2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Square-root H_2 and H_∞ Synthesis Algorithms for Sequentially Semi-separable Systems

Ruxandra Mustata
Delft Center for Systems and Control
Delft University of Technology
2628 CD Delft, The Netherlands
Email: r.i.mustata@tudelft.nl

Rufus Fraanje
Delft Center for Systems and Control
Delft University of Technology
2628 CD Delft, The Netherlands
Email: p.r.fraanje@tudelft.nl

Michel Verhaegen
Delft Center for Systems and Control
Delft University of Technology
2628 CD Delft, The Netherlands
Email: m.verhaegen@tudelft.nl

***Abstract*—We consider the problem of designing H_2 and H_∞ linear estimators for time-varying spatially interconnected systems distributed in one spatial dimension. In general, numerical implementation of the algebraic Riccati equation (ARE) solution for such systems is a complex and computationally expensive operation. However, the spatially interconnected systems can be described by state-space models whose matrices have a special structure, called "sequentially semi-separable" (SSS). Using only efficient and structure-preserving arithmetic operations, the H_2 and H_∞ estimation problems are solved by means of square-root array algorithms. Our solution has thus, linear computational complexity and is a viable approach for large scale systems.**

I. Introduction

The subject of estimation and control theory assesses a fundamental importance to H_2 and H_∞ filtering problems, as their solution is required in many cases. As the mathematical models of large-scale distributed physical systems have increased dimensionality, the numerical computation of the algebraic Riccati equation (ARE) solution is a complex and computationally expensive task [1]. However, most of these complex large-scale systems consist of spatially distributed interconnected subsystems and efficient distributed methods have been derived for solving estimation and control problems.

The model class of spatially distributed interconnected subsystems describes very well: a number of physical systems involving discretized partial differential equations, such as flexible structures [2], turbulent flow relaminarization [3]; heterogeneous subsystems with various interconnection patterns, such as highway traffic control [4], formation flights [5], large adaptive mirrors for telescopes [6]; and time-space correlated shifts, such as frozen-flow propagating wavefronts [7].

These systems typically have very high dimensionality. Given a number of N interconnected subsystems, each of order n, the state-space model of the overall system will have an order of nN, thus most matrix arithmetic operations will have a computational complexity of $\mathcal{O}(n^3N^3)$. In this framework, a traditional approach for solving optimal estimation or control problems is a very computationally expensive operation. Therefore, the need for developing efficient analysis and controller synthesis techniques has arisen. Structure preserving arithmetics along with fast ($\mathcal{O}(n^3N)$), iterative and structure preserving ARE solvers that exploits the interconnected system matrix structure have been derived by Rice *et al.* [8]. Their proposed method for solving the ARE is based on the matrix sign function, which might be slow convergent and numerically instable if no other expensive iterative refinement is used [9], [10]. Currently, the preferred method for solving H_2 and H_∞ filtering problems involves square-root array algorithms, which are typically more numerically stable and reliable [11].

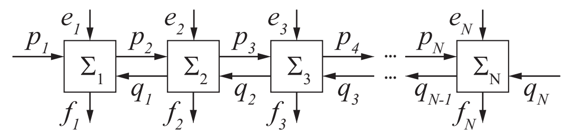

Fig. 1. A string of N interconnected subsystems.

The goal of this paper is to derive a square-root array algorithm for efficiently ($\mathcal{O}(n^3N)$) solving the ARE's involved in the H_2 and H_∞ estimation problems, for spatially-varying one-dimensional string interconnected systems.

The paper is structured as follows: in Section 2, we present an overview of the subsystem models and interconnection structure. In Section 3 we formulate the H_2 and H_∞ estimation problems and we approach them from a square-root algorithmic point of view. In Section 4 we adapt the square-root algorithm to the SSS structured matrix framework. Section 5 presents the experimental results of our algorithm. Finally, Section 6 concludes the paper.

II. Preliminaries

A. Sequentially semi-separable matrices

The subsystem model and interconnections considered in this paper are such as those discussed in [8]. The structure of such interconnected subsystems is presented in Figure 1. A subsystem Σ_s is described by a linear model of the form

$$\begin{bmatrix} f_s \\ q_{s-1} \\ p_{s+1} \end{bmatrix} = \begin{bmatrix} D_s & P_s & U_s \\ Q_s & R_s & 0 \\ V_s & 0 & W_s \end{bmatrix} \begin{bmatrix} e_s \\ q_s \\ p_s \end{bmatrix}, \tag{1}$$

where $e_s \in \mathbb{R}^{n_{e,s}}$ and $f_s \in \mathbb{R}^{n_{f,s}}$ are the input and the output vectors of the subsystem Σ_s and $q_s \in \mathbb{R}^{n_{q,s}}$ and $p_s \in \mathbb{R}^{n_{p,s}}$ describe the interconnections between subsystems.

When considering a string of N interconnected subsystems of type (1), with zero boundary inputs ($p_1 = 0, q_N = 0$), we can derive the behavior of the interconnected system by 'lifting' the input and output vectors over the index $s \in \{1, 2, \ldots, N\}$, e.g. $\overline{e} = [e_1^T \; e_2^T \; \ldots \; e_N^T]^T$ and $\overline{f} = [f_1^T \; f_2^T \; \ldots \; f_N^T]^T$, where $e_i \in \mathbb{R}^{n_{e,i}}$ and $f_i \in \mathbb{R}^{n_{f,i}}$, and by resolving the interconnection variables. Thus, the lifted linear equation describing the behavior of the system will be

$$\overline{f} = \overline{S}\,\overline{e}, \tag{2}$$

where $\overline{e} \in \mathbb{R}^{N_e}$ and $\overline{f} \in \mathbb{R}^{N_f}$ are the lifted input and output vectors, with $N_e = n_{e,1} + \ldots + n_{e,N}$ and $N_f = n_{f,1} + \ldots + n_{f,N}$.

The consequent matrix $\overline{S} \in \mathbb{R}^{N_e \times N_f}$ has a special block-partitioned structure called 'Sequentially Semi-Separable' (SSS). We explicitly write the block-partitioned matrix $\overline{S}$ as follows:

$$\overline{S} = [S_{s,t}], \text{ for } s,t = 1, 2, \ldots N,$$

where $S_{s,t} \in \mathbb{R}^{n_{e,s} \times n_{f,t}}$ and

$$S_{s,t} = \begin{cases} D_s, & \text{if } s = t, \\ P_s R_{s+1} \ldots R_{t-1} Q_t, & \text{if } t > s, \\ U_s W_{s-1} \ldots W_{t+1} V_t, & \text{if } t < s. \end{cases}$$

Note that the SSS structured matrix $\overline{S}$ is defined only by some lower-dimensional blocks D_s, P_s, Q_s, R_s, U_s, V_s, W_s, which are called 'generators' of a SSS matrix. For such matrices, we will use the following notation:

$$\overline{S} = SSS(P_s, R_s, Q_s, D_s, U_s, W_s, V_s). \tag{3}$$

Such a structure has been efficiently exploited in [12], where usual arithmetic operations, such as matrix addition, multiplication, inversion, permutation and transposition, are performed at the level of generators. Efficient methods for solving more complex numerical problems, such as QR factorization [13], Lyapunov or Ricatti solvers [8] or model order reductions [14], have been further developed. Moreover, the class of SSS matrices is closed under all these operations.

B. Permutations of sequentially semi-separable matrices

Typically, in the context of array algorithms, one must form a prearray out of given data, perform a sequence of elementary operations and then read the quantities of interest out of the resulting postarray. For example, the square-root algorithms for solving H_2 and H_∞ estimation problems, as illustrated in the subsequent equations (16) and (24), require the triangularization of the prearrays in the left terms via LQ factorizations. In [13], Eidelman *et al.* proposed an efficient ($\mathscr{O}(n^3 N)$) algorithm for performing QR factorizations for SSS matrices. We will use this algorithm to derive its equivalent form for LQ factorizations for SSS matrices. However, in the subsequent equations (16) and (24) below, the resulting prearrays will not be SSS, but block SSS, i.e. they are column and/or row stacking of SSS matrices. Therefore, the procedure for the QR factorization, as presented in [13], can not be directly applied for the block SSS prearrays. Permuting rows and columns of block SSS matrices can produce equivalent SSS matrices and vice versa. These permutations are essential in order to preserve the linear complexity of the QR factorization and to avoid using more computationally expensive block arithmetic. In the following, we will explain how a block SSS matrix can be permuted into a SSS matrix and what are the consequences of these operations at the level of generators.

For example, let us consider two SSS systems of type (2), described by two SSS matrices $\overline{S}_i = SSS(P_{s,i}, R_{s,i}, Q_{s,i}, D_{s,i}, U_{s,i}, W_{s,i}, V_{s,i})$, with $i \in \{1, 2\}$. In order to show the row permutation of two SSS matrices, let us assume that the same input $\overline{e}$ is applied to both systems. If we concatenate the two resulting output vectors, the input-output dependency of the overall system can be written as

$$\begin{bmatrix} \overline{f}_1 \\ \overline{f}_2 \end{bmatrix} = \begin{bmatrix} \overline{S}_1 \\ \overline{S}_2 \end{bmatrix} \overline{e}. \tag{4}$$

However, we are interested in obtaining a linear dependency involving a SSS matrix, as in (2), not a two-block SSS matrix, as in (4). To this purpose, a permutation matrix Π_L is used for shuffling the output vector of (4), along with the corresponding rows of the two-block SSS matrix, such that $\Pi_L \begin{bmatrix} \overline{f}_1 \\ \overline{f}_2 \end{bmatrix} = \overline{f}$ and $\Pi_L \begin{bmatrix} \overline{S}_1 \\ \overline{S}_2 \end{bmatrix} = \overline{S}$. The generators of the SSS matrix $\overline{S}$ will be: $D_s = \left[D_{s,1}^T D_{s,2}^T\right]^T$, $P_s = diag(P_{s,1}, P_{s,2})$, $R_s = diag(R_{s,1}, R_{s,2})$, $Q_s = \left[Q_{s,1}^T Q_{s,2}^T\right]^T$, $U_s = diag(U_{s,1}, U_{s,2})$, $W_s = diag(W_{s,1}, W_{s,2})$, $V_s = \left[V_{s,1}^T V_{s,2}^T\right]^T$.

A similar approach is used when dealing with column permutation of two SSS matrices. Let us consider an output vector $\overline{f}$ that is obtained as a linear combination of two input vectors, such as

$$\overline{f} = \left[\overline{S}_1 \; \overline{S}_2\right] \begin{bmatrix} \overline{e}_1 \\ \overline{e}_2 \end{bmatrix}. \tag{5}$$

Now a permutation matrix Π_R is used for shuffling the input vector of (5), along with the corresponding columns of the two-block SSS matrix, such that another system of type (2) is derived, with $\left[\overline{S}_1 \; \overline{S}_2\right] \Pi_R = \overline{S}$ and $\Pi_R^T \begin{bmatrix} \overline{e}_1 \\ \overline{e}_2 \end{bmatrix} = \overline{e}$. The SSS matrix $\overline{S}$ is given by the following generators: $P_s = \begin{bmatrix} P_{s,1} & P_{s,2} \end{bmatrix}$, $R_s = diag(R_{s,1}, R_{s,2})$, $Q_s = diag(Q_{s,1}, Q_{s,2})$, $D_s = \begin{bmatrix} D_{s,1} & D_{s,2} \end{bmatrix}$, $U_s = \begin{bmatrix} U_{s,1} & U_{s,2} \end{bmatrix}$, $V_s = diag(V_{s,1}, V_{s,2})$, $W_s = diag(W_{s,1}, W_{s,2})$.

C. Sequentially semi-separable structured systems

In the following, we will show how the representation (1) can be used in order to derive a linear time-varying state-space model with SSS matrices. Let us consider the following subsystem model:

$$\begin{bmatrix} x_{s,k+1} \\ q_{s-1,k} \\ p_{s+1,k} \\ y_{s,k} \end{bmatrix} = \begin{bmatrix} A_{s,k} & B_{s,k}^q & B_{s,k}^p & B_{s,k} & 0 \\ C_{s,k}^q & W_{s,k}^q & 0 & L_{s,k}^q & J_{s,k}^q \\ C_{s,k}^p & 0 & W_{s,k}^p & L_{s,k}^p & J_{s,k}^p \\ C_{s,k} & H_{s,k}^q & H_{s,k}^p & 0 & D_{s,k} \end{bmatrix} \begin{bmatrix} x_{s,k} \\ q_{s,k} \\ p_{s,k} \\ w_{s,k} \\ v_{s,k} \end{bmatrix},$$

where $x_{s,k} \in \mathbb{R}^{n_{x,s}}$ is the state vector, $y_{s,k} \in \mathbb{R}^{n_{y,s}}$ is the output vector, $w_{s,k} \in \mathbb{R}^{n_{w,s}}$ and $v_{s,k} \in \mathbb{R}^{n_{v,s}}$ are the process and the measurement noise, respectively. After resolving the interconnection variables $q_{s,k}$ and $p_{s,k}$, the lifted system will have a state-space realization of the form

$$\begin{cases} \overline{x}_{k+1} &= \overline{A}_k\overline{x}_k + \overline{B}_k\overline{w}_k \\ \overline{y}_k &= \overline{C}_k\overline{x}_k + \overline{D}_k\overline{v}_k \end{cases} \tag{6}$$

where $\overline{x}_k \in \mathbb{R}^{N_x}$, $\overline{y}_k \in \mathbb{R}^{N_y}$, $\overline{w}_k \in \mathbb{R}^{N_w}$ and $\overline{v}_k \in \mathbb{R}^{N_v}$ are the lifted state, output, process and measurement noise vectors, respectively, with $N_x = n_{x,1} + \ldots + n_{x,N}$ and N_y, N_w and N_v defined similarly. By using the notation (3), the SSS structured matrices $\overline{A}_k$, $\overline{B}_k$, $\overline{C}_k$ and $\overline{D}_k$ can be written in terms of their generators as follows:

$$\overline{A}_k = SSS(B^q_{s,k}, W^q_{s,k}, C^q_{s,k}, A_{s,k}, B^p_{s,k}, W^p_{s,k}, C^p_{s,k}), \tag{7}$$

$$\overline{B}_k = SSS(B^q_{s,k}, W^q_{s,k}, L^q_{s,k}, B_{s,k}, B^p_{s,k}, W^p_{s,k}, L^p_{s,k}), \tag{8}$$

$$\overline{C}_k = SSS(H^q_{s,k}, W^q_{s,k}, C^q_{s,k}, C_{s,k}, H^p_{s,k}, W^p_{s,k}, C^p_{s,k}), \tag{9}$$

$$\overline{D}_k = SSS(H^q_{s,k}, W^q_{s,k}, J^q_{s,k}, D_{s,k}, H^p_{s,k}, W^p_{s,k}, J^p_{s,k}). \tag{10}$$

III. ESTIMATION PROBLEMS

In this section we state the H_2 and H_∞ estimation problems and we present their solutions, *cf* [11].

Recall the LTV state-space model (6). Without loss of generality, in the following we will consider that the measurement vector is directly corrupted by noise, thus $J^q_{s,k}$ and $J^p_{s,k}$ are zero matrices and $D_{s,k}$ are identity matrices. The model (6) becomes

$$\begin{cases} \overline{x}_{k+1} &= \overline{A}_k\overline{x}_k + \overline{B}_k\overline{w}_k \\ \overline{y}_k &= \overline{C}_k\overline{x}_k + \overline{v}_k \end{cases} \tag{11}$$

with known state-space matrices $\overline{A}_k$, $\overline{B}_k$ and $\overline{C}_k$ having the SSS structure defined by (7)-(9), with unknown initial state $\overline{x}_0$ and unknown stochastic processes $\overline{w}_k$ and $\overline{v}_k$.

The estimation problem that we typically want to solve consists in finding some linear combination of the state

$$\overline{z}_k = \overline{L}_k\overline{x}_k,$$

where $\overline{z}_k \in \mathbb{R}^{N_z}$ and $\overline{L}_k$ is a known matrix, using the set of observations $\{\overline{y}_0, \ldots, \overline{y}_{k-1}\}$. If we denote the estimate of $\overline{z}_k$ based on the observations from time 0 to time $k-1$ with $\hat{\overline{z}}_{k|k-1} = \mathscr{F}(\overline{y}_0, \ldots, \overline{y}_{k-1})$, where $\mathscr{F}(\cdot)$ is a linear operator, then the corresponding estimation error will be $\tilde{\overline{z}}_{k|k-1} = \overline{z}_k - \hat{\overline{z}}_{k|k-1}$. The main goal in estimation problems is making this quantity small in a certain sense. The H_2 and H_∞ estimation problems make different assumptions on the stochastic processes involved in the system description and use different cost functions for evaluating the estimation error.

A. H_2 estimation problem

In H_2 estimation problems, the stochastic exogenous signals are assumed to be zero-mean white noise sequences with known statistical properties. On this basis, we can define the joint covariance matrix of the noise sequences as

$$\mathbf{E}\begin{bmatrix}\overline{w}_k \\ \overline{v}_k\end{bmatrix}\begin{bmatrix}\overline{w}_l^T & \overline{v}_l^T\end{bmatrix} = \begin{bmatrix}\overline{Q}_k & \overline{S}_k \\ \overline{S}_k^T & \overline{R}_k\end{bmatrix}\delta_{k-l}, \tag{12}$$

where δ_k is the unit pulse. Moreover, we point out that the measurement noise covariance matrix $\overline{R}_k$ is assumed to be positive definite.

The goal of the H_2 estimation problem is to find a linear estimate of $\overline{z}_k$ such that the mean squared estimation error is minimized. We can now formulate the H_2 estimation problem.

Problem 1: (H_2 Estimation Problem) Given the state-space model (6) with known measurement sequence $\{\overline{y}_0, \ldots, \overline{y}_{k-1}\}$ and unknown zero-mean white-noise sequences $\overline{w}_k$ and $\overline{v}_k$ having known joint covariance matrix given by (12), find a linear estimate of $\overline{z}_k$, denoted by $\hat{\overline{z}}_{k|k-1}$, that minimizes the mean squared estimation error, i.e.,

$$\min_{\mathscr{F}(.)} \mathbf{E}\sum_{j=0}^{k}(\overline{z}_k - \hat{\overline{z}}_{k|k-1})^T(\overline{z}_k - \hat{\overline{z}}_{k|k-1}).$$

The solution to the H_2 estimation problem is the minimum variance unbiased estimation, given by the Kalman filter.

Theorem 1: (H_2/Kalman Filter) [11] The solution to Problem 1 is given by the linear estimate $\hat{\overline{z}}_{k|k-1} = \overline{L}_k\hat{\overline{x}}_{k|k-1}$, where $\hat{\overline{x}}_{k|k-1}$ denotes the 1-step ahead prediction of the state, given by the following recursion

$$\hat{\overline{x}}_{k+1|k} = \overline{A}_k\hat{\overline{x}}_{k|k-1} + \overline{K}_k(\overline{y}_k - \overline{C}_k\hat{\overline{x}}_{k|k-1}).$$

Here,

$$\overline{K}_k = (\overline{S}_k + \overline{A}_k\overline{P}_{k|k-1}\overline{C}_k^T)R_{e,k}^{-1}, \tag{13}$$

$$\overline{R}_{e,k} = \overline{R}_k + \overline{C}_k\overline{P}_{k|k-1}\overline{C}_k^T, \tag{14}$$

and $\overline{P}_{k|k-1}$ is the state estimation error covariance matrix (EECM) and satisfies the following Riccati recursion

$$\begin{aligned}\overline{P}_{k+1|k} = {} & \overline{A}_k\overline{P}_{k|k-1}\overline{A}_k^T + \overline{B}_k\overline{Q}_k\overline{B}_k^T - (\overline{S}_k + \overline{A}_k\overline{P}_{k|k-1}\overline{C}_k^T)\times \\ & \times\overline{R}_{e,k}^{-1}(\overline{S}_k + \overline{A}_k\overline{P}_{k|k-1}\overline{C}_k^T)^T.\end{aligned} \tag{15}$$

The state EECM can also be derived by means of a square-root array algorithm, which will be described in the following.

For the state-space model (6), let the initial state, $\overline{x}_{0|-1}$, and the square-root of its covariance matrix, $\overline{P}^{1/2}_{0|-1}$, be given. The state EECM $\overline{P}_{k+1|k}$ can be derived by finding an orthogonal transformation Φ_k such that, for $k = 0, 1, 2, \ldots$,

$$\begin{bmatrix}\overline{C}_k\overline{P}^{1/2}_{k|k-1} & \overline{R}^{1/2}_k & 0 \\ \overline{A}_k\overline{P}^{1/2}_{k|k-1} & \overline{X}_k & \overline{Q}^{1/2}_{x,k}\end{bmatrix}\Phi_k = \begin{bmatrix}\overline{R}^{1/2}_{e,k} & 0 & 0 \\ \overline{G}_k & \overline{P}^{1/2}_{k+1|k} & 0\end{bmatrix}, \tag{16}$$

where

$$\overline{X}_k = \overline{S}_k\overline{R}_k^{-T/2}$$

$$\overline{Q}_{x,k} = \overline{B}_k\overline{Q}_k\overline{B}_k^T - \overline{S}_k\overline{R}_k^{-1}\overline{S}_k^T.$$

Then, the Kalman filter of type (13) will be given by

$$\overline{K}_k = \overline{G}_k\overline{R}^{-1/2}_{e,k}, \tag{17}$$

and the state EECM is obtained by

$$\overline{P}_{k+1|k} = \overline{P}^{1/2}_{k+1|k}\overline{P}^{T/2}_{k+1|k}. \tag{18}$$

Note that all necessary quantities for updating the estimated state and the state EECM can be recovered from the post-array of (16), so no other additional operations are required once the triangularization is solved.

B. H_∞ estimation problem

In H_∞ estimation problems, no assumptions are made on the statistics and distributions of the stochastic processes involved in the model. The disturbances are only considered to be bounded. The objective of the H_∞ estimation is to find an estimate of $\overline{z}_k$ such that the maximum energy gain from the unknown disturbances $\overline{w}_k$, $\overline{v}_k$ and initial state $\overline{x}_0$ to the estimation error $\tilde{\overline{z}}_{k|k-1}$ is minimized or bounded over the whole time horizon. We can now formulate the H_∞ estimation problem.

Problem 2: (H_∞ Estimation Problem) Consider the state-space model (6) with known measurement sequence $\{\overline{y}_0,\dots,\overline{y}_{k-1}\}$, unknown disturbances $\overline{w}_k$, $\overline{v}_k$ and unknown initial state vector $\overline{x}_0$, with given covariance matrix $\overline{P}_0$. Given a scalar $\gamma \geq 0$, find, if it exists, an estimate of $\overline{z}_k$ that bounds the worst-case energy gain over all possible disturbances of fixed energy, i.e.,

$$\sup_{\overline{x}_0,\{\overline{w}_k\},\{\overline{v}_k\}} \frac{\sum_{j=0}^{k} \tilde{\overline{z}}_{k|k-1}^T \tilde{\overline{z}}_{k|k-1}}{\overline{x}_0^T \overline{P}_0^{-1} \overline{x}_0 + \sum_{j=0}^{k} \overline{w}_j^T \overline{w}_j + \sum_{j=0}^{k} \overline{v}_j^T \overline{v}_j} \leq \gamma^2. \tag{19}$$

An alternative representation of (19) is the following indefinite-quadratic form [11]:

$$J_k = \overline{x}_0^T \overline{P}_0^{-1} \overline{x}_0 + \sum_{j=0}^{k} \overline{w}_j^T \overline{w}_j + \sum_{j=0}^{k} \begin{bmatrix} \overline{y}_j - \overline{C}_j \overline{x}_j \\ \hat{\overline{z}}_{j|j-1} - \overline{L}_j \overline{x}_j \end{bmatrix}^T \begin{bmatrix} I_{N_y} & 0 \\ 0 & -\gamma^2 I_{N_z} \end{bmatrix} \begin{bmatrix} \overline{y}_j - \overline{C}_j \overline{x}_j \\ \hat{\overline{z}}_{j|j-1} - \overline{L}_j \overline{x}_j \end{bmatrix}. \tag{20}$$

Imposing $J_k > 0$ guarantees the existence of the solution to the H_∞ estimation problem, as stated by the following theorem.

Theorem 2: (H_∞ Filter) [11] The solution to Problem 2 is given by an H_∞ filter of level γ, which exists if and only if

$$\overline{R}_k = \begin{bmatrix} I_{N_y} & 0 \\ 0 & -\gamma^2 I_{N_z} \end{bmatrix} \text{ and } \overline{R}_{e,k} = \overline{R}_k + \begin{bmatrix} \overline{C}_k \\ \overline{L}_k \end{bmatrix} \overline{P}_{k|k-1} \begin{bmatrix} \overline{C}_k^T & \overline{L}_k^T \end{bmatrix} \tag{21}$$

have the same inertia, i.e. the same number of positive, negative and zero eigenvalues, for all time instants from 0 to k. The matrix $\overline{P}_{k|k-1}$ satisfies the following Riccati recursion

$$\begin{aligned} \overline{P}_{k+1|k} &= \overline{A}_k \overline{P}_{k-1|k} \overline{A}_k^T + \overline{B}_k \overline{B}_k^T - \overline{A}_k \overline{P}_{k-1|k} \begin{bmatrix} \overline{C}_k^T & \overline{L}_k^T \end{bmatrix} \times \\ &\quad \times \overline{R}_{e,k}^{-1} \begin{bmatrix} \overline{C}_k^T & \overline{L}_k^T \end{bmatrix}^T \overline{P}_{k-1|k} \overline{A}_k^T. \end{aligned} \tag{22}$$

If this is the case, then an H_∞ estimation is given by $\hat{\overline{z}}_{k|k-1} = \overline{L}_k \hat{\overline{x}}_{k|k-1}$, where $\hat{\overline{x}}_{k|k-1}$ denotes the predicted state estimate given by the recursion

$$\hat{\overline{x}}_{k+1|k} = \overline{A}_k \hat{\overline{x}}_{k|k-1} + \overline{K}_k \left(\overline{y}_k - \overline{C}_k \hat{\overline{x}}_{k|k-1} \right),$$

with

$$\overline{K}_k = \overline{A}_k \overline{P}_{k-1|k} \begin{bmatrix} \overline{C}_k^T & \overline{L}_k^T \end{bmatrix} \overline{R}_{e,k}^{-1}. \tag{23}$$

The solution to the H_∞ estimation problem has the same Kalman filter-like structure as the solution as the H_2 estimation problem, except for some differences in the Riccati equations, such as: the presence of an indefinite "covariance" matrix $\overline{R}_k$; the assumption that $\overline{Q}_k$ is an identity matrix and $\overline{S}_k$ is a zero matrix, as they result from the indefinite-quadratic criterion formulation (20); the fact that the matrix $\overline{L}_k$ also enters the Riccati equation; and the additional condition (21) giving information about the existence of a solution. However, the similarities between the two problems have been exploited in [11], where a special kind of indefinite metric space, called a Krein space, along with a special type of projections, called J-unitary transformations, have been introduced and used for a more convenient formulation of the H_∞ estimation problem. This way, previous derivations for some of the main results in Kalman filtering theory have been extended to the H_∞ case, by means of Krein spaces. Among these methods, we focus on the square-root implementation of the Riccati equation and, in the following, we present such square-root array algorithms for the H_∞ estimation problem with SSS matrices.

As the H_∞ estimation problems are formulated in Krein spaces, the orthogonal transformations are replaced here by J-unitary transformations. A transformation Ψ is called J-unitary if $\Psi J \Psi^T = J$, where J is a signature matrix, i.e. a diagonal matrix with only -1 and $+1$ diagonal elements.

For a state-space model of type (6), with unknown and bounded disturbance sequences $\overline{w}_k$ and $\overline{v}_k$, with given initial state $\overline{x}_{0|-1}$ and given square root of its covariance matrix $\overline{P}_{0|-1}$, the state EECM $\overline{P}_{k+1|k}$ of type (22) can be derived by finding a J-unitary transformation Ψ_k, such that, for $k = 0,1,2,\dots$,

$$\begin{bmatrix} \begin{bmatrix} \overline{C}_k \\ \overline{L}_k \end{bmatrix} \overline{P}_{k|k-1}^{1/2} & \overline{R}_k^{1/2} & 0 \\ \overline{A}_k \overline{P}_{k|k-1}^{1/2} & 0 & \overline{B}_k \end{bmatrix} \Psi_k = \begin{bmatrix} \overline{R}_{e,k}^{1/2} & 0 & 0 \\ \overline{G}_k & \overline{P}_{k+1|k}^{1/2} & 0 \end{bmatrix}, \tag{24}$$

where

$$\overline{R}_k^{1/2} = \begin{bmatrix} I_{N_y} & 0 \\ 0 & \gamma I_{N_z} \end{bmatrix} \tag{25}$$

and the signature matrix is

$$J = \begin{bmatrix} J_1 & \\ & I_{N_w} \end{bmatrix}, \text{ with } J_1 = \begin{bmatrix} I_{N_x} & \\ & \begin{bmatrix} I_{N_y} & 0 \\ 0 & -I_{N_z} \end{bmatrix} \end{bmatrix}. \tag{26}$$

Then, both the state EECM of type (22) and the Kalman filter of type (23) can be derived using equations (17)-(18).

IV. Square-root array algorithms with SSS matrices

Both iterations (16) and (24) require a LQ factorization of a particular prearray having a block SSS structure. Therefore, the LQ factorization described in [13] can not be directly applied. In the following, we will show how we can still exploit the SSS structure in the blocks of the prearrays to efficiently compute the postarrays.

Let us consider that the prearray in either (16) or (24) is denoted by $\mathscr{A}$. This matrix can be lifted into a SSS matrix, denoted by $\overline{\mathscr{A}}$, using row and column permutation matrices Π_L and Π_R, respectively:

$$\overline{\mathscr{A}} = \Pi_L \mathscr{A} \Pi_R. \tag{27}$$

The matrix $\overline{\mathscr{A}}$ can now be triangularized using a single SSS structured LQ decomposition, thus

$$\overline{\mathscr{A}}\ \overline{\Phi} = \overline{\mathscr{L}}, \tag{28}$$

where $\overline{\mathscr{L}}$ is an SSS lower triangular matrix and $\overline{\Phi}$ is an SSS orthogonal (or J-unitary) matrix, for H_2 (or H_∞) estimation.

However, we are interested in finding a triangular matrix that has a similar block SSS structure as the matrix $\mathscr{A}$. Thus, using (27), we rewrite (28) in terms of this matrix and we get $\mathscr{A}\ \Pi_R\ \overline{\Phi} = \Pi_L^T \overline{\mathscr{L}}$. As the permutation matrix Π_R is orthogonal, the product $\Pi_R\ \overline{\Phi}$ remains orthogonal (or J-unitary). However, the permutation matrix Π_L^T will interchange the rows of the matrix $\overline{\mathscr{L}}$. This yields to the loss of its lower triangular form, such that the postarray may not be block lower triangular anymore. To guarantee that it will be lower block triangular, our approach in handling the block SSS matrix $\mathscr{A}$ implies first partitioning it into two SSS row blocks and then triangulate the two SSS row blocks, in turns.

We start by only permuting the columns of the matrix $\mathscr{A}$. This is done by right multiplying $\mathscr{A}$ with a permutation matrix Π_R, in order to obtain a matrix with two SSS row blocks, denoted by $\overline{\mathscr{A}}_1$ and $\overline{\mathscr{A}}_2$:

$$\begin{bmatrix} \overline{\mathscr{A}}_1 \\ \overline{\mathscr{A}}_2 \end{bmatrix} = \mathscr{A}\Pi_R. \tag{29}$$

Then, we search for a decomposition of the form

$$\begin{bmatrix} \overline{\mathscr{A}}_1 \\ \overline{\mathscr{A}}_2 \end{bmatrix} \begin{bmatrix} \overline{\Phi}_1 & \overline{\Phi}_2 \end{bmatrix} = \begin{bmatrix} \overline{\mathscr{L}}_{11} & 0 \\ \overline{\mathscr{L}}_{21} & \overline{\mathscr{L}}_{22} \end{bmatrix}, \tag{30}$$

where $\overline{\mathscr{L}}_{11}$ and $\overline{\mathscr{L}}_{22}$ are SSS lower triangular matrices, $\overline{\mathscr{L}}_{21}$ is a SSS matrix and $\begin{bmatrix} \overline{\Phi}_1 & \overline{\Phi}_2 \end{bmatrix}$ is an orthogonal (or J-unitary) block SSS matrix.

Using (29), we can rewrite (30) as follows:

$$\mathscr{A}\ \Pi_R \begin{bmatrix} \overline{\Phi}_1 & \overline{\Phi}_2 \end{bmatrix} = \begin{bmatrix} \overline{\mathscr{L}}_{11} & 0 \\ \overline{\mathscr{L}}_{21} & \overline{\mathscr{L}}_{22} \end{bmatrix}.$$

As $\begin{bmatrix} \overline{\Phi}_1 & \overline{\Phi}_2 \end{bmatrix}$ is orthogonal (or J-unitary), left multiplying it with Π_R preserves orthogonality. Moreover, the lower triangular form of the right term in (30) is also preserved.

The above mentioned triangularization can be easily implemented in a sequence of SSS structure preserving operations, synthesized in the following.

The SSS square-root iteration k:

Step 1. Find an orthogonal (or J-unitary) transformation $\overline{\Phi}_{1,k}$ such that $\overline{\mathscr{A}}_{1,k}\overline{\Phi}_{1,k} = \overline{\mathscr{L}}_{11,k}$.

Step 2. Compute $\overline{\mathscr{L}}_{21,k} = \overline{\mathscr{A}}_{2,k}\overline{\Phi}_{1,k}$.

Step 3. Find an orthogonal transformation $\overline{\Phi}_{2,k}$ such that $\overline{\mathscr{A}}_{2,k}\left(I - \overline{\Phi}_{1,k}\overline{\Phi}_{1,k}^T\right)\overline{\Phi}_{2,k} = \overline{\mathscr{L}}_{22,k}$.

Steps 1 and 3 comprise an SSS structured LQ decomposition each, while Step 2 consists of an SSS matrix multiplication. All operations have linear computational complexity [13], [8].

We can now explicitly present the square-root array algorithms for solving the H_2 and H_∞ estimation problems for SSS structured systems.

Algorithm 1: *H_2 Square-root array algorithm for SSS structured systems.*

Given the matrices $\overline{A}_k$, $\overline{B}_k$, $\overline{C}_k$, $\overline{Q}_k$, $\overline{R}_k$, $\overline{S}_k$. Initialize with:

$$\hat{\overline{x}}_{0|-1} = \mathbf{E}[\overline{x}_0] \text{ and } \overline{P}_{0|-1}^{1/2} = \left(\mathbf{E}[(\overline{x}_0 - \hat{\overline{x}}_{0|-1})(\overline{x}_0 - \hat{\overline{x}}_{0|-1})^T]\right)^{1/2}$$

For $k = 0, 1, 2, \ldots,$

1. Find a permutation matrix Π_R such that
$$\begin{aligned} \overline{\mathscr{A}}_{1,k} &= \begin{bmatrix} \overline{C}_k \overline{P}_{k|k-1}^{1/2} & \overline{R}_k^{1/2} & 0 \end{bmatrix} \Pi_R \text{ and} \\ \overline{\mathscr{A}}_{2,k} &= \begin{bmatrix} \overline{A}_k \overline{P}_{k|k-1}^{1/2} & \overline{X}_k & \overline{Q}_{\times,k}^{1/2} \end{bmatrix} \Pi_R \end{aligned}$$
are SSS matrices.
2. Execute *the SSS square-root iteration* for $\overline{\mathscr{A}}_{1,k}$ and $\overline{\mathscr{A}}_{2,k}$, and get $\overline{R}_{e,k}^{1/2} = \overline{\mathscr{L}}_{11,k}$, $\overline{G}_k = \overline{\mathscr{L}}_{21,k}$, and $\overline{P}_{k+1|k}^{1/2} = \overline{\mathscr{L}}_{22,k}$.

Algorithm 2: *H_∞ Square-root array algorithm for SSS structured systems.*

Given the scalar γ, the matrices $\overline{A}_k$, $\overline{B}_k$, $\overline{C}_k$, $\overline{L}_k$ and $\overline{R}_k^{1/2}$ as in (25). Initialize with:

$$\hat{\overline{x}}_{0|-1} = \mathbf{E}[\overline{x}_0] \text{ and } \overline{P}_{0|-1}^{1/2} = \left(\mathbf{E}[(\overline{x}_0 - \hat{\overline{x}}_{0|-1})(\overline{x}_0 - \hat{\overline{x}}_{0|-1})^T]\right)^{1/2}$$

For $k = 0, 1, 2, \ldots,$

1. Find a permutation matrix Π_R such that
$$\begin{aligned} \overline{\mathscr{A}}_{1,k} &= \begin{bmatrix} \begin{bmatrix} \overline{C}_k \\ \overline{L}_k \end{bmatrix} \overline{P}_{k|k-1}^{1/2} & \overline{R}_k^{1/2} & 0 \end{bmatrix} \Pi_R \text{ and} \\ \overline{\mathscr{A}}_{2,k} &= \begin{bmatrix} \overline{A}_k \overline{P}_{k|k-1}^{1/2} & 0 & \overline{B}_k \end{bmatrix} \Pi_R \end{aligned}$$
are SSS matrices.
2. Execute *the SSS square-root iteration* for $\overline{\mathscr{A}}_{1,k}$ and $\overline{\mathscr{A}}_{2,k}$ with the signature matrix given by the matrix J_1 in (26), and get $\overline{R}_{e,k}^{1/2} = \overline{\mathscr{L}}_{11,k}$, $\overline{G}_k = \overline{\mathscr{L}}_{21,k}$, and $\overline{P}_{k+1|k}^{1/2} = \overline{\mathscr{L}}_{22,k}$.

V. EXPERIMENTAL RESULTS

For our experiments, we used a structured model derived from the discretization of a 1-dimensional spatially heterogeneous wave equation, *cf.* [8]. The generators of the SSS matrices $\overline{A}_k$, $\overline{B}_k$ and $\overline{C}_k$, as shown in (7)-(9), are as follows:

$$A_{s,k} = \begin{bmatrix} 0 & 1 \\ \frac{-5}{2} & 0 \end{bmatrix}, B_{s,k}^q = B_{s,k}^p = \begin{bmatrix} 0 & 0 \\ \frac{2}{3} & \frac{2}{3} \end{bmatrix}, C_{s,k}^q = C_{s,k}^p = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix},$$
$$B_{s,k} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, C_{s,k} = \begin{bmatrix} 1 & 0 \end{bmatrix}, W_{s,k}^q = W_{s,k}^p = \begin{bmatrix} 0 & \frac{-1}{8} \\ 0 & 0 \end{bmatrix},$$

and the other terms are 0, for $s \in \{0, 1, \ldots, N\}$.

Although the derivations of the previous sections were made for the more general case of time varying systems, for simplicity reasons we considered a time invariant approach when running our experiments.

We ran our simulations using MATLAB and a toolbox called ”Sequentially Semi-Separable Matrix Toolbox“ [15].

First, the state EECM is computed for the unstructured model, using the square-root algorithm. We consider this state EECM to give the most accurate estimation of the state. Then, two state EECM's are computed for the SSS structured model; one using the square-root algorithm and one using the Riccati propagation. The relative errors of these two state EECM's with respect to the unstructured state EECM are then calculated. Finally, the accuracy of the two methods is shown by comparing the two relative errors. In order to prevent the effects of the round-off errors, we computed the SSS structured state EECM's in single precision and the unstructured state EECM in double precision. Figure 2 shows the relative error in the computation of the state EECM's for a system with $N = 50$ subsystems, for both H_2 and H_∞ estimation problems. The H_∞ estimators have been obtained using the tolerance $\gamma = 10^{-3}$. As can be seen from Figure 2, the solution obtained with the SSS structured square-root algorithm has much better numerical stability then the the solution obtained using SSS structured Riccati propagations.

In this paper we claim that our approach is less computationally expensive than the classic approach. To test this assertion, we ran experiments where we measured the time taken to compute H_2 and H_∞ estimation, for systems with $N \in \{10,\ 25,\ 50,\ 100,\ 175,\ 250,\ 500\}$ subsystems. As seen in Figure 3, for the high system orders, the Riccati propagation computed in unstructured arithmetic shows an increased computational time when compared to the SSS structured approaches. As expected, the SSS structured square-root algorithm has a linear computational complexity. Although the square-root algorithm shows a slightly increased computing time compared to the Riccati propagation, the use of the square-root algorithm is recommended due to its stability and better accuracy.

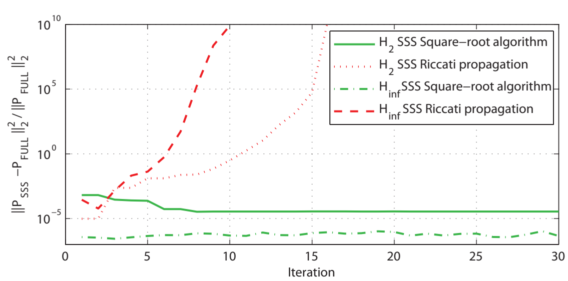

Fig. 2. Accuracy of the state EECM for H_2 and H_∞ estimation problems, computed in SSS arithmetic.

VI. Conclusions

In this paper we have considered the problem of designing H_2 and H_∞ linear estimators for time-varying spatially interconnected systems distributed in one spatial dimension. One of the most difficult aspects of current numerical implementations for algebraic Riccati equation (ARE) solvers lays in the high computational complexity. We based our work on a de-composition technique called "sequentially semi-separable" structures to lower the computational effort. Our

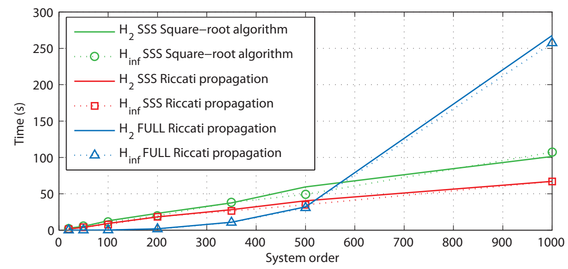

Fig. 3. Computational time comparison between single precision SSS square-root algorithm, single precision SSS Riccati propagation and double precision full-matrix square-root algorithm, for H_2 and H_∞ estimation problems.

novel approach to increase stability of the estimator is to employ a square-root array algorithm. The experimental results confirm that this method is highly promising for large scale systems.

References

[1] K. Yasuda and K. Hirai, "Upper and lower bounds on the solution of the algebraic Riccati equation Automatic Control", *IEEE Transactions on, IEEE*, vol. 24, pp. 483-487, 1979.

[2] F. Wu and S.E. Yildizoglu, "A distributed parameter dependent control design for a flexible beam problem", *American Control Conference, 2002. Proceedings of the 2002*, vol. 4, pp. 3174–3179, 2002.

[3] M. Hogberg, T.R. Bewley and D.S. Henningson, "Relaminarization of Re-tau = 100 turbulence using gain scheduling and linear state-feedback control", *Physics of Fluids*, vol. 15(11), pp. 3572-3575, 2003.

[4] R. Horowitz and P. Varaiya, "Control design of an automated highway system", *Proceedings of the IEEE*, vol. 88(7), pp. 913-925, 2000.

[5] J.M. Fowler and R. D'Andrea, "A formation flight experiment", *Control Systems Magazine, IEEE*, vol. 23(5), pp. 35-43, 2003.

[6] S. Jiang, P.G. Voulgaris, L.E. Holloway and L.A. Thompson, "Distributed control of large segmented telescopes", *American Control Conference, 2006*, pp. 1942-1947, 2006.

[7] R. Fraanje, J. Rice, M. Verhaegen and N. Doelman, "Fast reconstruction and prediction of frozen flow turbulence based on structured Kalman filtering", *J. Opt. Soc. Am. A*, vol. 27(11), pp. A235-A245, 2010.

[8] J. Rice, and M. Verhaegen, "Distributed control: A sequentially semi-separable approach for heterogeneous linear systems" , *IEEE Trans. Automat. Control*, vol. 54, pp. 1270-1283, 2009.

[9] R. Granat, B. Kagstrom and D. Kressner, "A Parallel Schur Method for Solving Continuous-Time Algebraic Riccati Equations", *Computer-Aided Control Systems, 2008. CACSD 2008. IEEE International Conference on*, pp. 583-588, 2008.

[10] J.D. Gardiner, "A Stabilized Matrix Sign Function Algorithm for Solving Algebraic Riccati Equations", *SIAM J. Sci. Comput.*, vol. 18, pp. 1393-1411, 1997.

[11] B. Hassibi, A.H. Sayed and T. Kailath, "Indefinite-quadratic estimation and control: a unified approach to H_2 and H_∞ theories", Society for Industrial and Applied Mathematics, 1999.

[12] 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.

[13] Y. Eidelman, I. Gohberg, V. Olshevsky, "The QR iteration method for Hermitian quasiseparable matrices of an arbitrary order", *Linear Algebra and its Applications*, vol. 404, pp. 305-324, 2005.

[14] J. Rice and M. Verhaegen, "Structure preserving model order reduction of heterogeneous 1-D distributed systems" , *Proceedings of the 2009 conference on American Control Conference*, vol. 54, pp. 4109-4114, 2009.

[15] I. Houtzager. "Sequentially Semi-Separable Matrix Toolbox", http://www.dcsc.tudelft.nl/∼datadriven/sss/ssstoolbox_product_page.html, version 0.6, 2-Feb-2011.