2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Structure preserving model order reduction of heterogeneous 1-D distributed systems

Justin K. Rice, Michel Verhaegen

Abstract— **We consider the problem of model order reduction for spatially-varying interconnected systems distributed in one spatial dimension. The sequentially semi-separable matrix structure of such systems can be exploited to allow efficient structure preserving model order reduction using the matrix sign function. Iterative algorithms are provided for fast computation, which is demonstrated on an example.**

I. Introduction

The control of spatially distributed interconnected systems has recently been of great interest in practical applications such as turbulent flow relaminarization [1], air traffic control [2], biochemical reactions [3], and wind turbine farms [4] (see also the special issue [5]).

The challenge has been in the computational cost of designing effective controllers and the complexity of implementing them. For 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. Much research has been dedicated to surmounting this computational obstacle. In [6], a special matrix structure($\mathcal{H}$-matrix) has been exploited in iterative methods for finding fast $\mathcal{O}(N\log(N))$ approximate solutions to Lyapunov and Riccati equations for systems governed by discretized PDE's. In [7] an efficient LMI(linear matrix inequality) method for spatially homogenous interconnected structures was developed, which was extended to finitely many heterogeneous subsystems in an array with boundary conditions [8] in $\mathcal{O}(n^{2\alpha}N^{\alpha})$(where $3.5 < \alpha < 5$). In [9] a different matrix structure('Sequentially Semi Separable', (SSS)) was exploited to find efficient ($\mathcal{O}(n^3N)$) nearly optimal distributed controllers for such systems, and in this paper we will use the same structure.

We address the problem of model order reduction of spatially varying systems connected on one-dimensional strings. Significant advances on this issue have been made from the subsystem point of view: for spatially invariant, infinite dimensional systems: [10][11] and the extension to finite dimensional spatially varying systems: [12] (see also [13] for arbitrary interconnection structures). While such methods preserve the underlying distributed structure of the system, in the heterogeneous case, due to the sizes of the LMI's to be solved, the computational complexity still grows as $\mathcal{O}(N^{\sim 3})$.

There has also been considerable work from the large-scale computations community, for example [14],[15] and a survey, [16], where the large matrices are assumed to have some kind of structure that is preserved to some extent under iterative computations, or can be used with a proper orthogonal decomposition. In some cases these results are limited, in that the reduced system in general does not have the same structure as the original, for example, in [17], structure preserving computations are performed on Hierarchical matrices to solve Lyapunov equations in a low rank factored form, and then a rank revealing, and structure destroying, computation (an SVD) is done to perform an approximate balancing.

These methods seem to work well when the large scale system can be well approximated by a low order system, on the scale for which SVD's may be computed efficiently, but in some cases we will want the reduced system to preserve the structure of the underlying system [18], for example, such that it will still admit a distributed implementation (as in [12]), and also to be very high order (although lower than before) for better accuracy. Such a method, in $\mathcal{O}(N)$, will be discussed in this paper for systems with a Sequentially Semi-Separable structure.

We will take a similar line of study as that for control in [19]: in section II we will briefly show how certain types of interconnected subsystems lead to lifted systems with an SSS matrix structure, which is preserved under various arithmetic operations. These operations can be used to efficiently compute an SSS approximation of the matrix sign function of a given SSS matrix, and thereby, for example, solve Lyapunov and Riccati equations and check the stability and norm for SSS matrices (section III). Further, by using certain permutation techniques, we will show how the matrix sign function may be used to compute a structure preserving partial system balancing, which, when combined with the techniques for Lyapunov and Riccati equations in section IV will result in a computationally efficient($\mathcal{O}(N)$) structure preserving model order reduction for 1-dimensional spatially distributed heterogeneous systems with guaranteed stability and computable error bounds. We will demonstrate the $\mathcal{O}(N)$ complexity and structure preserving nature on a numerical example in section V.

978-1-4244-4524-0/09/$25.00 ©2009 AACC

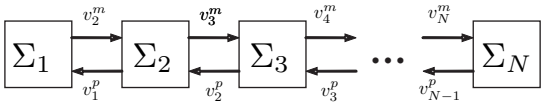

Fig. 1. String interconnection

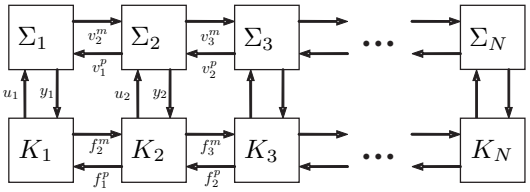

Fig. 2. Controller implementation

II. Subsystem model/Interconnection Structure

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

$$\Sigma_s : \quad \begin{bmatrix} \dot{x}_s \\ v_{s-1}^p \\ v_{s+1}^m \\ y_s \end{bmatrix} = \begin{bmatrix} A_s & B_s^p & B_s^m & B_s \\ C_s^p & W_s^p & 0 & V_s^p \\ C_s^m & 0 & W_s^m & V_s^m \\ C_s & H_s^p & H_s^m & D_s \end{bmatrix} \begin{bmatrix} x_s \\ v_s^p \\ v_s^m \\ u_s \end{bmatrix} \tag{2}$$

where v_s^m and v_s^p are interconnections to other subsystems(see Figure 1), and y_s and u_s are measured outputs and controlled inputs. The W_s^* terms represent information feedthrough between subsystems Σ_{s+1} and Σ_{s-1}. A generalization of this subsystem has appeared in [8] and associated papers. We will generally allow each subsystem Σ_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. Examples of such subsystem models are available in the literature, such as multiple vehicle systems [9], flight formations [20], offshore bases [21], and discretizations of various PDE's, [7], [22] etc.

If N of these subsystems(2) are connected together in a string (see Figure 1) with zero boundary inputs ($v_1^m = 0$,$v_N^p = 0$) and the interconnection variables are resolved, we obtain the interconnected system:

$$\bar{\Sigma} : \quad \begin{bmatrix} \dot{\bar{x}} \\ \bar{y} \end{bmatrix} = \begin{bmatrix} \bar{A} & \bar{B} \\ \bar{C} & \bar{D} \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{u} \end{bmatrix} \tag{3}$$

where the $\overline{\text{overline}}$ indicates a 'lifted' variable; for vectors: $\bar{x} = \begin{bmatrix} x_1^T & x_2^T & \dots & x_N^T \end{bmatrix}^T$, and the interconnected system matrices $(\bar{A}, \bar{B}, \bar{C}, \bar{D})$ have a very special structure, called 'Sequentially Semi Separable'(SSS). For example, for $N = 5$, we obtain equation (1) at the bottom of the page. Such matrices will be denoted:

$$\bar{A} = \mathcal{SSS}(B^m, W^m, C^m, A, B^p, W^p, C^p) \tag{4}$$

where the arguments of $\mathcal{SSS}()$ are called the 'generator' matrices of $\bar{A}$, and the $\mathcal{SSS}$ notation of the other matrices can be easily derived. This type of data-sparse structured matrix has recently been studied with respect to LTV(linear time varying) systems theory and inversion [23], scattering theory [24], and for their own sake [25]. The facts in which we are interested are that SSS matrices can be stored using only a linear amount of memory, there exist algorithms of only linear computational complexity($\mathcal{O}(N)$) for SSS matrix-matrix addition and multiplication, and inversion, 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[6]) 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.

In [19] efficient $\mathcal{O}(N)$ methods were developed to find a controller, $\bar{K} : \left[\begin{array}{c|c} \bar{A}_K & \bar{B}_K \\ \hline C_K & D_K \end{array}\right]$, for the distributed system (3), which has the same structure as the subsystems, and can be connected as in Figure 2. 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 [7] and [8], without any additional computation.

However, the systems $\bar{\Sigma}$ and $\bar{K}$ may still be very inefficiently represented, and since the computational complexity of SSS operations generally grows cubically with the size of the generator matrices[24], it is to our advantage to find an approximation of, for example, $\bar{\Sigma}$ with SSS matrices of smaller generators. For decreasing the 'SSS orders'(that is, $\max_s(\dim(W_s^m), \dim(W_s^p)))$ of $\bar{A}, \bar{B}, \bar{C}, \bar{D}$, a number of $\mathcal{O}(N)$ techniques are available [24][23] from LTV systems theory(see [19] for a discussion), but we might also want to actually decrease the state size of $\bar{\Sigma}$, i.e. decrease the size of $\bar{A} \in \mathbb{R}^{nN \times nN}$ to some smaller $n_{small}N$. This will be the focus of the following discussion.

In the next subsections we will list some results having to do with arithmetic operations on SSS matrices, and the computation and convergence of the matrix sign iteration; these are just for reference for the following results, for proofs and explanations see [19] and the references therein.

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

III. Computational Methods

For computational complexity, we use 'big O' notation,

Definition 1: 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.

A. Background: SSS matrices

For an SSS matrix: $\bar{A} = \mathcal{SSS}(B^m, W^m, C^m, A, B^p, W^p, C^p)$, $s \in \{1, 2, ...N\}$, many matrix-matrix operations are $\mathcal{O}(N)$, but cubic in the sizes of the generator matrices. For example, if $B_s^m, W_s^m, C_s^m, A_s, B_s^p, W_s^p, C_s^p \in \mathbb{R}^{n\times n}$, $\forall s \in \{1, 2, ...N\}$, then computing $\bar{A}^2$ will take $40n^3N$ flops [24].

However, under the original LTV interpretation of SSS matrices ([23]), it is 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 (W_s^p and W_s^m in $\bar{A}$), respectively. The class of SSS matrices of maximum lower and upper orders a_l and a_u with N diagonal terms is denoted as $\mathcal{SSS}^{a_l,a_u,N}$.

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

Lemma 1: For conformably partitioned matrices $\bar{A} \in \mathcal{SSS}^{a_l,a_u,N}$ and $\bar{B} \in \mathcal{SSS}^{b_l,b_u,N}$, then $\bar{A} + \bar{B} = \bar{C} \in \mathcal{SSS}^{c_l,c_u,N}$ where $c_l \leq a_l + b_l$, $c_u \leq a_u + b_u$, and $\bar{A}\bar{B} = \bar{D} \in \mathcal{SSS}^{d_l,d_u,N}$ 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 [24], [25]. ∎

This is important since we intend to use iterative algorithms for controller synthesis, and evidently each iteration will cost more than the previous. In practice, by considering the LTV interpretation of SSS matrices [23], one can use LTV model order reduction techniques individually on the lower triangular and upper triangular sections of the SSS matrix to decrease the SSS order. For certain types of iterations and problem data, the number of iterations to convergence can be bounded based on certain assumptions, and the results can be made robust to such model order reduction induced errors [19]. In the rest of this section, we will briefly review a few results for the matrix sign function for SSS matrices and SSS structure preserving permutations, which we then use in an $\mathcal{O}(N)$ structure preserving block diagonalization in section III-D, an integral part of the procedure for the main result of this paper: the structure preserving model order reduction in section IV.

B. $\mathcal{SSS}$ matrix sign computation and previous applications

The matrix sign function [26],[27], defined for a square matrix X with no eigenvalues on the imaginary axis, will be important in our computations.

Definition 3: [27] 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 [26]

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

For SSS matrices, $\bar{X}$, given the 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 $\bar{X} = PJP^{-1}$

which we will denote $\mathcal{A}$, then in [19] it was shown that using Algorithm 1:

Lemma 2: 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, $\bar{S} \in SSS^{\bullet,\bullet,N}$ to $sign(\bar{X})$ can be calculated to within some prespecified positive tolerance $\epsilon > \|\bar{S} - sign(\bar{X})\|_2$ in $\mathcal{O}(N)$.

We note that using this iteration, the sign iteration can be used to check the Hurwitz stability of an SSS matrix (or the Schur stability, through a Cayley transform)[19]. Also Lyapunov and Riccati equations for SSS matrices may be solved in a similar way, and that discrete time Lyapunov(Stein) equations may be converted to continuous time through a Cayley transformation and solved with the sign iteration, or in a more direct manner with Smith Squared iterations [28], on which can be derived similar convergence bounds.

C. Structure preserving permutations of SSS matrices

We can also permute SSS matrices together to form SSS block matrices, and vice versa (in the same way that block Toeplitz matrices may be permuted to form Toeplitz block matrices), where the SSS orders will increase only additively; i.e. we can move between the two equivalent representations:

$$\begin{bmatrix} \bar{e} \\ \bar{f} \end{bmatrix} = \begin{bmatrix} \bar{W} & \bar{X} \\ \bar{Y} & \bar{Z} \end{bmatrix} \begin{bmatrix} \bar{g} \\ \bar{h} \end{bmatrix}, \quad \begin{bmatrix} \bar{e} \\ f \end{bmatrix} = \bar{P} \begin{bmatrix} \bar{g} \\ h \end{bmatrix} \tag{5}$$

First we have:

Lemma 3: Given SSS matrices

$$\begin{aligned} \bar{W} &\in \mathcal{SSS}^{w_l,w_u,N}, \quad \bar{X} \in \mathcal{SSS}^{x_l,x_u,N} \\ \bar{Y} &\in \mathcal{SSS}^{y_l,y_u,N}, \quad \bar{Z} \in \mathcal{SSS}^{z_l,z_u,N} \end{aligned}$$

there exist permutation matrices such that

$$\Pi_L \begin{bmatrix} \bar{W} & \bar{X} \\ \bar{Y} & \bar{Z} \end{bmatrix} \Pi_R = \bar{P}$$

for some $\bar{P} \in \mathcal{SSS}^{p_l,p_u,N}$ where $p_l \leq w_l + x_l + y_l + z_l$, $p_u \leq w_u + x_u + y_u + z_u$. *Proof:* See [19] for formulas for $\bar{P}$ in terms of the generators of $\bar{W}, \bar{X}, \bar{Y}, \bar{Z}$. ∎

Further, the permutation matrices are also block SSS:

$$\Pi_L = \left[\mathcal{SSS}(0,0,0,\begin{bmatrix} I \\ 0 \end{bmatrix},0,0,0), \quad \mathcal{SSS}(0,0,0,\begin{bmatrix} 0 \\ I \end{bmatrix},0,0,0) \right]$$

$$\Pi_R = \begin{bmatrix} \mathcal{SSS}(0,0,0,\begin{bmatrix} I & 0 \end{bmatrix},0,0,0) \\ \mathcal{SSS}(0,0,0,\begin{bmatrix} 0 & I \end{bmatrix},0,0,0) \end{bmatrix}$$

The opposite permutation can also be done.

Lemma 4: Given some block SSS matrix $\bar{P} \in \mathcal{SSS}^{p_l,p_u,N}$ and lifted vectors $\begin{bmatrix} e \\ f \end{bmatrix}$, $\begin{bmatrix} g \\ h \end{bmatrix}$, then there exist permutation matrices such that

$$\Phi_L \bar{P} \Phi_R = \begin{bmatrix} \bar{W} & \bar{X} \\ \bar{Y} & \bar{Z} \end{bmatrix}$$

with some $\bar{W}, \bar{X}, \bar{Y}, \bar{Z} \in \mathcal{SSS}^{p_l,p_u,N}$. *Proof:* See [19] for formulas for the generators of $\bar{W}, \bar{X}, \bar{Y}, \bar{Z}$ in terms of the generators of $\bar{P}$. ∎

If all of the generators of the SSS matrices in Lemmas 3 and 4 are appropriately sized, then we have that $\Pi_L = \Pi_R^T = \Phi_L^T = \Phi_R$.

D. Structure Preserving Block Diagonalization

In this subsection, we will use both the structure preserving permutation and the sign iteration for a new application to SSS matrix computations; structure preserving block diagonalization. We will first show a simplified proof of a version of a well known result ([29]) for unstructured real matrices:

Lemma 5: Given some matrix $X \in \mathbb{R}^{z \times z}$ with positive real eigenvalues and some scalar α such that $X - \alpha I$ has Jordan decomposition

$$X - \alpha I = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} R & 0 \\ 0 & L \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}^{-1} \tag{6}$$

with $L \in \mathbb{R}^{x \times x}$, $R \in \mathbb{R}^{y \times y}$, $\lambda(L) < 0$, $\lambda(R) > 0$. If $0 \notin \lambda(P_{11}) \bigcup \lambda(P_{22})$, then

$$V = sign(X - \alpha I) + \underbrace{\begin{bmatrix} I_R & 0 \\ 0 & -I_L \end{bmatrix}}_{K} \tag{7}$$

block diagonalizes X: $VXV^{-1} = \begin{bmatrix} Y_1 & 0 \\ 0 & Y_2 \end{bmatrix}$, where $Y_1 \in \mathbb{R}^{x \times x}$, $Y_2 \in \mathbb{R}^{y \times y}$, $\lambda(Y_1) \bigcup \lambda(Y_2) = \lambda(X)$ with $\lambda(Y_1) > \alpha$ and $\lambda(Y_2) < \alpha$. *Proof:*

$$\begin{aligned} V &= sign(X - \alpha I) + K = PKP^{-1} + K \\ &= (PK + KP)P^{-1} = 2 \begin{bmatrix} P_{11} & 0 \\ 0 & -P_{22} \end{bmatrix} P^{-1} \\ VXV^{-1} &= 2 \begin{bmatrix} P_{11} & 0 \\ 0 & -P_{22} \end{bmatrix} P^{-1} X \frac{1}{2} P \begin{bmatrix} P_{11}^{-1} & 0 \\ 0 & -P_{22}^{-1} \end{bmatrix} \\ &= \begin{bmatrix} P_{11}(R + \alpha I)P_{11}^{-1} & 0 \\ 0 & P_{22}(L + \alpha I)P_{22}^{-1} \end{bmatrix} \end{aligned}$$

∎

The interesting and useful part of this result is that we need not actually compute the Jordan decomposition, only $sign(X)$ and K. Furthermore, while we don't know *a priori* the sizes of I_R and I_L, after computing $sign(X)$ it's easy to see that X has $\frac{N + Tr(sign(X))}{2}$ eigenvalues in the RHP and $\frac{N - Tr(sign(X))}{2}$ in the LHP.

Since these quantities can all be efficiently calculated for SSS matrices, we can apply this result to certain matrices with SSS structure, however, we often desire the resulting blocks to have structure $\mathcal{SSS}^{\bullet,\bullet,N}$, not $\mathcal{SSS}^{\bullet,\bullet,\frac{x}{z}N}$ and $\mathcal{SSS}^{\bullet,\bullet,\frac{y}{z}N}$, so we will also need some structure preserving permutations.

For some SSS matrix $\bar{X} \in \mathcal{SSS}^{\bullet,\bullet,N}$ with dimensions $\bar{X} \in \mathbb{R}^{zN \times zN}$ and positive real spectrum $\lambda(\bar{X}) \in \mathbb{R}_0^+$ we can:

1) Find an $\alpha > 0$ such that $\bar{X} - \alpha I$ has xN positive eigenvalues and yN negative eigenvalues(where $x + y = z$) and satisfies assumptions $\mathcal{A}$.
2) permute $\bar{X}$ (and thus also $sign(\bar{X} - \alpha I)$) into appropriate sized blocks:

$$\Phi \bar{X} \Phi^T = \begin{bmatrix} \bar{Y}_{11} & \bar{Y}_{12} \\ \bar{Y}_{21} & \bar{Y}_{22} \end{bmatrix} \begin{matrix} \}xN \\ \}yN \end{matrix} \tag{8}$$

3) and calculate $V = \Phi sign(\bar{X} - \alpha I)\Phi^T + \begin{bmatrix} I_R & 0 \\ 0 & -I_L \end{bmatrix}$ resulting in $V(\Phi \bar{X} \Phi^T)V^{-1} = \begin{bmatrix} \bar{Z}_1 & 0 \\ 0 & \bar{Z}_2 \end{bmatrix}$

Where $\bar{Z}_1, \bar{Z}_2 \in \mathcal{SSS}^{\bullet,\bullet,N}$, $\lambda(\bar{Z}_1) > \alpha$, $\lambda(\bar{Z}_2) < \alpha$. The first step, picking an appropriate α, can be accomplished by first guessing $\alpha_0 = \frac{Tr(\bar{X})}{Nz}$, computing the number of resulting positive and negative eigenvalues via $Tr(sign(\bar{X} - \alpha_0 I))$, and then bisecting for α_i until $\frac{N - Tr(sign(\bar{X} - \alpha_i I))}{2}$ is an appropriate integer multiple of N. Step 2 can be accomplished using Lemma 3, and step 3 can be accomplished using Algorithm 1 for SSS matrices.

Assuming that the number of α bisections is independent of N, and that $\bar{X} - \alpha I \in \mathcal{A}, \quad \forall N \in \mathbb{N}$, we thus have an $\mathcal{O}(N)$ procedure for block diagonalization of SSS matrices.

IV. Partial Balancing and Structure Preserving Model Order Reduction

We can now proceed with a method for partial balancing and truncation (see e.g. [30] for an overview using LMIs). Given some SSS system $\bar{G} = \left[\begin{array}{c|c} \bar{A} & \bar{B} \\ \hline \bar{C} & 0 \end{array}\right]$, we compute $\bar{P}, \bar{Q} \succeq 0$ such that (for discrete time):

$$\bar{A}\bar{P}\bar{A}^* - \bar{P} + \bar{B}\bar{B}^* \preceq 0 \tag{9}$$

$$\bar{A}^*\bar{Q}\bar{A} - \bar{Q} + \bar{C}^*\bar{C} \preceq 0 \tag{10}$$

Note that we do not have a general method of efficiently solving LMI's such as those above for SSS matrices; in practice, we use our sign or Smith squared iterative methods to approximately solve the corresponding Lyapunov equations for some small negative ϵI on the right hand side, and then check that the residual error is small enough such that the resulting $\bar{P}$ and $\bar{Q}$ satisfy (9) and (10)[9].

Since $\bar{P}, \bar{Q} \succeq 0$, $\bar{R} = \bar{P}\bar{Q}$ will have real non-negative spectrum $\lambda(\bar{R}) \in \mathbb{R}_0^+$. In the case where $\bar{P}, \bar{Q}$ solve the

Lyapunov *equations*, the square roots of $\lambda(\bar{R})$ will be the Hankel singular values of the system. If we then pick some $\alpha > 0$ and apply the partial diagonalization procedure described above to calculate V and Π such that

$$V^{(-1)}\Pi\bar{R}\Pi^T V = \begin{bmatrix} \bar{\Sigma}_1 & 0 \\ 0 & \bar{\Sigma}_2 \end{bmatrix} \tag{11}$$

with $\lambda(\bar{\Sigma}_1) = \{\lambda(R) > \alpha\}$, $\lambda(\bar{\Sigma}_2) = \{\lambda(R) < \alpha\}$, and perform the corresponding 'partial balancing' state transformation: $\left[\begin{array}{c|c} V^{-1}\Pi^*\bar{A}\Pi V & V^{-1}\Pi^*\bar{B} \\ \hline \bar{C}\Pi V & 0 \end{array}\right] = \left[\begin{array}{cc|c} \hat{\bar{A}}_{11} & \hat{\bar{A}}_{12} & \hat{\bar{B}}_1 \\ \hat{\bar{A}}_{21} & \hat{\bar{A}}_{22} & \hat{\bar{B}}_2 \\ \hline \hat{\bar{C}}_1 & \hat{\bar{C}}_2 & 0 \end{array}\right]$ then the truncated system $\hat{\bar{G}} = \left[\begin{array}{c|c} \hat{\bar{A}}_{11} & \hat{\bar{B}}_1 \\ \hline \hat{\bar{C}}_1 & 0 \end{array}\right]$ can be shown to only be a state transformation away from the fully balanced truncated system[31], and thus has the same stability properties and error bound: $\hat{\bar{A}}_{11}$ is guaranteed stable and $\|\bar{G} - \hat{\bar{G}}\|_\infty < 2\sum(\lambda_i(\bar{\Sigma}_2))^{1/2}$, but the realization of $\hat{\bar{G}}$ retains the $\mathcal{SSS}^{\bullet,\bullet,N}$ structure. We note that this error upper bound can be efficiently calculated as $2Tr((\bar{\Sigma}_2)^{1/2})$ interpreted as the matrix square root, which can also be computed using the matrix sign [32].

Remark 1: In practice, the $0 \notin \lambda(P_{11}) \bigcup \lambda(P_{22})$ condition in Lemma 5 is not often violated when performing SSS model order reductions as discussed above, however, it does happen that either the spectrum of $\bar{R}$ has only a small 'gap' between $\lambda_{n_r N}$ and $\lambda_{n_r N+1}$ (for ordered eigenvalues, where $n_r N$ is the size of the desired $\bar{\Sigma}_1$), or P_{11} and P_{22} are ill conditioned due to near permutations in P, and hence either the calculation of the similarity transformation $\bar{V}$, or $\bar{V}$ itself, may be ill-conditioned. This may result in SSS matrices $\hat{\bar{A}}_{11}, \hat{\bar{B}}_1, \hat{\bar{C}}_1$ with weakly stable or unstable generators (W_s^m and W_s^p terms which are LTV marginally stable or unstable), which are difficult to implement due to the resulting large off-diagonal terms. This can often be remedied by solving Lyapunov inequalities of the form:

$$\bar{A}\bar{P}\bar{A}^* - \bar{P} + \bar{B}\bar{B}^* + \bar{T} \preceq 0 \tag{12}$$

$$\bar{A}^*\bar{Q}\bar{A} - \bar{Q} + \bar{C}^*\bar{C} + \bar{T} \preceq 0 \tag{13}$$

Where $\bar{T}$ is a block diagonal SSS matrix with realization $\bar{T} = \mathcal{SSS}(0,0,0,\begin{bmatrix} \kappa I_{n_r} & 0 \\ 0 & 0 \end{bmatrix}, 0,0,0)$ on each of its diagonals, for some $\kappa > 0$. This perturbs the spectra of $\bar{P}$ and $\bar{Q}$ and thus $\bar{R}$, at the expense of often increasing the resulting approximation error (see below), However, since solutions to (12) and (13) will still be valid solutions to (9) and (10, the error upper bound $2Tr((\bar{\Sigma}_2)^{1/2})$ can still be used.

V. Example

We pick a computational example with no physical meaning in order to better exhibit some of the issues mentioned above. Consider the discrete time system $\bar{G}: \begin{bmatrix} \bar{x}_{k+1} \\ \bar{y}_k \end{bmatrix} = \begin{bmatrix} \bar{A} & \bar{B} \\ \bar{C} & 0 \end{bmatrix} \begin{bmatrix} \bar{x}_k \\ \bar{u}_k \end{bmatrix}$ where we have picked $\bar{A} = -\frac{1}{10}\mathcal{SSS}(\begin{bmatrix} -2 \\ 1 \end{bmatrix}, -.1, \begin{bmatrix} 1 & 2.5 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 2 & .5 \end{bmatrix}, \begin{bmatrix} .3 \\ -.01 \end{bmatrix}, .7, \begin{bmatrix} 9 & 5 \end{bmatrix})$,

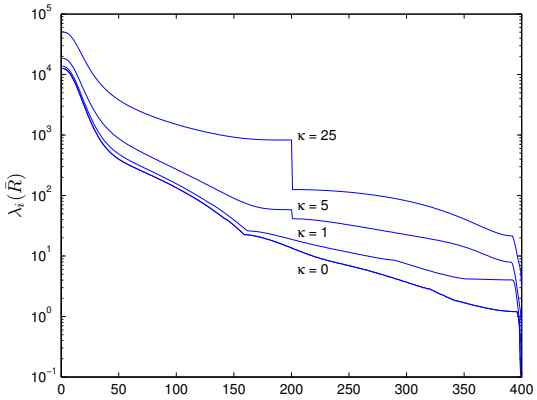

Fig. 3. ordered eigenvalues of $\bar{R}$ for different values of κ

$\bar{B} = \mathcal{SSS}(\begin{bmatrix} 1.6 \\ -.7 \end{bmatrix}, .4, -1.5, \begin{bmatrix} 1.3 \\ .2 \end{bmatrix}, \begin{bmatrix} 1.7 \\ -.2 \end{bmatrix}, .7, 1.7)$, $\bar{C} = \mathcal{SSS}(2.6, -.5, \begin{bmatrix} 0.2 & 1.7 \end{bmatrix}, \begin{bmatrix} .5, .3 \end{bmatrix}, 3.5, .4, \begin{bmatrix} .5 & -.5 \end{bmatrix})$ to be spatially invariant for $s = 1:200$ for simplicity, although as in [19], the computational methods and complexity hold also for heterogeneous sytems. The original system with state dimension 400 has l_2 induced norm $\|\bar{G}\|_\infty \approx 197.2$ and with MATLAB's `reduce()` using balanced truncation, a non-structured reduced order system $\hat{G}$ with state dimension 200 was found with $\|\bar{G} - \hat{G}\|_\infty \approx 7.4565$. Using the SSS structured methods (with SSS order of 10 for all calculations) above with $\kappa = 5$, an SSS structured reduced order system $\hat{\bar{G}}$ with state dimension 200 was found with $\|\bar{G} - \hat{\bar{G}}\|_\infty \approx 6.3795$. In this case, only 3 α-iterations were needed to find the correct value for splitting the eigenvalues, and by extending N to larger values, it was found that the estimate of linear computational complexity($\mathcal{O}(N)$) holds, as in [19] and [9], and the SSS model order reduction routines become faster than the MATLAB routines at $N \approx 350$ and ≈ 110 seconds.

In figure 3 we see how changing κ in (12) and (13) affects the eigenvalues of $\bar{R}$ for this problem. For $\kappa = 0$, there is not much gap(leading to more α-iterations), the resulting $\bar{V}$ matrix is ill-conditioned, and $\hat{\bar{A}}_{11}$ has large off diagonal elements (see figure 4) making it difficult to implement. However, for $\kappa = 5$ the resulting gap in $\lambda(R)$ is noticeable, leading to faster α-bisection convergence and a nicer realization of $\hat{\bar{A}}_{11}$ (see figure 5). Note that the figures show the entrywise $\log_{10}$ of the absolute values of the matrices, and we thus see that $\hat{\bar{A}}_{11}$ has an exponential spatial decay away from the diagonal(see the ESD operators of [33]), making it much easier to implement distributed controllers.

However, unnecessarily large κ values lead to decreased accuracy in the approximation, for example, with $\kappa = 25$ we have $\|\bar{G} - \hat{\bar{G}}\|_\infty \approx 8.0516$ and for $\kappa = 100$ we have $\|\bar{G} - \hat{\bar{G}}\|_\infty \approx 10.2025$, but an appropriate κ value can be iteratively found by bisection and checking that the off diagonal corners of $\bar{V}^{(-1)}$ approach 0.

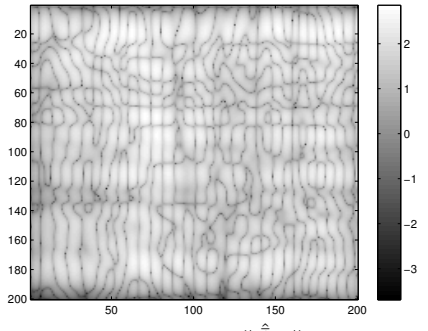

Fig. 4. Entrywise $\log_{10}(|\hat{\tilde{A}}_{11}|)$ for $\kappa = 0$

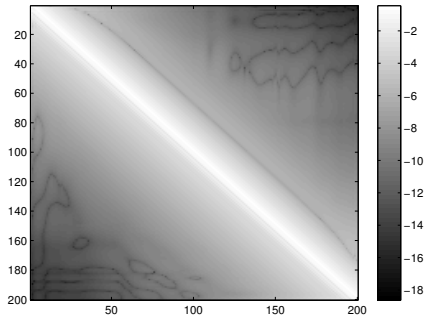

Fig. 5. Entrywise $\log_{10}(|\hat{\tilde{A}}_{11}|)$ for $\kappa = 5$

VI. Conclusion

In conclusion, we have described and demonstrated a model order reduction technique for certain 1-D heterogeneous distributed systems that scale as $\mathcal{O}(N)$, better than existing interconnected system methods [12], but preserve the distributed interconnection structure, unlike the computationally efficient but structure destroying rank-revealing factorization techniques proposed within the numerical linear algebra community [17].

The methods described herein can easily be extended to continuous time and other balanced truncation techniques such as are listed in [34], and probably also to $\mathcal{H}$ matrices [6], although the authors are unaware of the desirability of similarly structured reduced order models in such applications.

Acknowledgment

The authors thank Paolo Massioni, Jan-Willem van Wingerden, Tamas Keviczky, and the anonymous reviewers.

References

[1] M. Högberg, T. Bewley, and D. Henningson, "Relaminarization of Re_τ turbulence using gain scheduling and linear state-feedback control," *Physics of Fluids*, vol. 15, pp. 3572–3575, 2003.

[2] A. M. Bayen, R. L. Raffard, and C. J. Tomlin, "Adjoint-based constrained control of Eulerian transportation networks: Application to air traffic control," *Proc. Amer. Control Conf.*, pp. 5539–5545, 2004.

[3] M. R. Jovanović, M. Arcak, and E. D. Sontag, "Remarks on the stability of spatially distributed systems with a cyclic interconnection structure," *Proc. Amer. Control Conf.*, pp. 2696–2701, 2007.

[4] A. D. Hansen, P. Sorensen, F. Iov, and F. Blaabjerg, "Centralised power control of wind farm with doubly fed induction generators," *Renewable Energy*, vol. 31, pp. 935–951, 2006.

[5] "Special issue on control of industrial spatially distributed processes," *IEEE Trans. on Control Syst. Technol.*, vol. 11, pp. 609–773, 2003.

[6] 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.

[7] R. D'Andrea and G. E. Dullerud, "Distributed control design for spatially interconnected systems," *IEEE Trans. Autom. Control*, vol. 48, pp. 1478–1495, 2003.

[8] G. E. Dullerud and R. D'Andrea, "Distributed control of heterogeneous systems," *IEEE Trans. Autom. Control*, vol. 49, pp. 2113–2128, 2004.

[9] J. K. Rice and M. Verhaegen, "Distributed control: A sequentially semi-separable approach," *Proc. IEEE Conf. Decision and Control*, 2008.

[10] C. L. Beck, J. Doyle, and K. Glover, "Model reduction of multidimensional and uncertain systems," *IEEE Trans. Autom. Control*, vol. 41, pp. 1466–1477, 1996.

[11] C. Beck and R. D'Andrea, "Simplification of spatially distributed systems," *Proc. IEEE Conf. Decision and Control*, 1999.

[12] S. Samar and C. Beck, "Model reduction of heterogeneous distributed systems," *Proc. IEEE Conf. Decision and Control*, 2003.

[13] H. Sandberg and R. M. Murray, "Model reduction of interconnected linear systems using structured gramians," *IFAC World Congress*, 2008.

[14] J. A. Atwell and B. B. King, "Reduced order controllers for spatially distributed systems via proper orthogonal decomposition," *SIAM Journal of Scientific Computing*, vol. 26, pp. 128–151, 2004.

[15] J. M. Badia, P. Benner, R. Mayo, and E. S. Quintana-Orti, "Parallel algorithms for balanced truncation model reduction of sparse systems," *Lecture Notes in Computer Science, Springer*, 2006.

[16] A. Antoulas, D. Sorensen, and S. Gugercin, "A survey of model reduction methods for large-scale systems," *Contemporary mathematics*, vol. 280, pp. 193–219, 2001.

[17] U. Baur and P. Benner, "Gramian-based model reduction for data-sparse systems," *Preprint*, 2007.

[18] L. Li and F. Paganini, "Structured coprime factor model reduction based on LMIs," *Automatica*, vol. 41, pp. 145–151, 2005.

[19] J. K. Rice and M. Verhaegen, "Distributed control: A sequentially semi-separable approach for heterogeneous linear systems," *IEEE Trans. Autom. Control*, vol. Accepted for Publication, temporarily available at http://www.dcsc.tudelft.nl/∼jrice/Publications.htm, 2009.

[20] J. Fowler and R. D'Andrea, "A formation flight experiment," *IEEE Control Systems Magazine*, vol. 23, pp. 35–43, 2003.

[21] P. Sharma and C. Beck, "Modelling and distributed control of mobile offshore bases," *Proc. Amer. Control Conf.*, pp. 5238–5243, 2004.

[22] F. Wu and S. E. Yildizoglu, "Distributed parameter-dependent modeling and control of flexible structures," *Transactions of the ASME*, vol. 127, pp. 230–239, 2005.

[23] P. Dewilde and A. J. V. D. Veen, *Time-Varying systems and Computations*. Kluwer Academic Publishers, 1998.

[24] 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," *Report, Lawrence Livermore National Laboratory*, 2003.

[25] Y. Eidelman and I. Gohberg, "On a new class of structured matrices," *Integral Equations and Operator Theory*, vol. 34, pp. 292–324, 1999.

[26] J. Roberts, "Linear model reduction and solution of the algebraic Riccati equation by use of the sign function," *International Journal of Control*, vol. 32, pp. 677–687, 1980.

[27] C. S. Kenney and A. J. Laub, "The matrix sign function," *IEEE Trans. Autom. Control*, vol. 40, pp. 1330–1348, 1995.

[28] R. Smith, "Matrix equation $XA + BX = C$," *SIAM Journal of Applied Mathematics*, vol. 16, p. No. 1, 1968.

[29] A. Beavers and E. Denman, "A computational method for eigenvalues and eigevectors of a matrix with real eigenvalues," *Numer. Math.*, pp. 389–396, 1973.

[30] G. E. Dullerud and F. G. Paganini, *A Course in Robust Control Theory: a convex approach*. Springer, 2000.

[31] A. Varga, "Balancing free square-root algorithm for computing singular perturbation approximations," *Proc. IEEE Conf. Decision and Control*, pp. 1062–1065, 1991.

[32] N. J. Higham, "Stable iterations for the matrix square root," *Numerical Algorithms*, vol. 15, pp. 227–242, 1997.

[33] N. Motee and A. Jadbabaie, "Optimal control of spatially distributed systems," *IEEE Trans. Autom. Control*, to be published.

[34] S. Gugercin and A. C. Antoulas, "A survey of model reduction by balanced truncation and some new results," *International Journal of Control*, vol. 77, pp. 748–766, 2004.