of  $X$  is defined as  $sign(X) = P \begin{bmatrix} -I_L & 0 \\ 0 & I_R \end{bmatrix} P^{-1}$  where  $I_L$  and  $I_R$  are the same size as  $L$  and  $R$ , respectively.

The matrix sign is typically calculated using Algorithm 1,

---

#### Algorithm 1: Sign Iteration [41]

---

$$\begin{aligned} Z_0 &= X \\ \text{for } k &= 0, 1, 2, \dots \\ Z_{k+1} &= \frac{1}{2} (Z_k + Z_k^{-1}) \\ \text{end} \\ sign(X) &= \lim_{k \rightarrow \infty} Z_k \end{aligned}$$


---

which has been proven to be equivalent to a Newton's method, and correspondingly converges quadratically near the solution, but may start out slower. However, from the condition and damping of the spectrum of  $X$ , the number of iterations,  $k$ , to convergence can be bounded:

**Lemma 4:** For a matrix  $X$  with no purely imaginary eigenvalues, the number of sign iterations  $k$  to reach  $\|Z_k - sign(X)\|_2 \leq \epsilon$  for  $X = PJP^{-1}$  will be  $\mathcal{O}(\log_2(\eta)^2 + \log_2(\log_2(\epsilon^{-1} + cond(P))))$ , where  $\eta = \max_i \{1 + |\Re(\lambda_i)| + |\Im(\lambda_i)|^{-1} + (|\Im(\lambda_i)|/|\Re(\lambda_i)|)\}$ .

*Proof:* [23], Lemma 3.5. ■

So, given some  $\bar{X}$  of size  $N$  with SSS structure, we may compute iterations, each of only  $\mathcal{O}(N)$  (since SSS addition, inversion, and scalar multiplication are  $\mathcal{O}(N)$ ), to compute the matrix sign of  $\bar{X}$ . However, to claim  $\mathcal{O}(N)$  sign computation, we must also upper bound  $k$  with a constant, independently of  $N$ ,  $\forall N \in \mathbb{N}$ , using the following assumptions:

- A1:  $\rho(\bar{X}) < \beta_1 < \infty$
- A2:  $\min_i |\Re(\lambda_i(\bar{X}))| > \beta_2 > 0$
- A3:  $cond(P) < \beta_3 < \infty$  for  $\bar{X} = PJP^{-1}$

A1 and A2 imply that  $\eta$  in Lemma 4 will always be finite, but do not seem very restrictive, as they just imply some analytic continuity around the imaginary axis and a finitely bounded spectrum. A3 is the least intuitive, but keep in mind that  $\beta_3$  can be very large without unduly increasing  $k$ , for example  $\log_2(\log_2(10^{100})) < 9$ . From now on the set of matrices that satisfy A1, A2, and A3 will be denoted as  $\mathcal{A}$ . We can thus state:

**Lemma 5:** For the set of SSS matrices  $\bar{X} \in SSS^{\bullet, \bullet, N}$ ,  $\bar{X} \in \mathcal{A} \forall N \in \mathbb{N}$ , an approximation,  $\tilde{S} \in SSS^{\bullet, \bullet, N}$  to  $sign(\bar{X})$  can be calculated to within some prespecified positive tolerance  $\epsilon > \|\tilde{S} - sign(\bar{X})\|_2$  in  $\mathcal{O}(N)$ . □

**Remark 1:** A motivating example for a set of matrices  $\in \mathcal{A}$ ,  $\forall N \in \mathbb{N}$  might be certain Toeplitz matrices with asymptotic spectral properties (see e.g. [43]), associated with spatially invariant systems [44], although we do not restrict ourselves to such cases. See [45] for other examples and counterexamples.

We will use the conservative Frobenius norm to check convergence of the sign iteration, since we can calculate it exactly in  $\mathcal{O}(N)$  for SSS matrices ( $\|\bar{X}\|_F^2 = Tr(\bar{X}^T \bar{X})$ ). For convenience, for the rest of Section IV we will refer to  $\mathcal{O}(N)$  SSS 'solutions' of the matrix sign function, since  $\epsilon$  can be made arbitrarily small, even though in exact arithmetic, Algorithm 1 will never converge exactly to  $sign(X)$  in finite  $k$  unless  $\lambda(X) \in \{-1, 1\}$ . In practice, we say that Algorithm 1 has converged to  $sign(X)$  when  $\|Z_k - Z_{k+1}\|_F < \epsilon$ . The effects of using these and other approximations will be discussed in Section V.

### C. Stability Check by Matrix Sign

From the definition, it is easy to see that:

**Lemma 6:** For some matrix  $X$ ,  $sign(X) = -I$  if and only if  $\Re(\lambda(X)) < 0$ . □

Because of the fast, size independent, convergence of the sign iterations, this can be used as an  $\mathcal{O}(N)$  check on the stability of SSS matrices  $\in \mathcal{A}$ ,  $\forall N \in \mathbb{N}$ . In addition, through the Cayley transform  $X_c = \mathcal{C}(X_d) = (I + X_d)(-I + X_d)^{-1}$ , it is well known that  $\rho(X_d) < 1 \Leftrightarrow Re(\lambda(X_c)) < 0$ , so we can also check the spectral radius, or spectral norm ( $\|X\|_2^2 = \rho(X^T X)$ ) of SSS matrices for which  $\mathcal{C}(\bar{X}) \in \mathcal{A}$  or  $\mathcal{C}(\bar{X}^T \bar{X}) \in \mathcal{A}$  in  $\mathcal{O}(N)$ , which will be useful for  $H_\infty$  synthesis.

### D. Structure Preserving Permutation

Since we will be designing controllers, it is also necessary to check the stability of closed loop systems, which will have matrices with SSS blocks. Using block matrix algebra to compute the matrix sign iteration could fail when using Schur complements to calculate the inverses, so instead the rows and columns of block SSS matrices are permuted to form single SSS matrices, similar to how block Toeplitz matrices may be permuted to obtain matrices with Toeplitz blocks. The following result is for  $2 \times 2$  blocks, but it can be easily extended to other dimensions.

**Lemma 7:** Given matrices  $\bar{W}, \bar{X}, \bar{Y}, \bar{Z}$  with SSS representations:

$$\begin{aligned} \bar{W} &= SSS (B_s^{Wm}, W_s^{Wm}, C_s^{Wm}, A_s^W, B_s^{Wp}, W_s^{Wp}, C_s^{Wp}) \\ \bar{X} &= SSS (B_s^{Xm}, W_s^{Xm}, C_s^{Xm}, A_s^X, B_s^{Xp}, W_s^{Xp}, C_s^{Xp}) \\ \bar{Y} &= SSS (B_s^{Ym}, W_s^{Ym}, C_s^{Ym}, A_s^Y, B_s^{Yp}, W_s^{Yp}, C_s^{Yp}) \\ \bar{Z} &= SSS (B_s^{Zm}, W_s^{Zm}, C_s^{Zm}, A_s^Z, B_s^{Zp}, W_s^{Zp}, C_s^{Zp}) \end{aligned}$$

and lifted vectors  $\bar{e}, \bar{f}, \bar{g}, \bar{h}$ , the relations

$$\begin{bmatrix} \bar{e} \\ \bar{f} \end{bmatrix} = \begin{bmatrix} \bar{W} & \bar{X} \\ \bar{Y} & \bar{Z} \end{bmatrix} \begin{bmatrix} \bar{g} \\ \bar{h} \end{bmatrix}, \quad \begin{bmatrix} e \\ f \end{bmatrix} = \bar{P} \begin{bmatrix} g \\ h \end{bmatrix}$$

are equivalent, to within row and column permutations, where  $\bar{P} = SSS(\hat{B}_s^m, \hat{W}_s^m, \hat{C}_s^m, \hat{A}_s, \hat{B}_s^p, \hat{W}_s^p, \hat{C}_s^p)$  with

$$\begin{aligned}\hat{A}_s &= \begin{bmatrix} A_s^W & A_s^X \\ A_s^Y & A_s^Z \end{bmatrix}, \quad \hat{C}_s^{*} = \begin{bmatrix} \text{diag}(C_s^{W*}, C_s^{X*}) \\ \text{diag}(C_s^{Y*}, C_s^{Z*}) \end{bmatrix} \\ \hat{B}_s^{*} &= \text{diag}([B_s^{W*} \ B_s^{X*}], [B_s^{Y*} \ B_s^{Z*}]) \\ \hat{W}_s^{*} &= \text{diag}(W_s^{W*}, W_s^{X*}, W_s^{Y*}, W_s^{Z*})\end{aligned}$$

and the  $*$ 's are held constant as  $m$  or  $p$  in each term.  $\square$

The transformation from  $\begin{bmatrix} \bar{e} \\ \bar{f} \end{bmatrix} \rightarrow \begin{bmatrix} \bar{e}' \\ \bar{f}' \end{bmatrix}$  could be called a 'shuffle' permutation, since  $\bar{e}$  and  $\bar{f}$  are shuffled like a deck of cards to get  $\begin{bmatrix} \bar{e}' \\ \bar{f}' \end{bmatrix}$ . The reverse operation is also possible:

*Lemma 8:* Given an SSS matrix  $\bar{P}$  with conformably partitioned generators:

$$\bar{P} = SSS\left(\begin{bmatrix} B_s^{1m} \\ B_s^{2m} \end{bmatrix}, W_s^m, [C_s^{1m} \ C_s^{2m}], \begin{bmatrix} A_s^{11} & A_s^{12} \\ A_s^{21} & A_s^{22} \end{bmatrix}, \begin{bmatrix} B_s^{1p} \\ B_s^{2p} \end{bmatrix}, W_s^p, [C_s^{1p} \ C_s^{2p}]\right), \text{ the relations}$$

$$\begin{bmatrix} \bar{e}' \\ \bar{f}' \end{bmatrix} = \bar{P} \begin{bmatrix} \bar{g} \\ \bar{h} \end{bmatrix}, \quad \begin{bmatrix} \bar{e} \\ \bar{f} \end{bmatrix} = \begin{bmatrix} \bar{W} & \bar{X} \\ \bar{Y} & \bar{Z} \end{bmatrix} \begin{bmatrix} \bar{g} \\ \bar{h} \end{bmatrix}$$

are equivalent, to within row and column permutations, where:

$$\begin{aligned}\bar{W} &= SSS(B_s^{1m}, W_s^m, C_s^{1m}, A_s^{11}, B_s^{1p}, W_s^p, C_s^{1p}) \\ \bar{X} &= SSS(B_s^{1m}, W_s^m, C_s^{2m}, A_s^{12}, B_s^{1p}, W_s^p, C_s^{2p}) \\ \bar{Y} &= SSS(B_s^{2m}, W_s^m, C_s^{1m}, A_s^{21}, B_s^{2p}, W_s^p, C_s^{1p}) \\ \bar{Z} &= SSS(B_s^{2m}, W_s^m, C_s^{2m}, A_s^{22}, B_s^{2p}, W_s^p, C_s^{2p})\end{aligned}$$

and everything is assumed conformably partitioned.  $\square$

This result will allow us to permute a closed loop or block SSS matrix, calculate the matrix sign, and then repermute to obtain the sign solution of the original block matrix. This technique will be implicitly used in the sequel.

### E. Lyapunov and Riccati Equations With the Matrix Sign

In [23] it is shown that the matrix sign function can be used to find fast solutions of  $\mathcal{H}$ -matrix Lyapunov and Riccati equations. Lemmas 7 and 8 allow the extension of these methods to SSS matrices.

For a stable, continuous time Lyapunov equation,  $\mathcal{L}(X) = Q + A^T X + X A = 0$ , it has been shown ([41]) that  $\text{sign}\left(\begin{bmatrix} A & 0 \\ Q & -A^T \end{bmatrix}\right) = \begin{bmatrix} -I & 0 \\ 2X & I \end{bmatrix}$ . For  $\bar{A}, \bar{Q} \in SSS^{\bullet, \bullet, N}$ ,  $\bar{A} \in \mathcal{A}$ , this provides an  $\mathcal{O}(N)$  method for solving Lyapunov equations:

*Lemma 9:* For the set of SSS matrices  $\bar{A}, \bar{Q} \in SSS^{\bullet, \bullet, N}$  such that  $\bar{Q} = \bar{Q}^T$ ,  $\lambda(\bar{A}) \in \mathbb{C}_-$  and  $\bar{A} \in \mathcal{A} \forall N \in \mathbb{N}$ , the solution to the Lyapunov equation  $X\bar{A} + \bar{A}^T X = -\bar{Q}$ , will also be  $X \in SSS^{\bullet, \bullet, N}$ . Further, it can be calculated in  $\mathcal{O}(N)$ .

*Proof:* This follows from Lemmas 5, 7, and 8.  $\blacksquare$

The matrix sign function can also be used to solve symmetric algebraic Riccati equations ([41]) (as it was originally intended) of the form

$$\mathcal{R}(X) = XA + A^T X + Q - XRX = 0 \quad (9)$$

where  $Q = Q^T$ ,  $R = R^T$ , by applying sign iterations to the Hamiltonian matrix  $H = \begin{bmatrix} A & -R \\ -Q & -A^T \end{bmatrix}$ , to calculate  $\text{sign}(H) = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}$  and solving the linear system of equations:

$$\begin{bmatrix} S_{12} \\ S_{22} + I \end{bmatrix} X = -\begin{bmatrix} S_{11} + I \\ S_{21} \end{bmatrix} \quad (10)$$

which, following [46], we will call the 'matrix sign equations of  $H$ '. In [46] the following Lemma was proven:

*Lemma 10:* The Riccati equation (9) has a unique symmetric stabilizing solution if and only if  $H$  has no eigenvalues on the imaginary axis and  $\begin{bmatrix} S_{12} \\ S_{22} + I \end{bmatrix}$  is full rank.  $\square$

Thus we have an efficient method for solving Riccati equations:

*Lemma 11:* For the set of SSS matrices  $\bar{A}, \bar{Q}, \bar{R} \in SSS^{\bullet, \bullet, N}$  such that the corresponding Hamiltonian  $H \in \mathcal{A}$ ,  $\forall N \in \mathbb{N}$ , the unique symmetric solution to the Riccati equation (9), if it exists, will also be  $X \in SSS^{\bullet, \bullet, N}$ . Further, it can be calculated in  $\mathcal{O}(N)$ .

*Proof:* This follows from Lemmas 5, 7, and 8. The calculation of the unique solution, under the conditions of Lemma 10, is done by solving (10), which can be computed in  $\mathcal{O}(N)$  using the pseudo-inverse:  $U^+ = (U^T U)^{-1} U^T$ , since  $U = \begin{bmatrix} S_{12} \\ S_{22} + I \end{bmatrix}$  is full rank.  $\blacksquare$

### F. $\mathcal{H}_\infty$ Norm Bounding

To continue our development with Riccati equations, we can restate a result of the bounded real lemma as:

*Lemma 12:* For some  $\gamma > 0$ ,  $\lambda(A) \in \mathbb{C}_-$ , we have  $\|D + C(\sigma I - A)^{-1} B\|_\infty < \gamma$  if and only if  $\|D\|_2 < \gamma$ , and the Riccati equation corresponding to the Hamiltonian matrix:

$$H = \begin{bmatrix} A + BR_\gamma D^T C & BR_\gamma B^T \\ -C^T(I + DR_\gamma D^T)C & -(A + BR_\gamma D^T C)^T \end{bmatrix} \quad (11)$$

(where  $R_\gamma = (\gamma^2 I - D^T D)^{-1}$ ) has a stabilizing solution.

*Proof:* It is well known that  $\|D + C(\sigma I - A)^{-1} B\|_\infty < \gamma$  if and only if  $\|D\|_2 < \gamma$  and  $H$  has no eigenvalues on the imaginary axis, which in turn implies that the Riccati equation has a stabilizing solution [32]. Conversely, if  $\|D\|_2 < \gamma$  then  $BR_\gamma B^T \succeq 0$ , and then the existence of a stabilizing Riccati solution  $X$  implies that  $H$  has no eigenvalues on the imaginary axis [47].  $\blacksquare$

To bound the  $H_\infty$  norm of a transfer function, one need not actually compute the solution to the Riccati equation, but just verify that  $H$  has no eigenvalues on the imaginary axis. As is suggested in [48] we can use this condition for narrowing bounds on the  $H_\infty$  norm of transfer functions by bisection, but we should note that for SSS matrices, using the sign iteration to find the norm exactly will not work in  $\mathcal{O}(N)$ , since as  $\gamma$  approaches  $\|D + C(\sigma I - A)^{-1} B\|_\infty$  from above, the eigenvalues of (11) will approach the imaginary axis, thus breaking assumption A2, but we may compute an upper bound to within some fine tolerance:

*Lemma 13:* For the set of SSS matrices:  $\bar{A}, \bar{B}, \bar{C}, \bar{D} \in \text{SSS}^{\bullet, \bullet, N}$  with  $\lambda(\bar{A}) \in \mathbb{C}_-$  and  $\forall N \in \mathbb{N}$ :

- 1)  $\exists \Theta_m: \|\bar{D} + \bar{C}(\sigma I - \bar{A})^{-1}\bar{B}\|_\infty < \Theta_m < \infty$
- 2)  $\exists \beta > 0$ : the corresponding Hamiltonian (11)  $\bar{H} \in \mathcal{A} \forall \gamma > \beta + \|\bar{D} + \bar{C}(\sigma I - \bar{A})^{-1}\bar{B}\|_\infty$

an upper bound for the  $H_\infty$  norm can be computed to within some  $\gamma_{tol} > \beta$  in  $\mathcal{O}(N)$ .

*Proof:* At each  $\gamma$ , the blocks of the Hamiltonian matrix in (11) will be  $\in \text{SSS}^{\bullet, \bullet, N}$  due to the closure of SSS matrices under addition, multiplication, and inversion. Since each step of the bisection only requires a spectral norm bound and a sign iteration for which we may use the results of Lemma 6 and Algorithm 1, this procedure will be  $k\mathcal{O}(N)$ , where  $k$  is the necessary number of bisections. The proof of Lemma 1 can then be adapted to show that in this case bisection will take  $k = \mathcal{O}(\log_2(\Theta_m/(\gamma_{tol} - \beta)))$  steps to find an upper bound. We have set  $\beta$  and  $\Theta_m$  independently of  $N$ , so the entire procedure will be  $\mathcal{O}(N)$ . ■

We note that  $\beta$ , and thus  $\gamma_{tol}$ , can be set arbitrarily small by manipulating the bounds in assumptions A1, A2, A3.

When working with closed loop SSS block matrices, we can use Lemma 7 to permute the closed loop state vectors to obtain single SSS matrices, then use it again to form the Hamiltonian for sign iterations.

### G. $H_\infty$ Control Synthesis

To simplify our discussion of  $H_\infty$  synthesis for SSS matrices, we will first restate versions of some well known results in centralized  $H_\infty$  synthesis, specialized to suit our situation. For a state space system:

$$\Sigma: \begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} x \\ w \\ u \end{bmatrix} \quad (12)$$

if we define  $\mathcal{K}$  as the set of controllers  $K$  such that  $\mathcal{F}_l(\Sigma, K)$  is internally stable, then specifically, we would like to find some sub-optimal controller  $K \in \mathcal{K}$  such that  $\|\mathcal{F}_l(\Sigma, K)\|_\infty < \gamma$  where  $\gamma_{opt} + \gamma_{tol} \geq \gamma > \gamma_{opt} = \inf_{K \in \mathcal{K}} \|\mathcal{F}_l(\Sigma, K)\|_\infty$ . For some  $\gamma$  and under some general assumptions on  $\Sigma$ , we have the existence result:

*Lemma 14:* [49] For the system  $\Sigma$ , with the assumptions of the ‘general problem’ [49] of  $(A_1, B_2, C_2)$  stabilizable and detectable,  $D_{12}$  full column rank and  $D_{21}$  full row rank (all of the measurements  $y$  are corrupted by noise, and all of the inputs  $u$  are present in the cost term,  $z$ ), and the transfer function matrices of  $y = \Sigma_{(21)}w$  and  $z = \Sigma_{(12)}u$  have no invariant zeros on the imaginary axis, there exists a controller  $K \in \mathcal{K}$  such that  $\|\mathcal{F}_l(\Sigma, K)\|_\infty < \gamma$  if and only if  $\gamma > \max(\|\hat{D}_{11}\|_2, \|\tilde{D}_{11}\|_2)$ , and

- 1) The Hamiltonian matrices:

$$H_X = \begin{bmatrix} \hat{A} + \hat{B}_1 \hat{D}_{11}^{-1} \hat{C}_1 & (\hat{B}_1 \hat{D}_{11}^{-1} \hat{B}_1^T - \hat{B}_2 \hat{B}_2^T) \\ -(\gamma^2 \hat{C}_1^T \hat{D}_{11}^{-1} \hat{C}_1) & -(\hat{A} + \hat{B}_1 \hat{D}_{11}^{-1} \hat{C}_1)^T \end{bmatrix}$$

$$H_Y = \begin{bmatrix} (\hat{A} + \hat{B}_1 \hat{D}_{11}^{-1} \hat{C}_1)^T & \hat{C}_1^T \hat{D}_{11}^{-1} \hat{C}_1 - \hat{C}_2^T \hat{C}_2 \\ -(\gamma^2 \hat{B}_1 \hat{D}_{11}^{-1} \hat{B}_1^T) & -\hat{A} - \hat{B}_1 \hat{D}_{11}^{-1} \hat{C}_1 \end{bmatrix} \quad (13)$$

have no eigenvalues on the imaginary axis.

- 2) The matrix sign equations (10) of  $H_X$  and  $H_Y$  have unique solutions,  $X \succeq 0, Y \succeq 0$
- 3)  $\gamma^{-2} \rho(XY) < 1$ .

where the  $\tilde{\cdot}$  and  $\hat{\cdot}$  matrices in (13) are similar to those in [49] and are again defined in Appendix C.

*Proof:* This follows from the Riccati equations formulation in [49], and the proof of Lemma 10 in [46]. ■

As suggested in [50], for some  $\gamma > \gamma_{opt}$  we will perturb the Hamiltonians (13) by subtracting  $\varepsilon I \succ 0$  from the (2,1) blocks, or equivalently, adding  $\varepsilon I$  to the  $Q$  matrix of the corresponding Riccati equations (9), where  $\varepsilon$  is small enough such that there exist stabilizing solutions:  $X_\varepsilon \succ 0$  and  $Y_\varepsilon \succ 0$ . The scaled inverses of these perturbed solutions will satisfy the  $H_\infty$  Riccati Inequalities ([50]), allowing the computation of explicit controllers using LMI results:

*Lemma 15:* Given some sub-optimal  $\gamma$  and positive definite stabilizing solutions to the perturbed Riccati equations, a stabilizing controller:  $K = \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix}$  such that  $\|\mathcal{F}_l(\Sigma, K)\|_\infty < \gamma$  may be constructed as follows:

Set  $Z = (I - \gamma^{-2} Y_\varepsilon X_\varepsilon)^{-1}$  and define:

$$D'_K = -D_{12}^+ D_{11} D_{21}^+$$

$$B'_K = Z [B_2 D'_K + B_1 D_{21}^+] + Z \Omega_B^T$$

$$C'_K = -[D'_K C_2 + D_{12}^+ C_1] + \Omega_C$$

$$A'_K = Z \left[ \hat{A} + \hat{A} - A + \gamma (B_2 \Omega_C - \Omega_B^T C_2) \right. \\ \left. + \gamma^{-2} Y_\varepsilon (A + B_2 D'_K C_2)^T X_\varepsilon - B_2 D'_K C_2 \right] \\ - \left[ \begin{array}{c} \tilde{B}_1^T Z^T - D_{21}^T \Omega_B Z^T \\ \gamma^{-1} (C_1 + D_{12} D'_K C_2) Y_\varepsilon Z^T \\ \gamma^{-1} (B_1 + B_2 D'_K D_{21})^T X_\varepsilon \\ \hat{C}_1 - D_{12} \Omega_C \end{array} \right]^T \begin{bmatrix} -\gamma I & D_{cl}^T \\ D_{cl} & -\gamma I \end{bmatrix}^{-1} \quad (14)$$

where

$$\Omega_B = \gamma^{-2} (D_{21}^+)^T (D_{cl}^T C_1 + (\gamma^2 - D_{cl}^T D_{11}) \tilde{C}_2) Y_\varepsilon$$

$$\Omega_C = \gamma^{-2} D_{12}^+ (D_{cl} B_1^T + (D_{cl} D_{11}^T - \gamma^2) \hat{B}_2^T) X_\varepsilon$$

$$D_{cl} = D_{11} + D_{12} D'_K D_{21}$$

Then the state space matrices of  $K$  can be computed as:

$$A_K = A'_K - B'_K D_{22} P' C'_K, \quad B_K = B'_K - B'_K D_{22} P' D'_K$$

$$C_K = P' C'_K, \quad D_K = P' D'_K \quad (15)$$

where  $P' = (I + D'_K D_{22})^{-1}$  is assumed to exist.

*Proof:* The controller  $K'$  formulas follow from the LMI methods in [51] for the central controller and under our regularity assumptions, with a slight change of notation, and picking the free parameters such that  $X_\varepsilon$  and  $Y_\varepsilon$  need not be inverted. The conversion from  $K'$  to  $K$  is to reparameterize the controller to the  $D_{22} \neq 0$  case. ■

From these results, we can now state an SSS synthesis result.

*Lemma 16:* For the set of generalized plants (12) with SSS matrix realization  $(\bar{A}, \bar{B}_1, \bar{B}_2, \bar{C}_1, \bar{C}_2, \bar{D}_{11}, \bar{D}_{12}, \bar{D}_{21}, \bar{D}_{22} \in \text{SSS}^{\bullet, \bullet, N})$ , that satisfy the ‘general problem’ conditions of Lemma 14, and in addition,  $\forall N \in \mathbb{N}$ :

- 1)  $\exists \Theta_0: \gamma_{opt} = \min_{K \in \mathcal{K}} \|\mathcal{F}_l(\Sigma, K)\|_\infty < \Theta_0 < \infty$

- 2)  $\exists \tau > 0: \forall \gamma > \tau + \gamma_{opt}$  the corresponding perturbed Hamiltonians  $H_{\bar{X}_\varepsilon}$ ,  $H_{\bar{Y}_\varepsilon}$ , Riccati solutions:  $\bar{X}_\varepsilon$ ,  $\bar{Y}_\varepsilon$ , and  $\mathcal{C}(\gamma^{-2}\bar{X}_\varepsilon\bar{Y}_\varepsilon)$  are all  $\in \mathcal{A}$ .

for  $\gamma_{tol} > \tau$ , a controller  $\bar{K} \in \mathcal{K}$  with closed loop  $H_\infty$  norm less than  $\gamma_{tol} + \gamma_{opt}$ , with state space matrices  $\bar{A}_K, \bar{B}_K, \bar{C}_K, \bar{D}_K \in \text{SSS}^{\bullet, \bullet, N}$  exists, and can be calculated in  $\mathcal{O}(N)$ .

*Proof:* As can be seen in the Appendix, at each  $\gamma$ , the blocks of the Hamiltonian matrices in (13) will be  $\in \text{SSS}^{\bullet, \bullet, N}$  due to the closure of SSS matrices under addition, multiplication, and inversion. The proof is apparent from the construction of the controller matrices, since  $\bar{X}_\varepsilon$  and  $\bar{Y}_\varepsilon$  are SSS as shown in Lemma 11, and  $\bar{A}_K, \bar{B}_K, \bar{C}_K, \bar{D}_K$  are all calculated with arithmetic matrix operations from other SSS matrices. At each  $\gamma$  bisection iteration, the perturbed Riccati equations can be solved efficiently and the positive definite, spectral norm, and spectral radius conditions can be converted to stability problems and efficiently checked, using the results of Lemma 11, Lemma 6 and Algorithm 1. Due to the bounds  $\tau$  and  $\Theta_0$ , bisection will converge in a finite number of iterations, independent of  $N$ ,  $\forall N \in \mathbb{N}$ , as in the proof of Lemma 13. ■

*Remark 2:* We have not specified any method of checking the assumptions of the ‘general problem’ in Lemma 14 because we do not actually have an efficient method for explicitly checking stabilizability, detectability, and the location of transmission zeros. Instead, we suggest performing the SSS  $H_\infty$  synthesis calculations and checking the closed loop stability and performance of the resulting controller (if the Riccati equations have solutions!) using Lemmas 6 and 13 to see if it is valid.

It is now apparent why we do not attempt to find an  $H_\infty$  optimal controller. As  $\gamma$  approaches  $\gamma_{opt}$  from above, it has been shown [49] that the eigenvalues of  $\bar{H}_{\bar{X}_\varepsilon}, \bar{H}_{\bar{Y}_\varepsilon}, \bar{X}_\varepsilon, \bar{Y}_\varepsilon$ , or  $\mathcal{C}(\gamma^{-2}\bar{X}_\varepsilon\bar{Y}_\varepsilon)$  could asymptotically approach the imaginary axis, thus necessarily breaking assumption A2, and we will no longer be able to use the sign iterations efficiently. The other possible failure, the ill-conditioning of (10), will also make SSS calculations difficult. However, we can make  $\tau$  in Lemma 16 small by manipulating the bounds in A1, A2, and A3, and thus calculate a controller in  $\mathcal{O}(N)$  that is arbitrarily close to achieving the optimal solution in Problem 1.

## V. APPROXIMATIONS

In contrast to the final result of Section IV, in practice we usually do not want to compute a centralized almost-optimal controller, for the following reasons.

As mentioned in Section IV-A, for SSS matrix multiplication and addition, the order of the resulting SSS matrix will grow with each operation. Thus we cannot go on adding and multiplying SSS matrices in an efficient manner forever, as the complexity of SSS operations is cubically dependent on the orders. Since we have bounded the number of iterations needed for controller synthesis independently of  $N$ , the overall procedure will be  $\mathcal{O}(N)$ , as stated, but the hidden constant ( $c$  in Definition 1) may be prohibitively large for practical computation.

Furthermore, we see from the controller synthesis (15), (15), that the SSS controller matrices ( $\bar{A}_K, \bar{B}_K, \bar{C}_K, \bar{D}_K$ ) will result from *many* SSS additions and multiplications, resulting in high SSS orders, and that therefore the subcontroller coupling channels,  $f_s^m$  and  $f_s^p$  from (8), being the size of the sums of the lower

and upper SSS orders of  $\bar{A}_K, \bar{B}_K, \bar{C}_K, \bar{D}_K$ , may be very large, placing difficult requirements on the inter-controller communication.

In other words, the procedure described in Section IV for  $H_\infty$  control synthesis for SSS systems *will* produce distributed controllers in  $\mathcal{O}(N)$ , but due to the SSS order growth, the distributed implementation will be impractical and the  $\mathcal{O}(N)$  computational complexity will be slow. The solution to these problems lies in the use of order-reducing SSS approximations, to be discussed in the next subsection.

### A. Optimal Order Reduction

Many infinite dimensional distributed systems can be shown to have exponentially spatially decaying (ESD) [29] operators, a characterization that basically implies that the norm of the couplings between subsystems is bounded by some exponential decay in space.

In [28] it is shown that quadratically optimal controllers for ESD spatially invariant distributed systems are also ESD, and in [29] this is extended to the spatially varying case for the solutions of Lyapunov and Riccati equations of ESD systems. This leads to arguments for spatially truncating the controller for an almost optimal distributed implementation [28].

For very large ( $N \gg 0$ ) finite dimensional systems, we may also take this route: given a finite dimensional Riccati or Lyapunov equation solution with SSS structure,  $\bar{X}$ , we could compute a  $p^{\text{th}}$  spatial truncation approximation of SSS structure by calculating the middle  $2p + 1$  diagonals, and then finding the generator matrices of the truncation  $\bar{X}_p \in \text{SSS}^{p,p,N}$  using the construction algorithm [37] specialized to banded matrices in  $\mathcal{O}(N)$ . Alternatively, within the SSS framework, there is also a *Hankel norm* optimal approximation [37], [38]. That is, given some SSS matrix  $\bar{X} \in \text{SSS}^{q,q,N}$ , we may calculate  $\bar{X}_a$  that achieves:

$$\inf_{\bar{X}_a \in \text{SSS}^{p,p,N}} \|\bar{X}_a - \bar{X}\|_H \quad (16)$$

where  $\|Z\|_H = \max_i \|H_i(Z)\|$  and  $H_i(Z)$  are the Hankel blocks [38] of  $Z$ , and  $p < q$ . The matrix Hankel norm is intimately related to the spectral norm as  $\|Z\|_H \leq \|Z\|_2 \leq \sqrt{N}\|Z\|_H$  for some  $Z \in \mathbb{R}^{N \times N}$ , and there exist  $\mathcal{O}(N)$  algorithms for computing the solution to (16). We find that the  $\|\cdot\|_H$  optimal method often works better than spatial truncation in practice (see Section VI) and its use for order reduction will be implicit for the rest of the discussion.

This also provides an answer to the problem of the SSS order growing after each operation; after every few order-increasing operations, we can do an SSS order reducing approximation, making the iterative schemes very efficient in practice.

### B. Numerical Issues Relating to the Approximations

Of course, iteratively approximating SSS matrices using lower orders will call into question the use of the sign function methods for checking stability and solving Riccati equations. With respect to rounding and truncation errors, the sign function has been found to often work just as well as invariant subspace techniques, but it can be shown that occasionally ill-conditioned sign iterations will cause numerical instability

([52], [53]), and with intermediate low order approximations, such numerical difficulties are even more prominent. Hence we cannot ‘solve’ the Lyapunov and Riccati equations; there will always be non-zero residuals,  $\mathcal{L}(\tilde{X}), \mathcal{R}(\tilde{X})$ , the norms of which are not necessarily a good measure of the backwards error [54], [55], and the suggested criteria are not easily calculable using SSS methods. Furthermore, due to the potential fragility of optimal controllers [56] it is not acceptable to blindly use an approximated controller and assume that it will have the expected closed loop stability and performance; some kind of closed loop test is needed.

For checking matrix stability using Lemma 6, it is usually not feasible to run Algorithm 1 approximation-free, since the orders of  $\bar{Z}_k$  grow as  $2^k$ , but an unsuitable low-order approximation could bump an unstable eigenvalue into the left half plane, or vice-versa, falsifying the result. For stable matrices,  $\Re(\lambda(\bar{X})) < 0$ , the SSS order necessary to represent  $\bar{Z}_k$  exactly will eventually approach 0, since  $\lim_{k \rightarrow \infty} \bar{Z}_k = \text{sign}(\bar{X}) = -I \in \text{SSS}^{0,0,N}$ , but for non-symmetric  $\bar{Z}_k$ , due to the erratic behavior of the spectrum  $\lambda(\bar{Z}_k)$  during the iterations, any approximation,  $\tilde{Z}_k$ , must be extremely accurate:  $\|\bar{Z}_k - \tilde{Z}_k\|_2 < (\beta_2/\beta_1\beta_3)$  to guarantee that no eigenvalues cross the imaginary axis.

However, in the symmetric case,  $\bar{Z}_k = \bar{Z}_k^T$ ,  $\lambda(\bar{Z}_k) \in \mathbb{R}$ , standard perturbation theory [57] shows that

$$\max_{i \in \{1,2,\dots,N\}} \left| \lambda_i(\bar{Z}_k) - \lambda_i(\tilde{Z}_k) \right| \leq \|\bar{Z}_k - \tilde{Z}_k\|_F$$

and for symmetric sign iterations, it can be shown that any eigenvalues initially inside the unit circle  $|\lambda_i(\bar{X})| < 1$  will be transported outside of the unit circle after the first iteration:  $|\lambda_i(\bar{Z}_k)| \geq 1, \forall i \in \{1,2,\dots,N\}, \forall k \in \mathbb{N}$ , allowing the use of aggressive approximations without the danger of eigenvalues crossing the imaginary axis. Fortunately, using Lyapunov stability theory, we can convert non-symmetric stability problems into symmetric problems as follows.

It is well known [58] for some  $A, P \in \mathbb{R}^{N \times N}$ ,  $P = P^T$  with  $A^T P + P A \prec 0$ , that  $\Re(\lambda(A)) < 0$  if and only if  $P \succ 0$ . Hence for SSS matrices, one can use the matrix sign function as in Section IV-E to solve  $\bar{A}^T P + P \bar{A} + I = 0$  for  $P$  using iterative approximations. If the resulting  $\tilde{P} \succ 0$  and  $\bar{A}^T \tilde{P} + \tilde{P} \bar{A} \prec 0$ , which are both symmetric stability problems (and thus robust to approximation errors) then it is guaranteed that  $\Re(\lambda(\bar{A})) < 0$ .

As for the solutions to the Riccati equations, as discussed in Section IV-G for  $H_\infty$  synthesis, due to the analyticity of the Riccati equation [59], there exists some small  $\varepsilon$  such that we may add a perturbation  $\varepsilon I$  and still have a stabilizing solution  $(X_\varepsilon)$ , and this will also be a symmetric solution to the corresponding strict algebraic Riccati inequality for  $H_\infty$  synthesis [50]. Under the same reasoning, given some stabilizing approximate solution to the perturbed Riccati equation via the matrix sign function:  $\tilde{X}_\varepsilon$ , with small non-zero residual  $\mathcal{R}(\tilde{X}_\varepsilon)$ , if  $\varepsilon I \succ \mathcal{R}(\tilde{X}_\varepsilon)$ , which is simply a symmetric stability problem, then  $\tilde{X}_\varepsilon$  will still satisfy the Riccati strict inequality (and similar arguments hold for  $Y_\varepsilon$ ). The controller matrices  $(\bar{A}_k, \bar{B}_k, \bar{C}_k, \bar{D}_k)$  are also computed using approximations, but the closed loop  $H_\infty$  performance may also be verified *a posteriori*, with another Ric-

cati equation in the bounded real lemma (13), on which similar arguments may be applied.

Thus, the use of iterative low-order approximations in checking matrix stability, calculating solutions to Lyapunov and Riccati equations, and synthesizing sub-optimal  $H_\infty$  controllers is acceptable, since the *a posteriori* verifications of closed loop stability and performance may be converted to symmetric matrix stability problems, and thus efficiently and robustly (with respect to approximation errors) checked.

### C. Controller Communication Order Reduction

Even if the order of the SSS matrices of the Riccati solutions and state space matrices  $(\bar{A}_k, \bar{B}_k, \bar{C}_k, \bar{D}_k)$  computed from controller formulas (14) are limited using  $\|\cdot\|_H$  optimal approximation, the distributed controller (8) might still be very inefficiently represented (especially considering the construction of the  $\hat{R}_s$  terms). Thereby, the large size of the communication vectors might surpass the abilities of the controller communication links.

However, through Lemmas 7 and 8, the input-output and state dynamics of the controller (8) may be expressed as a single block SSS matrix:  $\begin{bmatrix} \hat{\xi}_s \\ u_s \end{bmatrix} = \bar{P} \begin{bmatrix} \xi_s \\ y_s \end{bmatrix}$  with

$$\bar{P} = \text{SSS} \left( \begin{bmatrix} \hat{G}_s^m \\ \hat{S}_s^m \end{bmatrix}, \hat{R}_s^m, \begin{bmatrix} \hat{N}_s^m & \hat{Q}_s^m \end{bmatrix}, \begin{bmatrix} F_s & G_s \\ N_s & Y_s \end{bmatrix}, \begin{bmatrix} \hat{G}_s^p \\ \hat{S}_s^p \end{bmatrix}, \hat{R}_s^p, \begin{bmatrix} \hat{N}_s^p & \hat{Q}_s^p \end{bmatrix} \right) \quad (17)$$

And by doing either a spatial truncation, or an SSS Hankel-norm optimal model order reduction on this matrix, the size of the  $\hat{R}$  terms may be reduced without destroying the distributed structure.

This also illustrates a useful feature of the controllers produced within the SSS framework: the communication links  $(f_s^m, f_s^p)$  may be large if desired, corresponding to a more centralized and higher performance controller. In fact, it can be shown that *any* centralized controller can be distributed in the form of (8) using large enough communication channels:  $n_{f_s^m}, n_{f_s^p} \geq \sum_{s=1}^N (n_{x_s} + \max(n_{x_s}, n_{y_s}) + \max(n_{x_s}, n_{u_s}) + \max(n_{u_s}, n_{y_s}))$ . Of course, for implementation, smaller is better, and as we will see in the next section, the freedom to pick the size of the communication links may be very useful in terms of design trade-offs.

However, it should be noted that neither the spatial truncation nor the  $\|\cdot\|_H$  optimal approximation of  $\bar{P}$  is guaranteed to preserve closed loop stability and performance, which should be checked for each order-reduced controller. There *do* exist distributed system model order reduction methods with guaranteed stability and error bounds such as [60], but these are similar in technique and complexity to the LMI synthesis methods described in [25], and thus too computationally expensive for our purposes.

## VI. EXAMPLE

One might suspect that the order-reducing approximations discussed in Section V, used after every few matrix operations during the iterative algorithms of Section IV, would produce controllers with very bad closed loop performance, or even

instability. However, using the discretized heterogeneous undamped wave equation as an example, we will see that the method described can produce low communication order controllers with closed-loop performance nearly that of the centralized optimal controller, computationally much more efficiently.

A. Introduction/Discretization

Consider the 1-dimensional spatially heterogeneous wave equation with position-fixed boundary conditions:

$$\begin{aligned} \frac{\partial^2 x}{\partial t^2} &= k(s) \frac{\partial^2 x}{\partial s^2} + b(s)u + q^1(s)w^1 \\ x(0) = x(L) &= 0 \end{aligned}$$

with some performance and measurement outputs:

$$\begin{aligned} z^1 &= f^1(s)x \\ z^2 &= f^2(s)u \\ y &= c(s)x + q^2(s)w^2 \end{aligned}$$

where  $x(s)$ ,  $u(s)$ ,  $y(s)$ ,  $z^1(s)$ ,  $z^2(s)$ ,  $w^1(s)$ , and  $w^2(s)$  are in Hilbert spaces defined on  $s \in [0, L]$ . Using a finite difference approximation,  $x, y, z^1, z^2, w^1, w^2, u$  can be restricted to be in some finite dimensional Euclidean space  $\mathbb{R}^N$ , where we approximate the spatial derivative to 4th order accuracy [61]:

$$\begin{aligned} \frac{\partial^2 x(s)}{\partial s^2} \approx \frac{1}{12\Delta_s^2} &(-x_{s-2} + 16x_{s-1} - 30x_s \\ &+ 16x_{s+1} - x_{s+2}) + \mathcal{O}(\Delta_s^4) \end{aligned}$$

where  $\Delta_s = L/N$  and with new discrete indices  $s \in \{0, 1, 2, \dots, N\}$ . The approximated system can be written in the form of (6) where

$$\begin{aligned} A_s &= \begin{bmatrix} 0 & 1 \\ -k_s \frac{5}{2\Delta_s^2} & 0 \end{bmatrix}, \quad B_s^m = B_s^p = \begin{bmatrix} 0 & 0 \\ \frac{2k_s}{3\Delta_s^2} & \frac{2k_s}{3\Delta_s^2} \end{bmatrix} \\ W_s^p &= W_s^m = \begin{bmatrix} 0 & -\frac{1}{8} \\ 0 & 0 \end{bmatrix} \\ B_s^1 &= \begin{bmatrix} 0 & 0 \\ q_s^1 & 0 \end{bmatrix}, \quad B_s^2 = \begin{bmatrix} 0 \\ b_s \end{bmatrix}, \quad D_s^{12} = \begin{bmatrix} 0 \\ f_s^2 \end{bmatrix} \\ C_s^p &= C_s^m = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, \quad C_s^1 = \begin{bmatrix} f_s^1 & 0 \\ 0 & 0 \end{bmatrix} \\ C_s^2 &= [c_s \ 0], \quad D_s^{21} = \begin{bmatrix} 0 & q_s^2 \end{bmatrix} \end{aligned}$$

and the other terms are 0. These parameters can then be used to form the lifted system of SSS matrices in (7), allowing the use of the computational tools described in Section IV to perform  $H_\infty$  controller synthesis. The second computational subsection will demonstrate the  $\mathcal{O}(N)$  computational complexity for nearly centrally optimal controllers, and the third will show how the communication-order reduction described in Section V-C can be used to approximate these controllers with very small sized communication channels.

B.  $\mathcal{O}(N)H_\infty$  Synthesis

To demonstrate the application to heterogeneous systems, the system parameters will be chosen to vary randomly in space. This is not meant to represent systems actually encountered in practice, but instead to demonstrate that there is no loss of

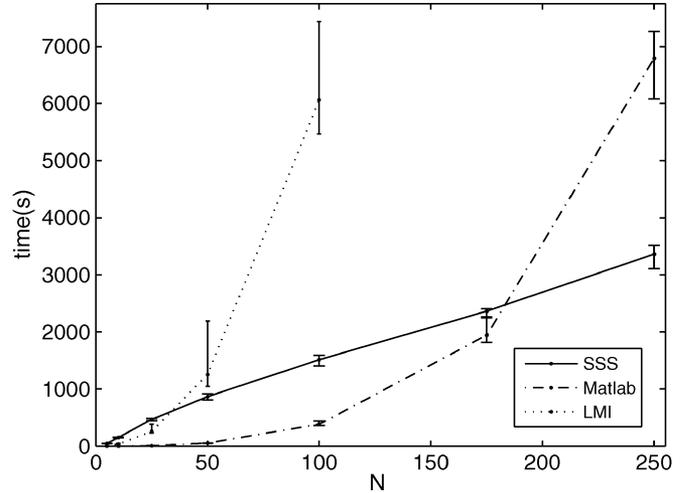


Fig. 3. Average computational time comparison of SSS vs MATLAB  $H_\infty$  vs distributed LMI synthesis routines. Error bars indicate maximum and minimum times.

algorithmic performance or gain in conservatism for the very heterogeneous case. Each coefficient ( $k_s, q_s^1, b_s, f_s^2, f_s^1, c_s, q_s^2$ ) was taken independently to be 1, plus a value picked from a uniform random distribution at each  $s \in \{1, 2, \dots, N\}$ ; for example,  $k_s = 1 + (1/2)\mathcal{U}(-1, 1)$ . To show the consistency of the iterative methods, 25 random systems were generated for each problem size  $N \in \{5, 10, 25, 50, 100, 175, 250\}$ , and  $H_\infty$  sub-optimal controllers produced using MATLAB's `hinfsyn` (using the Riccati solver option) and the SSS based solver with  $\gamma_{tot} = 10^{-2}$ . The  $\gamma_{opt}$  values ranged between  $\approx 2$  and  $\approx 3$ , and both the MATLAB and SSS based closed loop  $H_\infty$  norm values,  $\gamma_M$  and  $\gamma_{SSS}$ , were within the tolerance for each trial of each problem size. In Fig. 3 we see a comparison of the synthesis computation times, where the bars show the maximum and minimum time for each value of  $N$ , and the linear complexity of the SSS approach becomes an advantage after about  $N \approx 200$ .

The LMI based distributed control method as implemented in the Graph Control Toolbox [26] was also used to synthesize controllers for the reduced set of problem sizes  $N \in \{5, 10, 25, 50, 100\}$ , with the same  $\gamma$  tolerance as before. The results in Fig. 3 roughly confirm the polynomial computational complexity estimated in [25] (The number of LMI variables in [25] and [26] are about the same for this type of problem).

For reference to the algorithm discussions, SSS orders of  $w_u = w_l = 18$  were used in all iterative schemes, the sign iterations took about 10 iterations to converge,  $\epsilon$  was taken to be  $10^{-3}$ , and  $\rho(\bar{X}_\epsilon \bar{Y}_\epsilon)$  was the most prominent constraint for the greatest lower bound for  $\gamma_{opt}$ . To produce distributed controllers with nearly centrally-optimal performance, a small  $\gamma_{tot}$  was chosen, but if this constraint were made less severe, lower order  $(w_u, w_l)$  approximations could be used, with a considerable decrease in computation time.

C. Distributed Implementation

For a typical example with  $N = 50$ , Fig. 4 shows the decrease in closed loop  $H_\infty$  norm with the increase in communication order, using a simple suboptimal SSS order reduction algo-

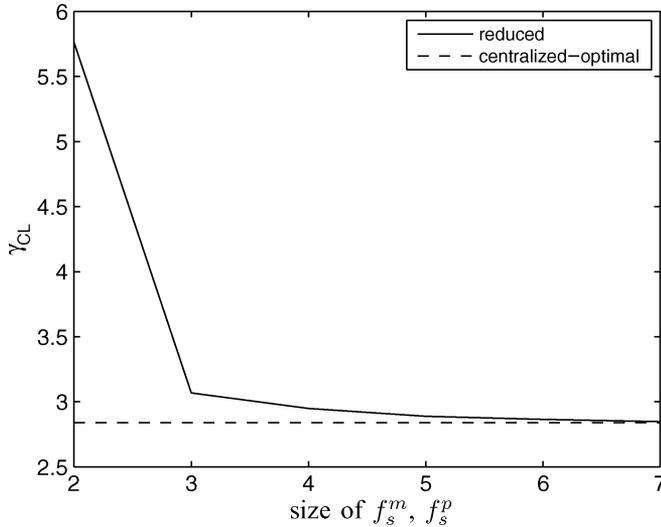


Fig. 4. Closed loop performance of communication-reduced controllers.

algorithm (see [37]) similar in result to the more complicated  $\|\cdot\|_H$  optimal solution discussed in Section V-C. Putting the system matrix (17),  $\bar{P}$  into the appropriate ‘proper’ ([37]) form for reduction took 0.45 seconds (the process is  $\mathcal{O}(N)$ ), and thereafter each order reduction consisted of simply truncating the appropriate matrices in (17) to the reduced dimension, taking negligible time. All reduced controllers with communication links of size at least 2 were stable, and as we see, there is an exponential-like decrease, with high performance controllers even for very small communication links. For comparison, using the truncation-based distribution method, it was necessary to truncate out to 17 spatial indices for a stabilizing controller, and 19 spatial indices to get decent performance ( $\gamma_T = 3.24$ ).

On this same example we used the LMI method as implemented in the Graph Control Toolbox [26], and obtained a closed loop performance of  $\gamma_{LMI} = 9.260$ , with communication size 6 (although it should be possible to reduce this to 2 [26]). This is somewhat surprising: while the original high SSS-order controller  $\bar{K}$  should be non-conservative with respect to the centralized methods, it is unexpected that the communication-reduced controllers would still have superior performance to the dedicated LMI technique, and this is likely not a general result (although it occurred for all examples tested by the authors).

## VII. CONCLUSION

To summarize, we have exhibited the Sequentially Semi-Separable matrix structure of a certain type of interconnected system distributed in a single spatial dimension. This structure was then exploited to find efficient  $\mathcal{O}(n^3N)$  computational complexity algorithms for stability analysis,  $H_\infty$  analysis and synthesis, and communication-order reduction of distributed controllers for spatially heterogeneous systems under some mild assumptions.

For heterogeneous systems, the method was found to be computationally much more efficient than centralized synthesis techniques with a negligible decrease in performance, and also

more efficient than distributed LMI methods [25], [26], although we note that these techniques do not yet take advantage of any structure in the calculations (and indeed are applicable to much more general system structures). Exploiting this structure for distributed computing as discussed in [26], or combining the SSS framework with other LMI solution techniques (e.g. [62]) to approach such problems seems promising, and should be investigated.

Subsystems of the form (6) connected in a ‘loop’ instead of a string also can be lifted to form an interconnected system with SSS state space matrices (although with much more complicated generators), and thus some of the results herein described (such as stability and performance analysis) should also be readily applicable to these systems. However, for such systems, after a controller,  $\bar{K}$ , with SSS matrix realization is computed, it is not yet clear how to extract subcontrollers,  $\bar{K}_s$ , that are also interconnected in a *loop*, instead of a string with very large feedthrough terms. This problem will be the subject of future research.

We also note that the techniques described herein can be applied to large spatially homogenous systems, as a special case of heterogeneous systems, although techniques dedicated to spatial invariance (e.g. [4], [15] for strings and loops) will be more computationally efficient, and may be more easily extended to robust or LPV control [3], [21].

As for extensions, our development has been for continuous time, but there is a discrete time counterpart to solving the Riccati equation with the matrix sign function [63], so many results should be easily extendable. Our methods for stability analysis and  $H_\infty$  control should extend to  $\mathcal{H}$  matrices [23], but it is not clear for the distributed implementation problems or the application to more general interconnected systems. Note that for simplicity we have used only fixed boundary conditions in the example, but by using ‘ghost points’, fairly arbitrary Neumann or Dirichlet conditions can be imposed within the SSS structure.

In this paper we have chosen the  $H_\infty$  problem as a kind of exposition of the usefulness of SSS matrices in distributed problems, but the ability to efficiently solve Lyapunov and Riccati equations for distributed systems is useful for other techniques as well. The use of SSS matrices for  $H_2$  distributed control is discussed in [64] and similar results should hold for e.g. positive real synthesis [65], or LQG control with  $H_\infty$  bound [66]. It is also likely that certain types of robustness analysis may be possible within this framework. Future research will be devoted to extending these results to multiple dimensions.

## APPENDIX A PROOF OF LEMMA 1

- 1) First assume that we have initial upper and lower bounds,  $\theta_u(0), \theta_l(0)$ , and define the ‘span’ as  $s(0) = \theta_u(0) - \theta_l(0)$ . At bisection  $k$ , it is easy to see that  $s(k) = (1/2)s(k-1) = (1/2)^k s(0)$ , so  $s(k) < \theta_{tol}$  for  $k > \log_2(s(0)/\theta_{tol}) \in \mathcal{O}(\log_2(s(0)/\theta_{tol}))$ .
- 2) Now assume that we don’t have initial upper and lower bounds, just  $\theta_{tol}$ . First we check  $f(x = \theta_{tol})$ . If  $f(\theta_{tol}) > 0$  then we have lower and upper bounds  $[0, \theta_{tol}]$  and we

are finished in one step, alternatively if  $f(\theta_{tol}) < 0$  then we set  $\theta_l = \theta_{tol}$  and check  $2\theta_{tol}$ ,  $4\theta_{tol}$ ,  $8\theta_{tol}$  and so on, until we find the smallest  $k \in \mathbb{N}$  such that we have an upper bound  $2^k\theta_{tol} > \theta$  or equivalently  $k > \log_2(\theta/\theta_{tol})$  which will obviously take  $\mathcal{O}(\log_2(\theta/\theta_{tol}))$  steps. We now set  $\theta_u = 2^k\theta_{tol}$  and  $\theta_l = 2^{k-1}\theta_{tol} = \theta_u/2$ , thus  $span = \theta_u - \theta_l = \theta_u/2 = \theta_l$ , and since  $\theta_l < \theta$ , then  $span < \theta$  and it will take  $\log_2(2^{k-1}\theta_{tol}/\theta_{tol}) < \log_2(\theta/\theta_{tol})$  to narrow the bounds to  $\theta_{tol}$ , just as in part (1). Since the doubling procedure to first obtain  $\theta_u$ , and the bisection thereafter are both  $\mathcal{O}(\log_2(\theta/\theta_{tol}))$ , thus so is the whole procedure.

#### APPENDIX B PROOF OF LEMMA 2

We will first show the conversion method, and then discuss the proof. We will first eliminate the  $Z_s^p$  terms through a change of variables, and then eliminate the  $Z_s^m$  terms.

After running the  $\tilde{Q}$  iteration (4), define  $\tilde{Z}_s^m = (I - \tilde{Q}_s Z_s^m)^{-1}$  and  $\tilde{Z}_s^p = (I - Z_s^m \tilde{Q}_s)^{-1}$ , and the new set of subsystems:

$$\hat{\Sigma}_s : \begin{bmatrix} \dot{x}_s \\ v_{s-1}^p \\ \hat{v}_{s+1}^m \end{bmatrix} = \begin{bmatrix} \hat{A}_s & \hat{B}_s^p & \hat{B}_s^m \\ \hat{C}_s^p & \hat{W}_s^p & Z_s^m \\ \hat{C}_s^m & 0 & \hat{W}_s^m \end{bmatrix} \begin{bmatrix} x_s \\ v_s^p \\ \hat{v}_s^m \end{bmatrix} \quad (18)$$

where

$$\begin{aligned} \hat{A}_s &= [A_s + B_s^m \tilde{Z}_s^m \tilde{Q}_s C_s^p], & \hat{Z}_s^m &= \tilde{Z}_s^p Z_s^m \\ \hat{B}_s^p &= [B_s^p + B_s^m \tilde{Z}_s^m \tilde{Q}_s W_s^p], & \hat{B}_s^m &= B_s^m \tilde{Z}_s^m \\ \hat{C}_s^m &= [C_s^m + W_s^m \tilde{Z}_s^m \tilde{Q}_s C_s^p], & \hat{C}_s^p &= \tilde{Z}_s^p C_s^p \\ \hat{W}_s^p &= \tilde{Z}_s^p W_s^p, & \hat{W}_s^m &= W_s^m \tilde{Z}_s^m \end{aligned}$$

which will be equivalent to the interconnection of (2), but is spatially strictly proper in one direction.

Now perform the backwards Stein iteration:

$$\check{Z}_{i-1} = \hat{Z}_{i-1}^m + \hat{W}_{i-1}^p \check{Z}_i \hat{W}_{i-1}^m$$

starting with  $\check{Z}_N = 0$ , and define the new set of subsystems:

$$\hat{\Sigma}_s : \begin{bmatrix} \dot{x}_s \\ \hat{v}_{s-1}^p \\ \hat{v}_{s+1}^m \end{bmatrix} = \begin{bmatrix} \hat{A}_s & \hat{B}_s^p & \hat{B}_s^m \\ \hat{C}_s^p & \hat{W}_s^p & 0 \\ \hat{C}_s^m & 0 & \hat{W}_s^m \end{bmatrix} \begin{bmatrix} x_s \\ \hat{v}_s^p \\ \hat{v}_s^m \end{bmatrix} \quad (19)$$

where

$$\begin{aligned} \hat{A}_s &= \hat{A}_s + \hat{B}_s^p \check{Z}_s \hat{C}_s^m, & \hat{B}_s^m &= \hat{B}_s^m + \hat{B}_s^p \check{Z}_s \hat{W}_s^m \\ \hat{C}_s^p &= \hat{C}_s^p + \hat{W}_s^p \check{Z}_s \hat{C}_s^m, & \hat{B}_s^p &= \hat{B}_s^p, & \hat{W}_s^p &= \hat{W}_s^p \\ \hat{C}_s^m &= \hat{C}_s^m, & \hat{W}_s^m &= \hat{W}_s^m \end{aligned}$$

and these will obviously be spatially strictly proper. The sufficiency thus comes from the equivalence of (19) to (2) and since (19) will always be well-posed.

#### APPENDIX C

We define:

$$\begin{aligned} \hat{B}_2 &= \bar{B}_2 \bar{D}_{12}^+, & \hat{A} &= \bar{A} - \hat{B}_2 \bar{C}_1; & \hat{B}_1 &= \bar{B}_1 - \hat{B}_2 \bar{D}_{11} \\ \hat{C}_2 &= \bar{D}_{21}^+ \bar{C}_2, & \hat{A} &= \bar{A} - \bar{B}_1 \hat{C}_2; & \hat{C}_1 &= \bar{C}_1 - \bar{D}_{11} \hat{C}_2 \\ \hat{C}_1 &= (I - \bar{D}_{12} \bar{D}_{12}^+) \bar{C}_1 & \hat{D}_{11} &= (I - \bar{D}_{12} \bar{D}_{12}^+) \bar{D}_{11} \\ \hat{B}_1 &= \bar{B}_1 (I - \bar{D}_{21}^+ \bar{D}_{21}) & \hat{D}_{11} &= \bar{D}_{11} (I - \bar{D}_{21}^+ \bar{D}_{21}) \\ \hat{D}_\gamma &= (\gamma^2 I - \hat{D}_{11} \hat{D}_{11}^T), & \hat{D}_{\gamma T} &= (\gamma^2 I - \hat{D}_{11}^T \hat{D}_{11}) \\ \check{D}_\gamma &= (\gamma^2 I - \check{D}_{11} \check{D}_{11}^T), & \check{D}_{\gamma T} &= (\gamma^2 I - \check{D}_{11}^T \check{D}_{11}) \end{aligned}$$

#### APPENDIX D

$$\begin{aligned} \bar{B}_1 &= SSS(B^m, W^m, L^m, B^1, B^p, W^p, L^p) \\ \bar{B}_2 &= SSS(B^m, W^m, V^m, B^2, B^p, W^p, V^p) \\ \bar{C}_1 &= SSS(J^m, W^m, C^m, C^1, J^p, W^p, C^p) \\ \bar{D}_{11} &= SSS(J^m, W^m, L^m, D^{11}, J^p, W^p, L^p) \\ \bar{D}_{12} &= SSS(J^m, W^m, V^m, D^{12}, J^p, W^p, V^p) \\ \bar{C}_2 &= SSS(H^m, W^m, C^m, C^2, H^p, W^p, C^p) \\ \bar{D}_{21} &= SSS(H^m, W^m, L^m, D^{21}, H^p, W^p, L^p) \\ \bar{D}_{22} &= SSS(H^m, W^m, V^m, D^{22}, H^p, W^p, V^p) \end{aligned}$$

#### ACKNOWLEDGMENT

The authors would like to thank the reviewers for their helpful comments, Dr. H. Koroglu, Dr. R. Fraanje, P. Massioni, Dr. T. Keviczky, I. Polat, R. Ellenbroek, and J. W. van Wingerden for their fruitful discussions, and especially C. Langbort for his help with the Graph Control Toolbox.

#### REFERENCES

- [1] D. S. Henningson, M. Hogberg, and M. Chevalier, "Optimal feedback control applied to boundary layer flow," *J. Turbulence*, vol. 6, no. 25, 2005.
- [2] T. R. Bewley and S. Liu, "Optimal and robust control and estimation of linear paths to transition," *J. Fluid Mech.*, vol. 365, pp. 205–349, 1998.
- [3] F. Wu and S. E. Yildizoglu, "Distributed parameter-dependent modeling and control of flexible structures," *Trans. ASME*, vol. 127, pp. 230–239, 2005.
- [4] R. D'Andrea and G. E. Dullerud, "Distributed control design for spatially interconnected systems," *IEEE Trans. Automat. Control*, vol. 48, no. 9, pp. 1478–1495, Sep. 2003.
- [5] R. Horowitz and P. Varaiya, "Control design of an automated highway system," *Proc. IEEE*, vol. 88, no. 7, pp. 913–925, Jul. 2000.
- [6] D. Swaroop and J. Hedrick, "String stability of interconnected systems," *IEEE Trans. Automat. Control*, vol. 41, no. 3, pp. 349–357, Mar. 1996.
- [7] D. D. Givone and R. P. Roesser, "Multidimensional linear iterative circuits-general properties," *IEEE Trans. Computers*, vol. 21, no. 10, pp. 1067–1073, Oct. 1972.
- [8] J. P. Lynch and K. H. Law, "Decentralized control techniques for large-scale civil structural systems," in *Proc. IMAC*, 2002, [CD ROM].
- [9] J. Fowler and R. D'Andrea, "A formation flight experiment," *IEEE Control Syst. Mag.*, vol. 23, pp. 35–43, 2003.
- [10] P. Massioni, "Distributed control for identical dynamically coupled systems: A decomposition approach," *IEEE Trans. Automat. Control*, vol. 54, no. 1, pp. 124–135, Jan. 2009.

- [11] S. Jiang, P. G. Voulgaris, L. E. Holloway, and L. A. Thompson, "Distributed control of large segmented telescopes," in *Proc. Amer. Control Conf.*, 2006, pp. 1942–1947.
- [12] R. P. Roesser, "A discrete state-space model for linear image processing," *IEEE Trans. Automat. Control*, vol. AC-20, no. 1, pp. 1–10, Jan. 1975.
- [13] 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.
- [14] M. Cantoni, E. Weyer, Y. Li, S. K. Ooi, I. Mareels, and M. Ryan, "Control of large scale irrigation networks," *Proc. IEEE*, vol. 95, no. 1, pp. 75–91, Jan. 2007.
- [15] R. W. Brockett and J. L. Willems, "Discretized partial differential equations: Examples of control systems defined on modules," *Automatica*, vol. 10, pp. 507–515, 1974.
- [16] M. Hovd, R. D. Braatz, and S. Skogestad, "SVD controllers for  $H_2$ -,  $H_\infty$ - and  $\mu$ -optimal control," *Automatica*, vol. 33, pp. 433–439, 1997.
- [17] J. Anthonis and H. Ramon, "Linear mechanical systems and dyadic transfer function matrices," *Automatica*, vol. 39, pp. 1353–1363, 2003.
- [18] D. L. Laughlin, M. Morari, and R. D. Braatz, "Robust performance of cross-directional basis-weight control in paper machines," *Automatica*, vol. 29, pp. 1395–1410, 1993.
- [19] M. Sundareshan and R. Elbanna, "Qualitative analysis and decentralized controller synthesis for a class of large-scale systems with symmetrically interconnected subsystems," *Automatica*, vol. 27, pp. 383–388, 1991.
- [20] M. Hovd and S. Skogestad, "Control of symmetrically interconnected plants," *Automatica*, vol. 30, pp. 957–973, 1994.
- [21] J. G. VanAntwerp, A. P. Featherstone, and R. D. Braatz, "Robust cross-directional control of large scale sheet and film processes," *J. Process Control*, vol. 11, pp. 149–177, 2001.
- [22] I. Rosen and C. Wang, "A multilevel technique for the approximate solution of operator Lyapunov and algebraic Riccati equations," *SIAM J. Numer. Anal.*, vol. 32, pp. 514–541, 1995.
- [23] L. Grasedyck, W. Hackbusch, and B. Khoromskij, "Solution of large scale algebraic matrix Riccati equations by use of hierarchical matrices," *Computing*, vol. 70, pp. 121–165, 2003.
- [24] C. Langbort and R. D'Andrea, "Distributed control of spatially reversible interconnected systems with boundary conditions," *SIAM J. Control Optim.*, vol. 44, pp. 1–28, 2005.
- [25] G. E. Dullerud and R. D'Andrea, "Distributed control of heterogeneous systems," *IEEE Trans. Automat. Control*, vol. 49, no. 12, pp. 2113–2128, Dec. 2004.
- [26] C. Langbort, R. Chandra, and R. D'Andrea, "Distributed control design for systems interconnected over an arbitrary graph," *IEEE Trans. Automat. Control*, vol. 49, no. 9, pp. 1502–1519, Sep. 2004.
- [27] F. Wu, "Distributed control for interconnected linear parameter-dependent systems," *Proc. IEEE*, vol. 150, no. 5, pp. 518–527, Sep. 2003.
- [28] B. Bamieh, F. Paganini, and M. A. Dahleh, "Distributed control of spatially invariant systems," *IEEE Trans. Automat. Control*, vol. 47, no. 7, pp. 1091–1107, Jul. 2002.
- [29] N. Motee and A. Jadbabaie, "Optimal control of spatially distributed systems," *IEEE Trans. Automat. Control*, to be published.
- [30] A. Robinson, "A survey of optimal control of distributed-parameter systems," *Automatica*, vol. 7, pp. 371–388, 1971.
- [31] N. R. S. Jr., P. Varaiya, M. Athans, and M. G. Safonov, "Survey of decentralized control methods for large scale systems," *IEEE Trans. Automat. Control*, vol. AC-23, no. 2, pp. 108–129, Apr. 1978.
- [32] K. Zhou, K. Glover, and J. C. Doyle, *Robust and Optimal Control*. Englewood Cliffs, NJ: Prentice Hall, 1996.
- [33] D. E. Knuth, "Big omicron and big omega and big theta," *ACM SIGACT News*, vol. 8, pp. 18–24, 1976.
- [34] R. Chandra, C. Langbort, and R. D'Andrea, "Distributed control design for spatially interconnected systems with robustness to small communication delays," *Math. Theory Networks Syst.*, 2004.
- [35] P. Sharma and C. Beck, "Modelling and distributed control of mobile offshore bases," in *Proc. Amer. Control Conf.*, 2004, pp. 5238–5243.
- [36] R. D'Andrea, "Linear matrix inequalities, multidimensional system optimization, and control of spatially distributed systems: An example," in *Proc. Amer. Control Conf.*, 1999, pp. 2713–2717.
- [37] S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, A.-J. van der Veen, and D. White, *Fast Stable Solvers for Sequentially Semi-Separable Linear Systems of Equations*. Lawrence Livermore National Laboratory, Tech. Rep., 2003.
- [38] P. Dewilde and A. J. V. D. Veen, *Time-Varying Systems and Computations*. Norwell, MA: Kluwer Academic Publishers, 1998.
- [39] Y. Eidelman and I. Gohberg, "On a new class of structured matrices," *Integral Equations Operator Theory*, vol. 34, pp. 292–324, 1999.
- [40] Y. Eidelman, I. Gohberg, and V. Olshevsky, "The QR iteration method for Hermitian quasiseparable matrices of an arbitrary order," *Linear Algebra Appl.*, vol. 404, pp. 305–324, 2005.
- [41] J. Roberts, "Linear model reduction and solution of the algebraic Riccati equation by use of the sign function," *Int. J. Control*, vol. 32, pp. 677–687, 1980.
- [42] C. S. Kenney and A. J. Laub, "The matrix sign function," *IEEE Trans. Automat. Control*, vol. 40, no. 8, pp. 1330–1348, Aug. 1995.
- [43] A. Bottcher and B. Silberman, *Introduction to Large Truncated Toeplitz Matrices*. New York: Springer, 1991.
- [44] B. Bamieh, "The structure of optimal controllers of spatially-invariant distributed parameter systems," in *Proc. IEEE Conf. Decision and Control*, 1997, pp. 1056–1061.
- [45] M. R. Jovanović and B. Bamieh, "On the ill-posedness of certain vehicular platoon control problems," *IEEE Trans. Automat. Control*, vol. 50, no. 1, p. 9, Jan. 2005.
- [46] C. Kenney, A. J. Laub, and E. Jonckheere, "Positive and negative solutions of dual Riccati equations by matrix sign function iteration," *Syst. Control Lett.*, vol. 13, pp. 109–116, 1989.
- [47] C. Scherer, *Theory of Robust Control (Course Notes)*. Delft Center for Systems and Control. Delft, The Netherlands, 2001.
- [48] S. Boyd, V. Balakrishnan, and P. Kabamba, "A bisection method for computing the  $H_\infty$  norm of a transfer matrix and related problems," *Math. Control Signals Syst.*, vol. 2, pp. 207–219, 1989.
- [49] P. Gahinet, "On the game riccati equations arising in  $H_\infty$  control problems," *SIAM J. Control Optim.*, vol. 32, pp. 635–647, 1994.
- [50] P. Gahinet and P. Apkarian, "A linear matrix inequality approach to  $H_\infty$  control," *Int. J. Nonlinear Robust Control*, pp. 421–448, 1994.
- [51] P. Gahinet, "Explicit controller formulas for LMI-based  $H_\infty$  synthesis," *Automatica*, vol. 32, pp. 1007–1014, 1996.
- [52] R. Byers, C. He, and V. Mehrmann, "The matrix sign function method and the computation of invariant subspaces," *SIAM J. Matrix Anal. Appl.*, vol. 18, pp. 615–632, 1997.
- [53] J. D. Gardiner, "A stabilized matrix sign function algorithm for solving algebraic Riccati equations," *SIAM J. Sci. Comput.*, vol. 18, pp. 1393–1411, 1997.
- [54] N. J. Higham, "Perturbation theory and backward error for  $AX - XB = C$ ," *BIT Numer. Math.*, pp. 124–136, 1993.
- [55] C. Kenney, A. J. Laub, and M. Wette, "Error bounds for Newton refinement of solutions to algebraic Riccati equations," *Math. Control, Signals, Syst.*, vol. 3, pp. 211–224, 1990.
- [56] L. Keel and S. Bhattacharyya, "Robust, fragile or optimal," in *Proc. Amer. Control Conf.*, 1997, pp. 1307–1313.
- [57] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Baltimore, MD: JHU Press, 1996.
- [58] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge, U.K.: Cambridge University Press, 1991.
- [59] D. F. Delchamps, "Analytic stabilization and the algebraic Riccati equation," in *Proc. IEEE Conf. Decision Control*, 1983, pp. 1396–1401.
- [60] S. Samar and C. Beck, "Model reduction of heterogeneous distributed systems," in *Proc. IEEE Conf. Decision Control*, 2003, pp. 5271–5276.
- [61] J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*. Belmont, CA: Wadsworth Publ. Co. 1989.
- [62] R. Wallina, C.-Y. Kaob, and A. Hanssona, "A cutting plane method for solving KYP-SDPs," *Automatica*, vol. 44, pp. 418–429, 2008.
- [63] C. S. Kenney, A. J. Laub, and P. M. Papadopoulos, "Matrix-sign algorithms for Riccati equations," *IMA J. Math. Control Inform.*, vol. 9, pp. 331–344, 1992.
- [64] J. K. Rice and M. Verhaegen, "Distributed control: A sequentially semi-separable approach," in *Proc. IEEE Conf. Decision Control*, 2008, pp. 2913–2919.
- [65] W. Sun, P. Khargonekar, and D. Shim, "Solution to the positive real control problem for linear time-invariant systems," *IEEE Trans. Automat. Control*, vol. 39, no. 10, pp. 2034–2046, Oct. 1994.
- [66] D. S. Bernstein and W. M. Haddad, "LQG control with an  $H_\infty$  performance bound: A Riccati equation approach," *IEEE Trans. Automat. Control*, vol. AC-34, no. 3, pp. 293–306, Mar. 1989.



**Justin K. Rice** (S'08) received the B.S. and M.Eng degrees in mechanical and aerospace engineering from Cornell University, Ithaca, NY, in 2005 and 2006, respectively, and is currently pursuing the Ph.D. degree at the Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands.

His interests include distributed and interconnected systems and control theory, particularly in application to renewable energy technologies, such as wind turbines, and in the study of complex biological systems, such as schools of fish, flocks of birds, and octopuses.



**Michel Verhaegen** received an engineering degree in aeronautics from the Delft University of Technology, Delft, The Netherlands, in 1982 and the doctoral 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 Research Fellow of the U.S. National Research Council (NRC), affiliated with the NASA Ames Research Center in California, and 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 was appointed as Full Professor with the Faculty of Applied Physics, University of 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) in Munich and is participating in several European Research Networks. His main research interest is the interdisciplinary domain of numerical 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 industry. Application areas include smart structures, adaptive optics, and vehicle mechatronics.