Electronic Transactions on Numerical Analysis.
Volume 38, pp. 113-145, 2011.

ISSN 1068-9613.

ETNA
Kent State University
http://etna.math.kent.edu

TWO EFFICIENT SVD/KRYLOV ALGORITHMS FOR MODEL ORDER REDUCTION OF LARGE SCALE SYSTEMS*

YOUNÈS CHAHLAOUI†

Abstract. We present two efficient algorithms to produce a reduced order model of a time-invariant linear dynamical system by approximate balanced truncation. Attention is focused on the use of the structure and the iterative construction via Krylov subspaces of both controllability and observability matrices to compute low-rank approximations of the Gramians or the Hankel operator. This allows us to take advantage of any sparsity in the system matrices and indeed the cost of our two algorithms is only linear in the system dimension. Both algorithms efficiently produce good low-rank approximations (in the least square sense) of the Cholesky factor of each Gramian and the Hankel operator. The first algorithm computes low-rank approximation of each Gramian independently. The second algorithm works directly on the Hankel operator, and it has the advantage that it is independent of the chosen realization. Moreover, it is also an approximate Hankel norm method. The two reduced order models produced by our methods are guaranteed to be stable and balanced. We study the convergence of our iterative algorithms and the properties of the fixed point iteration. We also discuss the stopping criteria and the choice of the reduced order.

Key words. model order reduction, approximate balanced truncation, Stein equations, Hankel map, Krylov subspaces, approximate Hankel norm method, low-rank approximations

AMS subject classifications. 15A24, 65P99, 93B40, 93C55, 93D99

1. Introduction. Most techniques for model reduction of linear dynamical systems are based on the dominant subspaces of Gramians (energy functions for in- and outgoing signals) or the dominant subspaces of their product [1]. These Gramians are the solutions of Lyapunov equations in the continuous case, or the discrete Lyapunov or Stein equations in the discrete case. Efficiently computing these solutions (or their dominant subspaces) when the system matrices are large and sparse is still a challenging problem; see for instance [4, 5, 6]. In fact, direct methods ignore sparsity in the Lyapunov/Stein equations and are not easy to parallelize. Balanced truncation is one of the most used model reduction methods, and has the desirable property that from a stable model it produces a reduced model that is guaranteed to be stable with a global a priori $\mathcal{H}_\infty$-error bound, but its use is constrained by its complexity. Moreover, balanced truncation is not optimal as it is not minimizing any system norm. A refinement to an optimal approximation method with respect to the Hankel-norm of the system leads to the Hankel-norm approximation [18]. Despite the beauty of the theory it should be stressed that its numerical use is often nontrivial. It is interesting to note that as far as the $\mathcal{H}_\infty$ norm of the error system is concerned (for which we proposed an easy evaluation method in [9]), the Hankel-norm approximation need not provide better results than balanced truncation. The high complexity of balanced truncation is due to the fact that we solve two Lyapunov/Stein equations and then compute a singular value decomposition of the product of these solutions, which both have complexity $O(N^3)$, where N is the dimension of the original system. And so for systems with $N \gtrsim 1000$ the cost of balanced truncation is prohibitively expensive. Even the "square root" version of balanced truncation, where one consider the Cholesky factors of the Gramians instead of the Gramians themselves, has a prohibitive complexity due to the full balancing SVD [1]. However, if the Cholesky factors have low rank the computational cost will be significantly reduced.

*Received January 15, 2010. Accepted for publication November 17, 2010. Published online on April 13, 2011. Recommended by R. Freund. This work was supported by Engineering and Physical Sciences Research Council grant EP/E050441/1.

†Centre for Interdisciplinary Computational and Dynamical Analysis (CICADA), School of Mathematics, The University of Manchester, Manchester, M13 9PL, UK (Younes.Chahlaoui@manchester.ac.uk, http://www.maths.manchester.ac.uk/~chahlaoui/).

Penzl and others [2, 30] have observed that solutions to Lyaponuv/Stein equations associated with linear time-invariant (LTI) systems often have low numerical rank, which means that there is a sharp and early cutoff in the Gramian eigenvalues and by consequence also in the Hankel singular values of the system. Indeed, the idea of low-rank methods is to take advantage of this low-rank structure to obtain approximate solutions in low-rank factored form. The principal outcome of these approaches is that the complexity and the storage are reduced from $O(N^3)$ flops and $O(N^2)$ words of memory to $O(N^2n)$ flops and $O(Nn)$ words of memory, respectively, where n is the "reduced" order and so the "approximate" rank of the Gramians ($n \ll N$). In fact, these low-rank schemes are the only way to solve efficiently very large scale Lyapunov/Stein equations. Moreover, approximating directly the Cholesky factors of the Gramians and using these approximations to provide a reduced model has a comparable cost to that of the popular moment matching methods. It requires only matrix-vector products and linear system solves.

There are many methods to approximate the Gramians of an LTI system. Among the most popular are the Smith method [33], the alternating direction implicit (ADI) iteration method [39], and the Smith(l) method [29]. But all these schemes are computing the solution in dense form, which is prohibitively expensive for large problems. Other methods, such as those in [1, 23, 24, 29, 31, 32], use Krylov subspace ideas and take advantage of any sparsity, but they usually fail to yield approximate solutions of high accuracy. Here we show how to efficiently approximate recursively the Gramians by a low-rank factorization, or equivalently to approximate their Cholesky factors by a low-rank approximation, and at the same time exploit the possible sparsity of the model matrices. We present two efficient iterative methods that can be used for the model reduction of either time varying or time invariant systems. The two reduced order models produced are guaranteed to be stable and balanced. The first method is mainly dedicated to the low-rank approximation of the Gramians, while the second method approximates not only the Gramians but also the Hankel map of the system, which means that it will be independent of the state space realizations of the system. It also provides an approximation to the Hankel-norm model order reduction based methods, which are optimal but very hard to handle. The first key fact about approximate balanced truncation is that we define our reduced order model via its Gramians, from which we construct the projection matrices. The second is that an error bound for the difference between systems can be obtained via the error bound on the difference between their Gramians. In [9] we presented some hints on how to choose the projection matrices in order to have better $\mathcal{H}_\infty$ and $\mathcal{H}_2$ error norms.

This paper is organized as follows. First, in Section 2 we recall some principal notions for linear time-invariant dynamical systems. In Section 3, we present the idea of approximate balanced truncation and we analyze the quality of the reduced order model as a function of the closeness of the projector matrices to those obtained via balanced truncation. Sections 4 and 5 focus on the presentation and discussion of the two new algorithms for the low-rank approximation of the Gramians and the Hankel operator. In Section 4, we present the Recursive Low-Rank Gramian (RLRG) approximation algorithm. It uses the recursive constructibility of the controllability and observability matrices to efficiently produce low-rank approximations of the Cholesky factors of the Gramians. We study the convergence of a fixed point iteration and we give some of its properties. We finish this section by illustrating numerically all these results. In Section 5 the emphasis reverts to the Hankel operator. The Recursive Low-Rank Hankel (RLRH) approximation algorithm is presented. It also uses the recursive constructibility of the controllability and observability matrices, but this time to produce a low-rank approximation of the Hankel operator. This algorithm has the merit that it is independent of the choice of the realization in use. We present some results about approximate balanced truncation based on these two algorithms in Section 6. Both algorithms produce a stable balanced reduced order model. In Section 7, we complete the analysis of our methods by presenting a further discussion about two very important points: the stopping critera and the dynamic choice of the reduced order. The emphasis is on the integration of the second point into our algorithms. We finally illustrate the quality and effectiveness of our methods with some numerical results in the Section 8. We finish with some remarks and open questions in Section 9.

2. Linear time-invariant systems. In this work we concentrate on discrete-time systems, but all our results could be extended to the continuous-time case using the bilinear transformation [3]. The bilinear transformation, also known as Tustin's method, has the advantage that it is a conformal method. In other words, every feature in the continuous system will be preserved in the discretized system; moreover, the Gramians of the continuous system will be the same as for the discretized system. A linear time-invariant system is in general described by the difference equations

$$x_{k+1} = Ax_k + Bu_k, \quad y_k = Cx_k \tag{2.1}$$

with input $u_k \in \mathbb{R}^m$, state $x_k \in \mathbb{R}^N$ and output $y_k \in \mathbb{R}^p$, where $m, p \ll N$, and we assume that the matrices A, B, and C are of appropriate dimensions. We will assume also the system (2.1) to be stable, i.e., all eigenvalues of the matrix A are strictly inside the unit circle. The transfer function associated with the system is defined by $T_f(z) \doteq C(zI - A)^{-1}B$. The Gramians, defined by

$$\mathcal{G}_c = \sum_{i=0}^{\infty} \left(A^i B\right) \left(A^i B\right)^T, \quad \mathcal{G}_o = \sum_{i=0}^{\infty} \left(CA^i\right)^T \left(CA^i\right) \tag{2.2}$$

are solutions of the Stein equations

$$\mathcal{G}_c = A\mathcal{G}_c A^T + BB^T, \quad \mathcal{G}_o = A^T \mathcal{G}_o A + C^T C \tag{2.3}$$

and are also related to the input/output map as follow. Let us at each instant $j \geq k$ restrict inputs to be nonzero (i.e., $u_j = 0, \forall j \geq k$) and consider the outputs from the instant k. The state-to-outputs and inputs-to-state maps are given by

$$\underbrace{\begin{bmatrix} y_k \\ y_{k+1} \\ y_{k+2} \\ \vdots \end{bmatrix}}_{Y} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \end{bmatrix} \underbrace{\begin{bmatrix} B & AB & A^2B & \dots \end{bmatrix} \underbrace{\begin{bmatrix} u_{k-1} \\ u_{k-2} \\ u_{k-3} \\ \vdots \end{bmatrix}}_{U}}_{x(k)}.$$

The Hankel map $\mathcal{H}$ mapping U to Y is $\mathcal{H} = \mathcal{O}\mathcal{C}$, where

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \end{bmatrix}, \quad \mathcal{C} = \begin{bmatrix} B & AB & A^2B & \dots \end{bmatrix}$$

are respectively the observability and the controllability matrices. Notice that this map has rank at most N since $x(k) \in \mathbb{R}^N$, and that $\mathcal{G}_c = \mathcal{C}\mathcal{C}^T$, and $\mathcal{G}_o = \mathcal{O}^T\mathcal{O}$.

ETNA
Kent State University
http://etna.math.kent.edu

In applications, the Gramians can be often well approximated using low-rank approximations. These low-rank approximations are used instead of the original Gramians in the balanced truncation procedure to provide the reduced order model. This is the principle behind the so-called approximate balanced truncation method [28], which has very desirable properties. The combination of Krylov subspace ideas and the balanced truncation procedure implies that approximate balanced truncation inherits the desirable properties of both methods. The iterative computations will reduce significatively the cost (mainly from solving Stein equations) and make use of any sparsity in the data. The use of the balanced truncation procedure yields bounds on the quality of the approximations and a guarantee on the stability of the reduced order system. Next, we investigate this method.

3. Approximate balanced truncation. The balanced truncation procedure is based on the Cholesky factors of the Gramians (2.2) [1]. In practice, these Gramians are low rank matrices (at least numerically), so their Cholesky factors can be well approximated by low-rank approximations.

The Gramians are solutions of Stein equations of the form

$$p(A)\,\mathcal{G}\,p(A)^T - \mathcal{G} = -MM^T, \quad \text{where} \quad p(A) = A \text{ or } A^T, \qquad M = B \text{ or } C^T.$$

These equations appear frequently with a low rank MM^T in engineering applications [1]. This is the case for example whenever $m, p \ll N$. This property implies that the solution $\mathcal{G}$ is a low rank matrix. In theory, this matrix is positive definite whenever $\mathrm{rank}(\mathcal{O}_N) = N$ or $\mathrm{rank}(\mathcal{C}_N) = N$. However, it is often the case that the eigenvalues present a sharp and early cutoff and hence the Gramians are numerically of low rank.

The idea of approximate balanced truncation is to use the low-rank approximations of the Cholesky factors of the Gramians instead of the original Cholesky factors to provide an approximation to balanced truncation. Notice that even if the low-rank approximations were obtained from a discretization of the system, i.e., the discretized Gramians, any low-rank approximation of the discretized Gramian also should be a low-rank approximation of the corresponding continuous-time Gramian since the Gramians are preserved under a bilinear transformation [1, 3]. Moreover, since the continuous and discrete controllability (observability respectively) Gramians are equal, their fundamental subspaces are also equal. This property is used to obtain a reduced model of a continuous-time system whose projection matrices are computed from the bilinear discretized version of this system. The algorithm is as follows.

ALGORITHM 3.1. *Approximate Balanced Truncation (ABT).*

1: ***procedure*** ABT(A, B, C, n, tol)

2: *Use any algorithm to get low-rank approximations* $S, R \in \mathbb{R}^{N\times n}$ *of the Cholesky factors of the Gramians* $\mathcal{G}_c$ *and* $\mathcal{G}_o$*, respectively, such that*

$$\|\mathcal{G}_c - SS^T\| \le tol, \quad \|\mathcal{G}_o - RR^T\| \le tol.$$

3: *Calculate the singular value decomposition* $S^TR = U\Sigma V^T$.

4: *Let* $X = SU\Sigma^{-1/2}$*, and* $Y = RV\Sigma^{-1/2}$.

5: *The order* n *approximate truncated balanced realization is given by*

$$\hat{A} = Y^*AX, \quad \hat{B} = Y^*B, \quad \text{and} \quad \hat{C} = CX.$$

6: ***end procedure***

We use the SVD in Line 3 of the above algorithm to ensure that the projections matrices X and Y are "balanced". This is crucial because we approximate the Gramians independently. In practice, if the system has poles close to the unit circle, then one or both Gramians

ETNA
Kent State University
http://etna.math.kent.edu

are not well approximated. So we use the SVD to balance the error. We obtain a better reduced-order model that is balanced. A similar idea was also proposed by Varga in [38]. He called it *balancing-free square-root method*, and its advantage is that it has a potentially better numerical accuracy for systems that are poorly scaled originally.

Now, let us discuss the effect of the approximation of Gramians on the quality of the obtained reduced model [21]. We consider the nth order reduced system obtained by balanced truncation,

$$\mathcal{S}_{opt} = \left(\begin{array}{c|c} A_{opt} & B_{opt} \\ \hline C_{opt} & 0 \end{array}\right) = \left(\begin{array}{c|c} \pi_l^* A \pi_r & \pi_l^* B \\ \hline C\pi_r & 0 \end{array}\right),$$

where π_l and π_r are the balanced truncation projection matrices [1]. Similarly, let

$$\hat{\mathcal{S}} = \left(\begin{array}{c|c} \hat{A} & \hat{B} \\ \hline \hat{C} & 0 \end{array}\right) = \left(\begin{array}{c|c} Y^*AX & Y^*B \\ \hline CX & 0 \end{array}\right)$$

be the nth order reduced model obtained by an approximate balanced truncation. The following equation is then easily derived:

$$\hat{A}\Sigma\hat{A}^* + \hat{B}\hat{B}^* - \Sigma = Y^*EY - Y^*AEA^*Y,$$

where E is the error in the Gramian $\mathcal{G}_c$, i.e., $E \doteq \mathcal{G}_c - SS^T$, and Σ is a diagonal matrix. The diagonal elements of the matrix Σ are in fact a perturbation of the n Hankel singular values of the system $\hat{\mathcal{S}} = \{\hat{A}, \hat{B}, \hat{C}\}$ and also of the n dominant Hankel singular values of the system $\mathcal{S} = \{A, B, C\}$. This perturbation depends mainly on E. It is clear that the stability of the reduced system is not always guaranteed. However, instability does not seem to occur often in practice (see also [21]); in general we obtain a stable reduced system for each of our computational examples. But notice that one can use the idea of *implicit restart methods* to stabilize the resulting reduced order model if it is unstable [21].

The following result examines how close is the n reduced order model $\mathcal{S}_{opt}$, obtained by balanced truncation, to the n reduced order model $\hat{\mathcal{S}}$ obtained by approximate balanced truncation [21].

THEOREM 3.2. *If* $\|\pi_r - X\| \le \epsilon$, $\|\pi_l - Y\| \le \epsilon$, *then*

$$\|\mathcal{S}_{opt} - \hat{\mathcal{S}}\|_\infty \le \epsilon\left(\|C\|\|B\|\|A\|\left(\|\pi_l\|+\|\pi_r\|\right)+\|\mathcal{S}_1\|_\infty\|B\|+\|\mathcal{S}_2\|_\infty\|C\|\right)+O(\epsilon^2),$$

where $\mathcal{S}_1 \doteq \left(\begin{array}{c|c} A_{opt} & I \\ \hline C_{opt} & 0 \end{array}\right)$, $\quad \mathcal{S}_2 \doteq \left(\begin{array}{c|c} A_{opt} & B_{opt} \\ \hline I & 0 \end{array}\right)$.

Proof. Defining $E_r \doteq \pi_r - X$, and $E_l \doteq \pi_l - Y$, we have $\|E_r\| \le \epsilon$ and $\|E_l\| \le \epsilon$. For $E_A \doteq A_{opt} - \hat{A}$, $E_B \doteq B_{opt} - \hat{B}$, $E_C \doteq C_{opt} - \hat{C}$, we have

$$\begin{aligned} E_A &= \pi_l^* A\pi_r - Y^*AX = \pi_l^* A(\pi_r - X) + (\pi_l - Y)^* AX = \pi_l^* AE_r - E_l^* AX, \\ E_B &= \pi_l^* B - Y^*B = E_l^* B, \quad E_C = C\pi_r - CX = CE_r. \end{aligned}$$

Thus E_A, E_B and E_C satisfy

$$\|E_A\| \le \epsilon\|A\|\left(\|\pi_l\| + \|\pi_r\|\right) + \epsilon^2\|A\|, \quad \|E_B\| \le \epsilon\|B\|, \quad \|E_C\| \le \epsilon\|C\|.$$

We have $(e^{j\omega}I_n - \hat{A})^{-1} \approx (e^{j\omega}I_n - A_{opt})^{-1} + \hat{E}_A$ for every $\omega \in \mathbb{R}$, where $\hat{E}_A = (e^{j\omega}I_n - A_{opt})^{-1}E_A(e^{j\omega}I_n - \hat{A})^{-1}$ satisfies the same upper bound as E_A, i.e.,

$$\|\hat{E}_A\| \le \epsilon\|A\|\left(\|\pi_l\| + \|\pi_r\|\right) + \epsilon^2\|A\|. \tag{3.1}$$

Now, if we consider the $\mathcal{H}_\infty$ norm of the error system $\mathcal{S}_{opt} - \hat{\mathcal{S}}$ we have

$$T_{f\,opt}(e^{j\omega}) - \hat{T}_f(e^{j\omega}) = C_{opt}\left(e^{j\omega}I - A_{opt}\right)^{-1} B_{opt} - \hat{C}\left(e^{j\omega}I - \hat{A}\right)^{-1}\hat{B}.$$

Using (3.1) and the definitions of E_A, E_B and E_C we obtain

$$\begin{aligned}\|\mathcal{S}_{opt} - \hat{\mathcal{S}}\|_{\mathcal{H}_\infty} &= \|C_{opt}T_A B_{opt} - (C_{opt} - E_C)\left[T_A + \hat{E}_A\right](B_{opt} - E_B)\|_2 \\ &= \|E_C T_A B_{opt} + C_{opt}T_A E_B - (C_{opt} - E_C)\hat{E}_A(B_{opt} - E_B)\|_2,\end{aligned}$$

where $T_A = \left(e^{j\omega}I - A_{opt}\right)^{-1}$. Finally, using

$$\mathcal{S}_1 \doteq \left(\begin{array}{c|c} A_{opt} & I \\ \hline C_{opt} & 0 \end{array}\right) \quad \text{and} \quad \mathcal{S}_2 \doteq \left(\begin{array}{c|c} A_{opt} & B_{opt} \\ \hline I & 0 \end{array}\right),$$

it is easy to deduce the final result

$$\|\mathcal{S}_{opt} - \hat{\mathcal{S}}\|_\infty \leq \epsilon\left(\|C\|\|B\|\|A\|\left(\|\pi_l\| + \|\pi_r\|\right) + \|\mathcal{S}_1\|_\infty\|B\| + \|\mathcal{S}_2\|_\infty\|C\|\right) + O(\epsilon^2). \qquad \square$$

Hence for small ϵ, i.e., when X and Y are, respectively, close to π_r and π_l, we expect $\hat{\mathcal{S}}$ to be close to $\mathcal{S}_{opt}$. This result says that the quality of a reduced order model depends on the distance between the projection matrices and those of balanced truncation and the normality of the matrix A. In [21], this result was given informally without proof for the continuous-time case. Here we gave a proof for the discrete-time case, but this may not say much about the quality of approximations if A is far from normal. In that case the norms $\|A\|$, $\|\mathcal{S}_1\|_\infty$ and $\|\mathcal{S}_2\|_\infty$ will be very large and can destroy the sharpness of this bound. In general, the choice of coordinate system for $\hat{A}$, $\hat{B}$ and $\hat{C}$ plays an important role as well. Below, we will show two new methods that propose two possible choices for a good $\hat{\mathcal{S}}$.

Almost all methods proposed for approximate balanced truncation are based on the fact that one obvious way to build a factorization of the Gramian (say, e.g., the controllability Gramian $\mathcal{G}_c$) is iteratively using

$$\mathcal{C}_1 = B, \qquad \mathcal{C}_{i+1} = \left[\begin{array}{cc} \mathcal{C}_i & A^i B \end{array}\right]. \tag{3.2}$$

This is for example the case for all Smith-like methods [1, 8, 21, 27, 29]. But, this factor can also be constructed in two different ways [17]. The formula (3.2) leads to the idea of the modified low-rank Smith algorithm. A second approach is to write it as

$$\begin{aligned}\mathcal{C}_{i+1} &= \left[\begin{array}{cc} B & \left[\begin{array}{cccc} AB & \ldots & A^{i-1}B & A^iB \end{array}\right] \end{array}\right] \\ &= \left[\begin{array}{cc} B & A\left[\begin{array}{cccc} B & \ldots & A^{i-2}B & A^{i-1}B \end{array}\right] \end{array}\right] = \left[\begin{array}{cc} B & A\mathcal{C}_i \end{array}\right].\end{aligned}$$

If one has a good low-rank approximation of $\mathcal{C}_i$ we also will have a good low-rank approximation of $\mathcal{C}_{i+1}$ using this formula. This formulation leads to two new algorithms to compute good low-rank approximations of the Cholesky factor of the Gramians. Both methods are iterative low-rank Gramian methods, and can be included in the *low-rank square Smith method* family. These approaches have the important property that they can be generalized to time-varying systems as well, unlike the other methods. Actually, these approaches have already been used for the time-varying case, and periodic linear systems [8, 12]. In these papers, however, only a result for the time invariant case was presented and no proof or discussion of the convergence was given. Here we shall give a full proof/discussion of the convergence, the fixed points, the quality of the Gramians approximations, and show some attractive properties of the corresponding reduced model.

ETNA
Kent State University
http://etna.math.kent.edu

4. Recursive low-rank Gramian (RLRG) approximation.

As mentioned earlier, in practice the eigenvalues of the Gramians or the eigenvalues of their product present a sharp early cutoff [2, 30], which suggests approximating the Gramians at each step by a low-rank factorization. We show below how to obtain such approximations cheaply and exploit the sparsity of the model $\{A, B, C\}$. The Gramians can be obtained from the Stein iterations

$$\mathcal{G}_c(i+1) = A\mathcal{G}_c(i)A^T + BB^T, \quad \text{and} \quad \mathcal{G}_o(i) = A^T\mathcal{G}_o(i+1)A + C^TC, \tag{4.1}$$

for which the iterates $\mathcal{G}_c(i)$ and $\mathcal{G}_o(i)$ are always symmetric positive semi-definite, so we can substitute them by Cholesky-like factorizations

$$\mathcal{G}_c(i) = \mathcal{C}_i\mathcal{C}_i^T, \quad \text{and} \quad \mathcal{G}_o(i) = \mathcal{O}_i^T\mathcal{O}_i.$$

The key idea of the low-rank method is to approximate the factors $\mathcal{C}_i$ and $\mathcal{O}_i$ by their rank n_i approximations $S(i)$ and $R(i)$, respectively, at each iteration. Typically n_i is constant, i.e., $n_i = n$. We will show, later in this paper (Subsection 7.3), how to let the algorithm choose an appropriate n_i given some user criteria. The algorithm is as follows.

ALGORITHM 4.1. *Recursive low-rank Gramian (RLRG).*

---

procedure RLRG(A, B, C, n, tol)
  $S(0) \leftarrow 0 \in \mathbb{R}^{N\times n}$ ▷ *Initialize S*
  $R(0) \leftarrow 0 \in \mathbb{R}^{N\times n}$ ▷ *Initialize R*
  repeat
    Compute the singular value decompositions
$$\left[\begin{array}{c|c} B & AS(i-1) \end{array}\right] = U_c\Sigma_cV_c^T, \quad \left[\begin{array}{c|c} C^T & A^TR(i-1) \end{array}\right] = U_o\Sigma_oV_o^T.$$
    Let
$$\Sigma_c = \left[\begin{array}{c|c} \Sigma_{c1} & \\ \hline & \Sigma_{c2} \end{array}\right], \quad \Sigma_o = \left[\begin{array}{c|c} \Sigma_{o1} & \\ \hline & \Sigma_{o2} \end{array}\right], \quad \Sigma_{c1}, \Sigma_{o1} \in \mathbb{R}^{n\times n},$$
$$U_c = \left[\begin{array}{c|c} U_{c1} & U_{c2} \end{array}\right], \quad U_o = \left[\begin{array}{c|c} U_{o1} & U_{o2} \end{array}\right], \quad U_{c1}, U_{o1} \in \mathbb{R}^{N\times n}.$$
    Construct
$$S(i) \leftarrow U_{c1}\Sigma_{c1}, \quad R(i) \leftarrow U_{o1}\Sigma_{o1}, \quad E_c(i) \leftarrow U_{c2}\Sigma_{c2}, \quad E_o(i) \leftarrow U_{o2}\Sigma_{o2}.$$
  until *The stopping criterion is verified.* ▷ *See Subsection 7.2*
end procedure

The cost of this algorithm is linear in the largest dimension N. At each iteration, we need to multiply $AS(i)$ and $R(i)^TA$, which requires $4Nn\alpha$ flops, where α is the average number of nonzero elements in each row or column of the sparse matrix A. We need $O(N(n+m)^2)$ flops to form V_c and another $O(N(n+p)^2)$ flops to form V_o [19]. Notice that we have $N \gg n > m, p, \alpha$.

Using the EckartYoung theorem [19], it is immediate from the previous algorithm that

$$\mathcal{P}_i \doteq S(i)S(i)^T, \quad \mathcal{Q}_i \doteq R(i)R(i)^T$$

are the best rank-n approximations to $\mathcal{C}_i\mathcal{C}_i^T$ and $\mathcal{O}_i^T\mathcal{O}_i$, respectively. But this is not sufficient since we want to compare $\mathcal{P}_i$ and $\mathcal{Q}_i$ with $\mathcal{G}_c(i)$ and $\mathcal{G}_c(i)$, respectively. This is analyzed below.

THEOREM 4.2. *At each iteration, there exist unitary matrices $V_c^{(i)} \in \mathbb{R}^{(n+im)\times(n+im)}$, $V_o^{(i)} \in \mathbb{R}^{(n+ip)\times(n+ip)}$, satisfying*

$$\mathcal{C}_iV_c^{(i)} = \left[\begin{array}{c|c|c|c|c} S(i) & E_c(i) & AE_c(i-1) & \dots & A^{i-1}E_c(0) \end{array}\right],$$
$$\mathcal{O}_i^TV_o^{(i)} = \left[\begin{array}{c|c|c|c|c} R(i) & E_o(i) & A^TE_o(i-1) & \dots & (A^{i-1})^TE_c(0) \end{array}\right],$$

ETNA
Kent State University
http://etna.math.kent.edu

where $E_c(i)$ and $E_o(i)$ are the neglected parts at iteration i.

Proof. We just show the proof for $V_c^{(i)}$; that for $V_o^{(i)}$ is similar. At each step, the orthogonal matrix V_c is such that

$$\left[\begin{array}{c|c} B & AS(i-1) \end{array}\right] V_c = \left[\begin{array}{c|c} S(i) & E_c(i) \end{array}\right].$$

For $i = 1$ we have $\mathcal{C}_0 = \left[\begin{array}{c|c} S(0) & E_c(0) \end{array}\right]$. We prove the general result by induction. Suppose that there exists an orthogonal matrix $V_c^{(i)}$, such that

$$\mathcal{C}_i V_c^{(i)} = \left[\begin{array}{c|c|c|c|c} S(i) & E_c(i) & AE_c(i-1) & \dots & A^{i-1}E_c(0) \end{array}\right].$$

Since $\mathcal{C}_{i+1}$ can be obtained from $\mathcal{C}_i$ by $\mathcal{C}_{i+1} = \left[\begin{array}{c|c} B & A\mathcal{C}_i \end{array}\right]$, we choose

$$V_c^{(i)} = \begin{bmatrix} I_m & 0 \\ 0 & V_c^{(i)} \end{bmatrix} \begin{bmatrix} V_c & 0 \\ 0 & I_{(i+1)m} \end{bmatrix},$$

from which it follows that

$$\begin{aligned}
\mathcal{C}_{i+1} V_c^{(i+1)} &= \left[\begin{array}{c|c} B & A\mathcal{C}_i \end{array}\right] \begin{bmatrix} I_m & 0 \\ 0 & V_c^{(i)} \end{bmatrix} \begin{bmatrix} V_c & 0 \\ 0 & I_{(i+1)m} \end{bmatrix} \\
&= \left[\begin{array}{c|c} B & A\mathcal{C}_i V_c^{(i)} \end{array}\right] \begin{bmatrix} V_c & 0 \\ 0 & I_{(i+1)m} \end{bmatrix} \\
&= \left[\begin{array}{c|c|c|c|c} B & AS(i) & AE_c(i) & \dots & A^i E_c(0) \end{array}\right] \begin{bmatrix} V_c & 0 \\ 0 & I_{(i+1)m} \end{bmatrix} \\
&= \left[\begin{array}{c|c|c|c|c} S(i+1) & E_c(i+1) & AE_c(i) & \dots & A^i E_c(0) \end{array}\right]. \qquad \square
\end{aligned}$$

We can use this result to compare $\mathcal{G}_c(i)$ and $\mathcal{G}_o(i)$ with $\mathcal{P}_i$ and $\mathcal{Q}_i$, respectively. Note first that using the previous theorem we have

$$\begin{aligned}
\mathcal{G}_c(i) &= \mathcal{C}_i \mathcal{C}_i^T = \mathcal{C}_i V_c^{(i)} (V_c^{(i)})^T \mathcal{C}_i^T \\
&= \underbrace{S(i)S(i)^T}_{\mathcal{P}_i} + E_c(i)E_c(i)^T + \sum_{j=0}^{i-1} \left(A^{i-j}E_c(j)\right)\left(A^{i-j}E_c(j)\right)^T.
\end{aligned}$$

It follows that

$$\mathcal{G}_c(i) = \mathcal{P}_i + \sum_{j=0}^{i} \left(A^{i-j}E_c(j)\right)\left(A^{i-j}E_c(j)\right)^T. \tag{4.2}$$

Similarly we have

$$\mathcal{G}_o(i) = \mathcal{Q}_i + \sum_{j=0}^{i} \left(E_o(j)A^{i-j}\right)^T \left(E_o(j)A^{i-j}\right). \tag{4.3}$$

As our original system is supposed to be stable, we can bound the differences between $\mathcal{P}_i$ and $\mathcal{G}_c(i)$ and between $\mathcal{Q}_i$ and $\mathcal{G}_o(i)$ for all i,

$$\mathcal{E}_c(i) \doteq \mathcal{G}_c(i) - \mathcal{P}_i, \quad \mathcal{E}_o(i) \doteq \mathcal{G}_o(i) - \mathcal{Q}_i,$$

in terms of the "noise" levels as follows.

THEOREM 4.3. *Let $\mathcal{P}$ and $\mathcal{Q}$ be the solutions of*

$$\mathcal{P} = A\mathcal{P}A^T + I, \quad \mathcal{Q} = A^T\mathcal{Q}A + I.$$

Define the noise levels by $\eta_c = \max_{0\le i\le\infty} \|E_c(i)\|_2, \quad \eta_o = \max_{0\le i\le\infty} \|E_o(i)\|_2$. *Then*

$$\|\mathcal{E}_c(i)\|_2 \le \eta_c^2\|\mathcal{P}\|_2 \le \eta_c^2\frac{\kappa(A)^2}{1-\rho(A)^2}, \quad \|\mathcal{E}_o(i)\|_2 \le \eta_o^2\|\mathcal{Q}\|_2 \le \eta_o^2\frac{\kappa(A)^2}{1-\rho(A)^2}, \tag{4.4}$$

where $\kappa(A) = \|A\|\|A^{-1}\|$ *is the condition number of* A *and* $\rho(A)$ *its spectral radius.*

Proof. Here also we show only the bound for $\mathcal{E}_c(i)$; the second bound can be shown similarly. It follows from (4.2) that

$$\mathcal{E}_c(i+1) = A\mathcal{E}_c(i)A^T + E_c(i)E_c(i)^T.$$

With $\eta_c = \max_{0\le i\le\infty} \|E_c(i)\|_2$, we can consider the equation:

$$\mathcal{X}_{i+1} = A\mathcal{X}_iA^T + (\eta_c^2 I_N - E_c(i)E_c(i)^T), \quad \mathcal{X}_0 = 0.$$

Its iterates $\mathcal{X}_i$ are clearly positive semidefinite and hence converge to a solution $\mathcal{X}$, which is also positive semidefinite. Moreover, by linearity we have

$$\mathcal{E}_c(i+1) + \mathcal{X}_{i+1} = A(\mathcal{E}_c(i) + \mathcal{X}_i)A^T + \eta_c^2 I_N.$$

It then follows that $\lim_{i\to\infty} (\mathcal{E}_c(i) + \mathcal{X}_i) = \eta_c^2\mathcal{P}$, and we obtain $\|\mathcal{E}_c(i)\|_2 \le \eta_c^2\|\mathcal{P}\|_2$. The second bound follows from the eigen-decomposition of A. ∎

We also have the following result on the quality of the approximation of the product of the Gramians.

THEOREM 4.4. *Let $\mathcal{P}$ and $\mathcal{Q}$ be the solutions of*

$$\mathcal{P} = A\mathcal{P}A^T + I, \quad \mathcal{Q} = A^T\mathcal{Q}A + I.$$

Define $\eta_c = \max_{0\le i\le\infty} \|E_c(i)\|_2$, $\eta_o = \max_{0\le i\le\infty} \|E_o(i)\|_2$, *where* $E_c(i)$ *and* $E_o(i)$ *are the neglected parts in Line 7 of Algorithm 4.1. Then*

$$\|\mathcal{G}_c\mathcal{G}_o - \mathcal{P}\mathcal{Q}\|_2 \le \frac{\kappa(A)^2}{1-\rho(A)^2}\left(\eta_c^2\|\mathcal{G}_o\|_2 + \eta_o^2\|\mathcal{G}_c\|_2\right). \tag{4.5}$$

Proof. Consider the identity $\mathcal{G}_c\mathcal{G}_o - \mathcal{P}\mathcal{Q} = (\mathcal{G}_c - \mathcal{P})\mathcal{G}_o + \mathcal{P}(\mathcal{G}_o - \mathcal{Q})$. Taking norms yields

$$\|\mathcal{G}_c\mathcal{G}_o - \mathcal{P}\mathcal{Q}\|_2 \le \|\mathcal{G}_c - \mathcal{P}\|_2\|\mathcal{G}_o\|_2 + \|\mathcal{P}\|_2\|\mathcal{G}_o - \mathcal{Q}\|_2.$$

Finally, using the previous theorem we have

$$\|\mathcal{G}_c - \mathcal{P}\|_2 \le \frac{\eta_c^2\kappa(A)^2}{1-\rho(A)^2}, \quad \|\mathcal{G}_o - \mathcal{Q}\|_2 \le \frac{\eta_o^2\kappa(A)^2}{1-\rho(A)^2},$$

and from the fact that $\|\mathcal{P}\|_2$ is always bounded above by $\|\mathcal{G}_c\|_2$, we obtain by linearity that

$$\|\mathcal{G}_c\mathcal{G}_o - \mathcal{P}\mathcal{Q}\|_2 \le \frac{\kappa(A)^2}{1-\rho(A)^2}\left(\eta_c^2\|\mathcal{G}_o\|_2 + \eta_o^2\|\mathcal{G}_c\|_2\right). \quad ∎$$

This result says that if one Gramian is not well approximated, then the product of the Gramians, which is related to the Hankel singular values (the Hankel singular values are the square roots of the eigenvalues of the product of the Gramians), may not be well approximated.

One should remark that the previous bounds are not explicitly functions of the reduced order n. Both η_c and η_o are functions of n. They will be smaller for a good choice of n or generally for larger n. The term $\kappa(A)^2/(1-\rho(A)^2)$ will be very small when $\rho(A) \ll 1$ and $\kappa(A)$ is reasonable. Moreover, η_c and η_o can be taken equal to the maximum of $\|E_c(i)\|_2$ and $\|E_o(i)\|_2$, respectively, for $k \leq i \leq \infty$, since we can interpret the previous theorems as starting with any initial values. This is particulary useful if after step k the errors have converged to their minimal value, i.e., the convergence threshold ϵ_m. In fact, η_c and η_o are functions of the initial choice and one can write

$$\eta_c(k) = \max_{k \leq i \leq \infty} \|E_c(i)\|_2, \quad \eta_o(k) = \max_{k \leq i \leq \infty} \|E_o(i)\|_2.$$

Since $\eta_c(k)$ and $\eta_o(k)$ are typically decreasing we can replace them by the maximum over the last iteration steps. We will discuss different strategies for the stopping criterion later in Section 7.2.

4.1. Convergence of the RLRG algorithm. In this subsection we analyze the convergence of the recursive low-rank Gramian (RLRG) algorithm for a linear time invariant system $\{A, B, C\}$. The convergence will allow us to deduce important results about the fixed point of the RLRG algorithm. Although all material below applies to both approximations S and R, we focus on the controllability version only, i.e., on S.

First, note that the updating transformation for S is nonlinear and implicit. Thus to prove convergence of the RLRG algorithm, we will use a generalization of the *fixed point theorem*, due to Ortega and Reinboldt [26], called the *contraction mapping theorem*.

DEFINITION 4.5. *A linear operator Υ is nonexpansive if $\rho(\Upsilon) \leq 1$, and contractive if $\rho(\Upsilon) < 1$.*

THEOREM 4.6. *The nonlinear iteration $S_{i+1} = f(S_i), S_i \in \mathbb{R}^{N\times n}$ admits a fixed point S_f if and only if there exists a contractive linear operator ∇f, such that for all S we have*

$$f(S_f + tS) = f(S_f) + t\nabla f S + O(t^2).$$

The operator ∇f is called *Gâteaux-derivative* of f or the Fréchet derivative [22, 26]. For the RLRG algorithm, it is obvious that the differentiability depends on the differentiability of the SVD, which is guaranteed if there is a gap between the part that we keep and the part that we neglect in the algorithm [19], and this is supposed to be the case. To prove the convergence we thus have to prove that the updating mapping is contractive. For this, let us consider a perturbation of S, namely $S + \Delta$. We can define the SVD

$$\left[\begin{array}{c|c} AS & B \end{array}\right] = U \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \\ 0 & 0 \end{bmatrix} V^T, \text{ where } \Sigma_1 \in \mathbb{R}^{n\times n},$$

and using these U and V matrices, we have

$$\left[\begin{array}{c|c} A\Delta & 0 \end{array}\right] = U\hat{\Delta}V^T, \qquad \text{where} \qquad \hat{\Delta} = \begin{bmatrix} \hat{\Delta}_{11} & \hat{\Delta}_{12} \\ \hat{\Delta}_{21} & \hat{\Delta}_{22} \\ \hat{\Delta}_{31} & \hat{\Delta}_{32} \end{bmatrix} \tag{4.6}$$

is partitioned conformably with Σ. Let us consider the partitioned transformations

$$V = \left[\begin{array}{c|c} V_1 & V_2 \end{array}\right] = \left[\begin{array}{c|c} V_{11} & V_{12} \\ \hline V_{21} & V_{22} \end{array}\right], \quad U = \left[\begin{array}{c|c|c} U_1 & U_2 & U_3 \end{array}\right], \tag{4.7}$$

and define $\tilde{\Sigma}_1 \doteq \Sigma_1 + \Delta_{11}$ and $\tilde{\Sigma}_2 \doteq \Sigma_2 + \Delta_{22}$. To analyze the fixed point iteration we can distinguish two cases: V constant and V varying. If V is constant then the new version S_{i+1} is given by

$$\mathcal{S}_{i+1} = \left[\begin{array}{c|c} A\mathcal{S}_i & B \end{array}\right] V_1 = U \begin{bmatrix} \Sigma_1 \\ 0 \\ 0 \end{bmatrix},$$

and the perturbed version of S_{i+1} is given by

$$\mathcal{S}_{i+1} + \Delta_1 = \left[\begin{array}{c|c} A(\mathcal{S}_i + \Delta) & B \end{array}\right] V_1 = U \begin{bmatrix} \tilde{\Sigma}_1 \\ \hat{\Delta}_{21} \\ \hat{\Delta}_{31} \end{bmatrix},$$

and thus $\Delta_1 = \left[\begin{array}{c|c} A\Delta & 0 \end{array}\right] V_1 = A\Delta V_{11}$. Using the vec formulation we obtain

$$\mathrm{vec}(\Delta_1) = (V_{11}^T \otimes A)\mathrm{vec}(\Delta).$$

Here, the term $V_{11}^T \otimes A$ corresponds to the linear operator ∇f of the last theorem. As $\rho(V_{11}^T \otimes A) = \rho(V_{11})\rho(A) < 1$ ($\rho(V_{11}) \leq 1$ because V_{11} is a submatrix of the orthogonal matrix V) the mapping $\Delta \to \Delta_1$ is a contraction.

Let now V be varying as well. The new iterates S_{i+1} is still given by

$$\mathcal{S}_{i+1} = \left[\begin{array}{c|c} A\mathcal{S}_i & B \end{array}\right] V_1 = U_1\Sigma_1,$$

and the perturbed version is given by

$$\mathcal{S}_{i+1} + \Delta_1 = \left[\begin{array}{c|c} A(\mathcal{S}_i + \Delta) & B \end{array}\right] V_1(\Delta) = U_1(\Delta)\hat{\Sigma}_1,$$

and so $\Delta_1 = U_1(\Delta)\hat{\Sigma}_1 - U_1\Sigma_1$. If we write the transformation $U(\Delta)$ as

$$U(\Delta) = U \begin{bmatrix} I & -Q^T \\ Q & I \end{bmatrix} + O(\|\Delta\|_2^2),$$

then a first order approximation to Q can be obtained from [34, p.359], [35, p.206], [36]

$$K \begin{bmatrix} \tilde{\Sigma}_1\tilde{\Sigma}_1^T & \tilde{\Sigma}_1\hat{\Delta}_{21}^T + \hat{\Delta}_{12}\tilde{\Sigma}_2 & \tilde{\Sigma}_1\hat{\Delta}_{31}^T \\ \hat{\Delta}_{21}\tilde{\Sigma}_1 + \tilde{\Sigma}_-\hat{\Delta}_{12}^T & \tilde{\Sigma}_2\tilde{\Sigma}_2^T & \Sigma_2\hat{\Delta}_{32}^T \\ \hat{\Delta}_{31}\tilde{\Sigma}_1 & \hat{\Delta}_{32}\tilde{\Sigma}_2 & 0 \end{bmatrix} K + O(\|\Delta\|_2^2) = \begin{bmatrix} \hat{\Sigma}_1^2 & 0 & 0 \\ 0 & \hat{\Sigma}_2^2 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

where $K = \begin{bmatrix} I & Q^T \\ -Q & I \end{bmatrix}$. Now, if we consider the $(2:3,1)$ blocks, we have

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = -Q(\tilde{\Sigma}_1\tilde{\Sigma}_1^T) + \begin{bmatrix} \tilde{\Sigma}_2\tilde{\Sigma}_2^T & \tilde{\Sigma}_2\hat{\Delta}_{32}^T \\ \hat{\Delta}_{32}\tilde{\Sigma}_2 & 0 \end{bmatrix} Q + \begin{bmatrix} \hat{\Delta}_{21}\tilde{\Sigma}_1 + \tilde{\Sigma}_2\hat{\Delta}_{12}^T \\ \hat{\Delta}_{31}\tilde{\Sigma}_1 \end{bmatrix} + O(\|\Delta\|_2^2).$$

This equation can be solved to first order [34, p.359], [35, p.206], and if we neglect Σ_2 versus Σ_1 (i.e., $\|\Sigma_1^{-1}\|_2\|\Sigma_2\|_2 \simeq O(\|\Delta\|_2)$)[1], we obtain

$$\|Q - \begin{bmatrix} \hat{\Delta}_{21}\tilde{\Sigma}_1^{-1} \\ \hat{\Delta}_{31}\tilde{\Sigma}_1^{-1} \end{bmatrix}\|_2 \le \|Q\|_2 \frac{\|\tilde{\Sigma}_1^{-1}\|_2^2\|\tilde{\Sigma}_2\|_2^2}{1-\|\tilde{\Sigma}_1^{-1}\|_2^2\|\tilde{\Sigma}_2\|_2^2}. \tag{4.8}$$

And thus one obtains

$$\begin{aligned}\Delta_1 &= U\begin{bmatrix} I \\ \hat{\Delta}_{21}\tilde{\Sigma}_1^{-1} \\ \hat{\Delta}_{31}\tilde{\Sigma}_1^{-1} \end{bmatrix}\tilde{\Sigma}_1 - U_1\Sigma_1 + O(c) \\ &= U_1(\Sigma_1 + \hat{\Delta}_{11}) + U_2\hat{\Delta}_{21} + U_3\hat{\Delta}_{31} - U_1\Sigma_1 + O(c) \\ &= U_1\hat{\Delta}_{11} + U_2\hat{\Delta}_{21} + U_3\hat{\Delta}_{31} + O(c),\end{aligned}$$

where

$$c = \|Q\|_2 \frac{\|\tilde{\Sigma}_1^{-1}\|_2^2\|\tilde{\Sigma}_2\|_2^2}{1-\|\tilde{\Sigma}_1^{-1}\|_2^2\|\tilde{\Sigma}_2\|_2^2}.$$

From (4.6) we have

$$\begin{bmatrix} \hat{\Delta}_{11} \\ \hat{\Delta}_{21} \\ \hat{\Delta}_{31} \end{bmatrix} = \begin{bmatrix} U_1^T \\ U_2^T \\ U_3^T \end{bmatrix} \left[\begin{array}{c|c} A\Delta & 0 \end{array}\right] V_1 \tag{4.9}$$

so

$$\begin{aligned}\Delta_1 &= U_1U_1^T A\Delta V_{11} + U_2U_2^T A\Delta V_{11} + U_3U_3^T A\Delta V_{11} + O(c) \\ &= \underbrace{(U_1U_1^T + U_2U_2^T + U_3U_3^T)}_{I} A\Delta V_{11} + O(c).\end{aligned}$$

Therefore we have $\Delta_1 \simeq A\Delta V_{11} + O(c)$. Furthermore from (4.8) and (4.9) we have $\|Q\|_2 \approx \|A\Delta\|_2\|\Sigma_1^{-1}\|_2$, and so

$$c = \|Q\|_2 \frac{\|\tilde{\Sigma}_1^{-1}\|_2^2\|\tilde{\Sigma}_2\|_2^2}{1-\|\tilde{\Sigma}_1^{-1}\|_2^2\|\tilde{\Sigma}_2\|_2^2} \approx \|A\Delta\|_2 \frac{\|\tilde{\Sigma}_1^{-1}\|_2^3\|\tilde{\Sigma}_2\|_2^2}{1-\|\tilde{\Sigma}_1^{-1}\|_2^2\|\tilde{\Sigma}_2\|_2^2}.$$

Using the vec formulation we obtain finally that $\mathrm{vec}(\Delta_1) = (V_{11}^T \otimes A)\mathrm{vec}(\Delta) + O(c)$. Since $\rho(V_{11}^T \otimes A) = \rho(V_{11})\rho(A) < 1$, the mapping $\Delta \to \Delta_1$ is a contraction provided that $\|\tilde{\Sigma}_1^{-1}\|_2\|\tilde{\Sigma}_2\|_2$ is sufficiently small, i.e., the gap is sufficiently large. Under these conditions, the RLRG algorithm admits a fixed point. Furthermore, this fixed point has a very desirable property given by the theorem below. First we introduce the $\{A, B\}$-invariance.

DEFINITION 4.7. *A subspace $\mathcal{V}$ of $\mathbb{R}^N$ is said to be an $\{A, B\}$-invariant subspace if $\mathcal{V}$ is invariant under A and contains the image space of B (denoted by $\mathrm{Im}B$). We let $\mathcal{V} = \Gamma_A \mathrm{Im}B$.*

We have the following equivalences.

[1]Note that in this case

$$\|(\tilde{\Sigma}_1\tilde{\Sigma}_1^T)^{-1}O(\|\Delta\|_2^2)\| \ll O(\|\Delta\|_2^2)$$

LEMMA 4.8. [25] *For all* $\mathcal{V} = \Gamma_A \mathrm{Im} B$*, we have*

$$A\mathcal{V} \subset \mathrm{Im}B + \mathcal{V} \quad \Leftrightarrow \quad (A - BK)\mathcal{V} \subset \mathcal{V}.$$

THEOREM 4.9. *The fixed point of the RLRG algorithm is an* $\{A, B\}$*-invariant subspace provided that the matrix* V_{11} *in* (4.7) *is nonsingular.*

Proof. Let i be the iteration where we reach the fixed point, i.e., $\mathrm{Im}S(i) = \mathrm{Im}S(i+1)$, which is equivalent to say that there exists a square nonsingular matrix X, such that $S(i)X = S(i+1)$. Then, if we put ourselves in a coordinate system, where

$$S(i) = \begin{bmatrix} R \\ 0 \end{bmatrix}, \quad R \in \mathbb{R}^{n\times n},$$

(this can be obtained using for example a QR decomposition of $S(i)$ followed by a pre-multiplication of the matrix $S(i)$ by Q). The fixed singular subspace implies that we must have

$$S(i+1) = \begin{bmatrix} \hat{R} \\ 0 \end{bmatrix}, \quad \hat{R} \in \mathbb{R}^{n\times n}.$$

The two matrices R and $\hat{R}$ are related using (4.7) as follows

$$\left[\; \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] \left[\begin{array}{c} R \\ \hline 0 \end{array}\right] \;\middle|\; \left[\begin{array}{c} B_1 \\ \hline B_2 \end{array}\right] \;\right] \left[\begin{array}{c} V_{11} \\ \hline V_{21} \end{array}\right] = \left[\begin{array}{c|c} A_{11}R & B_1 \\ \hline A_{21}R & B_2 \end{array}\right] \left[\begin{array}{c} V_{11} \\ \hline V_{21} \end{array}\right] = \left[\begin{array}{c} \hat{R} \\ \hline 0 \end{array}\right].$$

And so, we have

$$A_{11}RV_{11} + B_1V_{21} = \hat{R}, \quad A_{21}RV_{11} + B_2V_{21} = 0. \tag{4.10}$$

If V_{11} is invertible it follows that $\mathrm{Im}S(i) = \begin{bmatrix} I \\ 0 \end{bmatrix}$ must be an $\{A, B\}$-invariant subspace since for $K = \left[\begin{array}{c|c} K_1 & 0 \end{array}\right] = \left[\begin{array}{c|c} -V_{21}V_{11}^{-1}R^{-1} & 0 \end{array}\right]$, we have

$$A - BK = \left[\begin{array}{c|c} A_{11} - B_1K_1 & A_{12} \\ \hline A_{211} - B_2K_1 & A_{22} \end{array}\right] = \left[\begin{array}{c|c} A_{11} - B_1K_1 & A_{12} \\ \hline 0 & A_{22} \end{array}\right],$$

which concludes our proof. □

For the observability, we speak about $\{A^T, C^T\}$ invariance instead of $\{A, B\}$ invariance. Moreover, we have the following corollary of Theorem 4.2.

COROLLARY 4.10. *At each iteration, there exists an orthogonal matrix* $V^{(i)} \in \mathbb{R}^{(n+im)\times n}$*, satisfying* $\mathcal{C}_iV^{(i)} = S(i)$*.*

Proof. For $i = 0$ we have $\mathcal{C}_0 \begin{bmatrix} I_n \\ 0 \end{bmatrix} = S(0)$. We prove the general result by induction. Suppose that there exists an orthogonal matrix $V^{(i)}$, such that $\mathcal{C}_iV^{(i)} = S(i)$. Since $\mathcal{C}_{i+1}$ and $S(i+1)$ can be obtained from $\mathcal{C}_i$ and $S(i)$ (Theorem 4.2 and its proof), respectively, as

$$\mathcal{C}_{i+1} = \left[\begin{array}{c|c} B & A\mathcal{C}_i \end{array}\right] \quad \text{and} \quad S(i+1) = \left[\begin{array}{c|c} B & AS(i) \end{array}\right] V_c^+, \quad \text{where} \quad V_c^+ = V_c(:, 1:n),$$

it follows that

$$\begin{aligned} S(i+1) &= \left[\begin{array}{c|c} B & AS(i) \end{array}\right] V_c^+ = \left[\begin{array}{c|c} B & A\mathcal{C}_iV^{(i)} \end{array}\right] V_c^+ \\ &= \left[\begin{array}{c|c} B & A\mathcal{C}_i \end{array}\right] \begin{bmatrix} I_m & 0 \\ 0 & V^{(i)} \end{bmatrix} V_c^+ = \mathcal{C}_{i+1}V^{(i+1)}, \end{aligned} \tag{4.11}$$

ETNA
Kent State University
http://etna.math.kent.edu

where $V^{(i+1)} = \begin{bmatrix} I_m & 0 \\ 0 & V^{(i)} \end{bmatrix} V_c^+$. ∎

Now we can characterize the fixed point.

THEOREM 4.11. *The RLRG algorithm has as a fixed point $S = S(\infty) = U\Sigma^{\frac{1}{2}}$, where the columns of U are the n dominant eigenvectors of the corresponding Gramian (also singular vectors, as the Gramian is a Hermitian positive semidefinite matrix) and Σ is a diagonal matrix of the corresponding singular values of the Gramian.*

Proof. We show the proof only for the controllability case, the other case being similar. Let $\sigma_j^{(i)}$, $j = 1, \ldots, n$, be the n first singular values of $\mathcal{C}_i$ and $\widehat{\sigma}_j^{(i)}$, $j = 1, \ldots, n$, those of $S(i)$. We have $\mathcal{C}_{i+1} = \left[\begin{array}{c|c} B & A\mathcal{C}_i \end{array}\right] = \left[\begin{array}{c|c} \mathcal{C}_i & A^iB \end{array}\right]$, which means that $\mathcal{C}_i$ is a submatrix of $\mathcal{C}_{i+1}$, and so

$$\sigma_j^{(i)} \leq \sigma_j^{(i+1)}, \quad j = 1, \ldots, n.$$

Then according to Theorem 4.2, there exists a unitary matrix $V^{(i)} \in R^{(n+im)\times(n+im)}$, such that

$$\mathcal{C}_i V^{(i)} = \left[\begin{array}{c|c} S(i) & \mathcal{E}(i) \end{array}\right], \quad \text{where} \quad \mathcal{E}(i) = \left[\begin{array}{c|c|c|c} E(i) & AE(i-1) & \ldots & A^{(i-1)}E(0) \end{array}\right],$$

and $E(j)$ the neglected part of $\left[\begin{array}{c|c} AS(j-1) & B \end{array}\right]$ at the iteration j. Then using the relation $\mathcal{C}_{i+1} = \left[\begin{array}{c|c} \mathcal{C}_i & A^iB \end{array}\right]$ we can write

$$\left[\begin{array}{c|c} S(i+1) & \mathcal{E}(i+1) \end{array}\right] {V^{(i+1)}}^T = \left[\begin{array}{c|c} \left[\begin{array}{c|c} S(i) & \mathcal{E}(i) \end{array}\right] {V^{(i)}}^T & A^iB \end{array}\right].$$

We can see easily that

$$\sigma_j\left(\left[\begin{array}{c|c} S(i+1) & \mathcal{E}(i+1) \end{array}\right] {V^{(i+1)}}^T\right) \geq \sigma_j\left(\left[\begin{array}{c|c} S(i) & \mathcal{E}(i) \end{array}\right] {V^{(i)}}^T\right),$$

and as $V^{(i)}$ are unitary matrices, we have

$$\sigma_j\left(\left[\begin{array}{c|c} S(i+1) & \mathcal{E}(i+1) \end{array}\right]\right) \geq \sigma_j\left(\left[\begin{array}{c|c} S(i) & \mathcal{E}(i) \end{array}\right]\right),$$

and finally, by construction $S(i)$ is the dominant part of $\left[\begin{array}{c|c} S(i) & \mathcal{E}(i) \end{array}\right]$ then

$$\sigma_j(S(i+1)) \geq \sigma_j(S(i)).$$

The n singular values of $S(i)$ are nondecreasing from one iteration to another, and as we have shown before that the fixed point is $\{A, B\}$-invariant, the space spanned by the columns of $S(i)$ converges to a maximal (in term of these singular values) subspace of dimension n. This maximal subspace is known as the n-maximal $\{A, B\}$ invariant subspace (see [25] for more details), and can proved to be the rank-n dominant approximation of the controllability matrix $\mathcal{C} \doteq \mathcal{C}_\infty$ and so of the controllability Gramian $\mathcal{G}_c = \mathcal{G}_c(\infty)$.

Formally, the RLRG algorithm is based on the fact that $\mathcal{C}_{i+1} = \left[\begin{array}{c|c} B & A\mathcal{C}_i \end{array}\right]$. Taking the limit when $i \longrightarrow \infty$ in both sides we get $\mathcal{C}_\infty = \left[\begin{array}{c|c} B & A\mathcal{C}_\infty \end{array}\right]$, so the n dominant left singular vectors (called also the n left fundamental subspace [34]) of $\mathcal{C} = \mathcal{C}_\infty$ are the corresponding fixed point. All this discussion leads to the conclusion that the RLRG algorithm has one fixed point corresponding to the n dominant singular subspace of the corresponding Gramian. ∎

Actually, we have a double convergence: one for the singular values and the other for the subspace. Recall that $S(i) = U_c(i)\Sigma_c(i)$, where $U_c(i)$ are the n dominant left singular

ETNA
Kent State University
http://etna.math.kent.edu

vectors of $\left[\begin{array}{c|c} AS(i-1) & B \end{array}\right]$ and $\Sigma_c(i)$ contains the n corresponding singular values; see Algorithm 4.1.

Numerically, the convergence for the subspace should be checked by computing the canonical angle [7] (or its cosine) between $U_c(i)$ and the dominant subspace of dimension n of the controllability Gramian $\mathcal{G}_c$ ($\measuredangle\,(U_c(i), \mathcal{G}_c)$). But, as the Gramian is not available we can check this convergence using the canonical angle between $U_c(i)$ and $U_c(i-1)$ ($\measuredangle(U_c(i), U_c(i-1))$). This convergence occurs very quickly as soon as $\mathrm{Im}B$ is enclosed in the subspace, then the algorithm takes a few iterations to reach the fixed point (for the subspace). The convergence rate of this iteration seems to be a function only of the number $\min(m, n)$ (respectively, $\min(p, n)$ for the observability Gramian) and not a function of the size of A or its spectral radius. On the other hand, the convergence for the singular values is mainly a function of the spectral radius of A.

The previous theorem has an important hidden outcome. It provides the link with Krylov subspace methods. We have $\mathrm{Im}\ \mathcal{C} = \mathcal{K}_\infty(A, B)$. So the n fundamental left subspace of $\mathcal{C}$, which is the fixed point iterations of the RLRG algorithm, is also the n dominant subspace of $\mathcal{K}_\infty(A, B)$. This is the reason why approximated balanced truncation is called a SVD/Krylov method.

4.2. Numerical illustration. We illustrate all this discussion using the following numerical example. We generate five random stable systems $\{A_i, B, C\}$ of order $N = 400$, with $m = 6$ inputs, $p = 4$ outputs (we keep the same B and C for all five systems), and the spectral radii $\rho(A_1) = 0.95$, $\rho(A_2) = 0.9$, $\rho(A_3) = 0.8$, $\rho(A_4) = 0.6$, $\rho(A_5) = 0.4$. We take $n = 30$. In the first two figures, we show the canonical angle between $U_c(i)$ and $U_c(i-1)$ ($\measuredangle(U_c(i), U_c(i-1))$) (Figure 4.1), and the canonical angle between $U_c(i)$ and the dominant subspace of dimension n of the controllability Gramian $\mathcal{G}_c$ ($\measuredangle\,(U_c(i), \mathcal{G}_c)$) (Figure 4.2). Figure 4.1 shows that there is a fixed point iteration, and Figure 4.2 shows that this fixed point is the dominant subspace of dimension n of the controllability Gramian $\mathcal{G}_c$. One should notice that we would like to avoid computing the exact dominant subspace of dimension n of the controllability Gramian $\mathcal{G}_c$ as it is expensive.

From these figures, it is very easy to see the effect of the spectral radius $\rho(A)$ on the convergence rate. The smaller the spectral radius the faster the convergence to the fixed subspace, but at the end, in general the quality of the approximation, measured by the canonical angle between subspaces, is of the same order. In Figure 4.2, we verify that the fixed subspace is effectively the dominant subspace of the Gramian of dimension n. In Figure 4.3, we can see that after a few iterations the noise level is converging also to a constant value, which is also a function of the spectral radius of A, i.e., the smaller the spectral radius is, the smaller is the noise level. Actually, we could use this convergence in the noise level to restart the algorithm in order to get a good approximation. This takes in general very few iterations. The convergence of the singular values is considered in the last two figures. Figure 4.5 shows the number of the Gramian singular values matched at each iteration, and in Figure 4.4 the corresponding distance between the two sets of the singular values. Here also the convergence is a function of the spectral radius of A. But the effect is more evident, and the slope is more significant as this spectral radius become smaller. In general, these numerical results confirm our previous results about the relationship between the spectral radius of A and the convergence rate and the quality of the approximation. When $\rho(A)$ is close to 1 (but still smaller than 1), we need many more iterations to get the same quality than when $\rho(A)$ is much smaller than 1. This will be very useful; if one has a continuous system on hand, we could choose a bilinear discretization in order to get the spectral radius of the resulting matrix A much smaller than 1 in order to get a very fast good approximation.

This convergence result will allow us later to deduce some useful properties of the re-

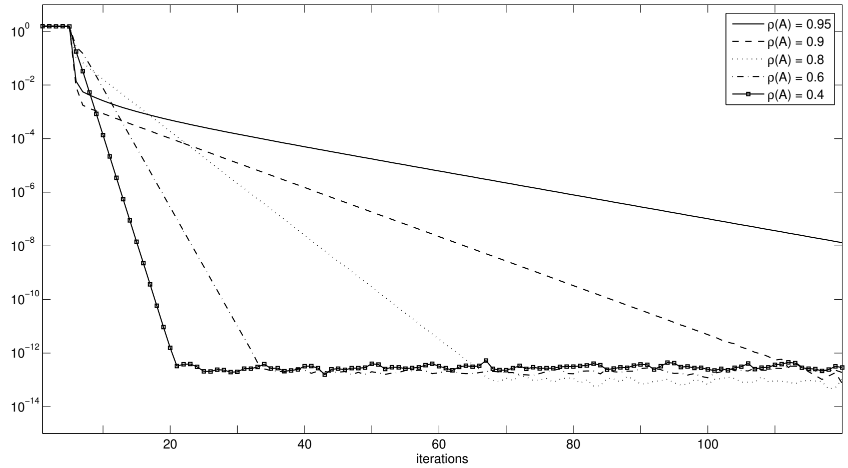

FIG. 4.1. $\measuredangle\,(U_c(i), U_c(i-1))$.

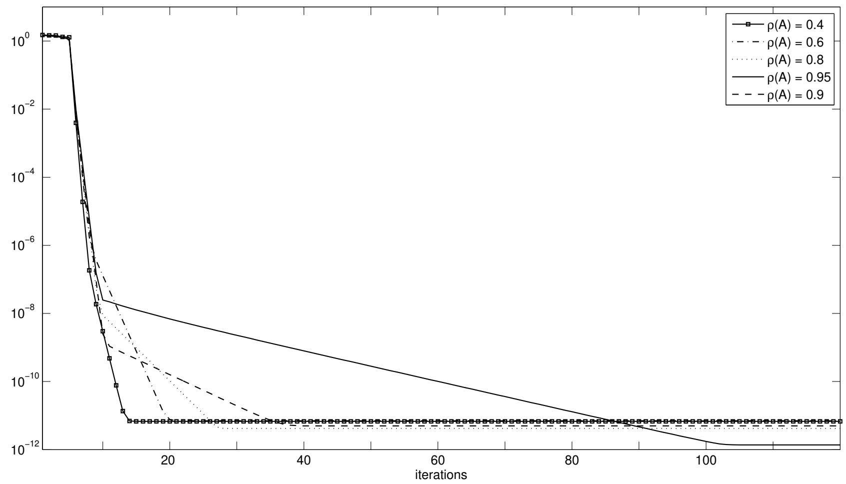

FIG. 4.2. $\measuredangle\,(U_c(i), \mathcal{G}_c)$.

duced model, especially about the stability and balancing.

Unfortunately, the RLRG algorithm produces an independent approximation of the two Gramians. So to obtain a reduced model we have to "balance" the projection matrices obtained from these two approximations. The quality of the approximation and indeed of the reduced model depends on the two "noise" level parameters μ_c and μ_o, which determine if the two Gramians are well approximated or not. These parameters are independent as we approximate Gramians independently from one another, and so one can imagine the case where one Gramian is well approximated and the other one not. So, this affects the quality of the approximation of the reduced model. For instance, if a bilinear transformation T is applied to the system $\{A, B, C\}$ to get a new system $\{T^{-1}AT, T^{-1}B, CT\}$, the corresponding controllability and observability matrices and Gramians, respectively, will be

$$\tilde{\mathcal{C}} = T^{-1}\mathcal{C}, \quad \hat{\mathcal{O}} = \mathcal{O}T, \quad \hat{\mathcal{G}}_c = T^{-1}\mathcal{G}_c T^{-T}, \quad \hat{\mathcal{G}}_o = T^T \mathcal{G}_o T.$$

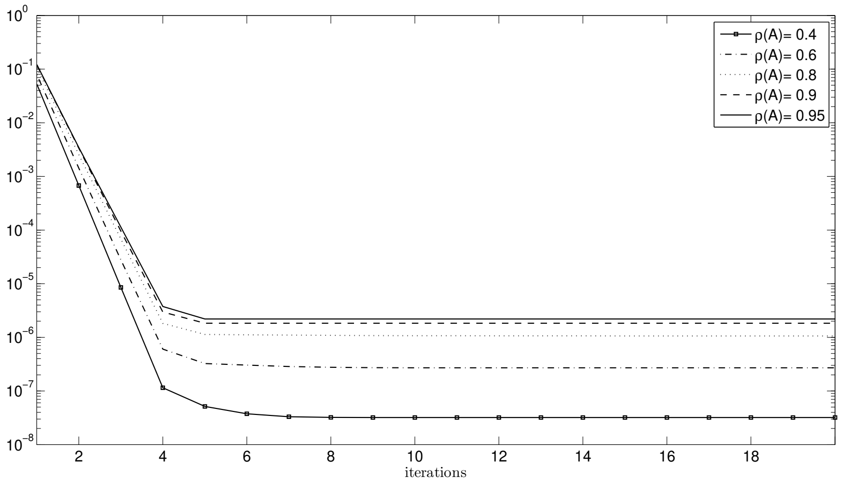

FIG. 4.3. *Noise level* η

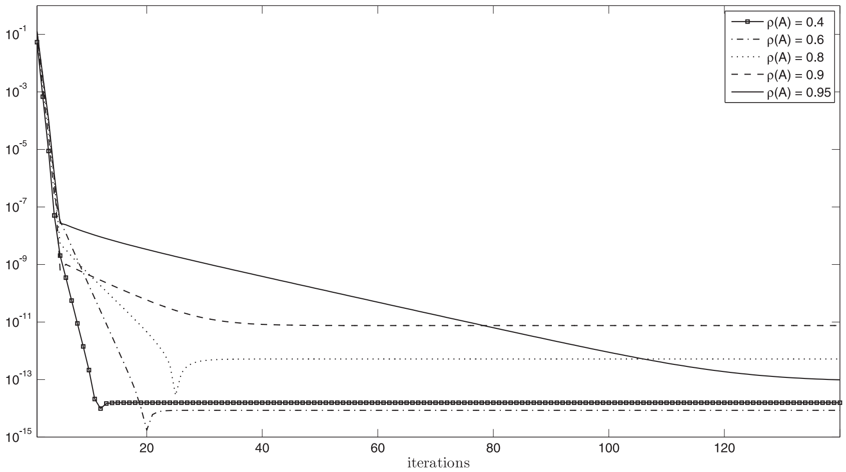

FIG. 4.4. $\max_i |\sigma_i^2(S(i)) - \sigma_i(\mathcal{G}_c)|$.

This transformation will also affect the product of the Gramians (which is taken into account for the balancing) as follows $\hat{\mathcal{G}}_c\hat{\mathcal{G}}_o = T^{-1}\mathcal{G}_c\mathcal{G}_oT$. We can see very easily that to have good approximations of the Gramians, one has to choose good realizations of the system, which means the choice of the matrices A, B, and C. This is not obvious, and could lead to a very bad result. In the following section we present an algorithm which avoids this problem.

5. Recursive low-rank Hankel (RLRH) approximation. The key idea of this approach is to use the underlying recurrences defining the so-called Hankel matrices. Because the system order at each instant is given by the rank of the Hankel matrix at that instant, one can approximate the system by approximating the Hankel matrix. This is the idea of the exact Hankel norm approximation methods [17]. In this case, the norm approximation problem is

$$\min_{\operatorname{rank}\hat{\mathcal{H}} \leq n} \|\mathcal{H} - \hat{\mathcal{H}}\|, \tag{5.1}$$

ETNA
Kent State University
http://etna.math.kent.edu

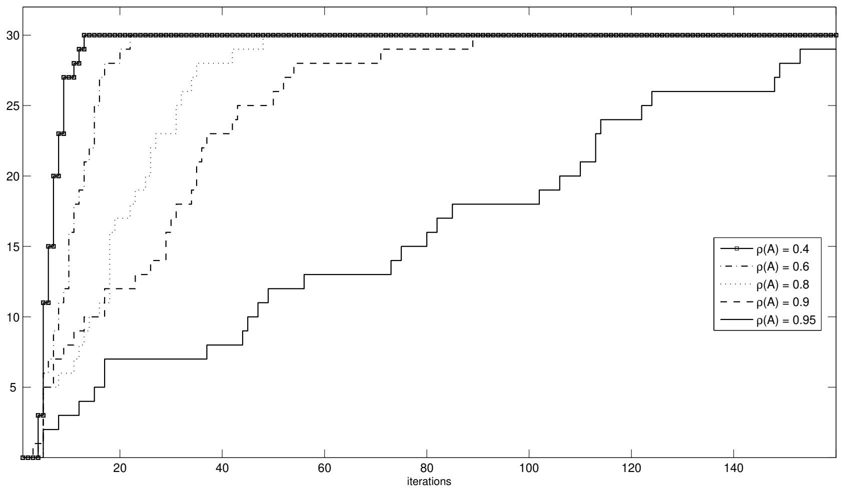

FIG. 4.5. *Number of $\sigma_i(\mathcal{G}_c)$ matched at each iteration.*

where $\mathcal{H}$ is the Hankel map which makes the correspondence between inputs and outputs; see Section 2. The problem (5.1) has many solutions, since only the largest singular values of the difference $E = \mathcal{H} - \hat{\mathcal{H}}$ is minimized, and $n-1$ others are free as long as they remain smaller. In general, to solve this problem, one has to select an appropriate representation of the desired high-order model that can be used computationally. A simple but high-complexity realization is given by the generalized companion form. Now, given this realization one can solve the problem (5.1) for a given precision which is measured using a Hermitian, strictly positive diagonal operator Γ (in fact it could be taken as $\Gamma = \epsilon I$ for some small value of ϵ), by solving

$$\sup_k \| \left((\mathcal{H} - \hat{\mathcal{H}})\Gamma^{-1} \right)_k \| \leq 1,$$

i.e., $\hat{\mathcal{H}}$ approximates $\mathcal{H}$ up to a precision given by Γ. This problem can be solved using the Schur-Takagi algorithm [17]. Indeed, Hankel norm approximation theory originates as a special case of the solution to the Schur-Takagi interpolation problem in the context of complex function theory. Several techniques were presented to find the optimal solution; see, e.g., the work of Dewilde and van der Veen [17, 37], Chandrasekaran and Gu [14, 15, 16], and Chandrasekaran et al [13]. The complexity of these techniques are normally of the order of $O(N^2)$ but can be made "fast" or "super fast" to be just of the order of $O(N)$. But in order to obtain this speed up, the matrices involved must have a special structure called the sequentially semi-separable matrix structure. This structure involves some rank conditions for optimality which cause some minor complications. This whole procedure has to be repeated for $\Gamma = c_k I$, where c_k eventually converges to a small optimal value. The principal idea of these algorithms is to use the SVD to approximate the Hankel matrices by matrices having a Hankel structure. Our algorithm follows the same line. It has the particularity that it approximates the Hankel matrices at each instant by a low rank approximation in a finite window. Let us now formulate this in more detail.

5.1. The RLRH algorithm. The key idea of this algorithm is to use the Hankel matrices $\mathcal{H}_i = \mathcal{O}_i\mathcal{C}_i$ representing the Hankel map $\mathcal{H} = \mathcal{O}\mathcal{C}$. As the system order is given by the rank of the Hankel map, it is a good idea to approximate the system by approximating the

Hankel matrices via a recursive SVD performed at each step. The technique is very similar to the previous algorithm, RLRG, but now we perform at each step the singular values decomposition of a product similar to the product $\mathcal{OC}$. Consider indeed the SVD of the matrix

$$\left[\begin{array}{c} C \\ \hline \hat{R}(i)^T A \end{array}\right] \left[\begin{array}{c|c} B & A\hat{S}(i) \end{array}\right] = U\Sigma V^T, \tag{5.2}$$

and partition $U := \left[\begin{array}{c|c} U_1 & U_2 \end{array}\right]$, $V := \left[\begin{array}{c|c} V_1 & V_2 \end{array}\right]$, where $U_1 \in \mathbb{R}^{(n+p)\times n}$ and $V_1 \in \mathbb{R}^{(n+m)\times n}$. Define then

$$\begin{aligned}
\left[\begin{array}{c|c} \hat{S}(i+1) & \hat{E}_c(i+1) \end{array}\right] &:= \left[\begin{array}{c|c} B & A\hat{S}(i) \end{array}\right] \left[\begin{array}{c|c} V_1 & V_2 \end{array}\right], \\
\left[\begin{array}{c|c} \hat{R}(i+1) & \hat{E}_o(i+1) \end{array}\right] &:= \left[\begin{array}{c|c} C^T & A^T\hat{R}(i) \end{array}\right] \left[\begin{array}{c|c} U_1 & U_2 \end{array}\right].
\end{aligned}$$

It follows that

$$\left[\begin{array}{c} \hat{R}(i+1)^T \\ \hline \hat{E}_o(i+1)^T \end{array}\right] \left[\begin{array}{c|c} \hat{S}(i+1) & \hat{E}_c(i+1) \end{array}\right] = \left[\begin{array}{c|c} \Sigma_1 & 0 \\ \hline 0 & \Sigma_2 \end{array}\right],$$

where Σ_2 contains the neglected singular values at this step. For the initialization at step $i = 0$ we use again $\hat{S}(i) = 0$, and $\hat{R}(i) = 0$. We summarize this algorithm as follows.

ALGORITHM 5.1. *Recursive Low-Rank Hankel (RLRH).*

```
procedure RLRH(A, B, C, n, tol)
    Ŝ(0) ← 0 ∈ R^{N×n}                                   ▷ Initialize Ŝ
    R̂(0) ← 0 ∈ R^{N×n}                                   ▷ Initialize R̂
    repeat
        Compute the singular value decomposition
```

$$\left[\begin{array}{c} C \\ \hline \hat{R}(i)^T A \end{array}\right] \left[\begin{array}{c|c} B & A\hat{S}(i) \end{array}\right] = U\Sigma V^T.$$

```
        Let U = [ U_1 | U_2 ], V = [ V_1 | V_2 ], U_1 ∈ R^{(n+p)×n}, V_1 ∈ R^{(n+m)×n}.
        Construct
```

$$\begin{aligned}
\hat{S}(i+1) &\leftarrow \left[\begin{array}{c|c} B & A\hat{S}(i) \end{array}\right] V_1, & \hat{R}(i+1) &\leftarrow \left[\begin{array}{c|c} C^T & A^T\hat{R}(i) \end{array}\right] U_1, \\
\hat{E}_c(i+1) &\leftarrow \left[\begin{array}{c|c} B & A\hat{S}(i) \end{array}\right] V_2, & \hat{E}_o(i+1) &\leftarrow \left[\begin{array}{c|c} C^T & A^T\hat{R}(i) \end{array}\right] U_2.
\end{aligned}$$

```
    until The stopping criterion is verified.             ▷ See Subsection 7.2
end procedure
```

Let us investigate the amount of work involved in our algorithm. First we need to form products of the type $A\hat{S}(i)$ and $\hat{R}^T(i)A$. If we assume the matrix A to be sparse and let α the number of non-zero elements per row or column of A, then the amount of work needed for this is $O(\alpha Nn)$ [19]. The construction of the left hand side of (5.2) requires an additional $2N(n+m)(n+p)$ flops and the application of the transformations U and V requires $O((p+n)(m+n)(2n+p+m))$ flops, and so the complexity of this algorithm is $O(N(p+n)(m+n))$ for each iteration. This is comparable to the work required by the RLRG algorihm.

As before we have some results linking the intermediate error matrices and the controllability and observability matrices.

THEOREM 5.2. *At each iteration, there exist unitary matrices $V^{(i)} \in \mathbb{R}^{(n+im)\times(n+im)}$ and $U^{(i)} \in \mathbb{R}^{(n+ip)\times(n+ip)}$ satisfying*

$$\mathcal{C}_i V^{(i)} = \left[\begin{array}{c|c|c} \hat{S}(i) & \hat{E}_c(i) & A\mathcal{C}_e(i) \end{array}\right], \quad \mathcal{O}_i^T U^{(i)} = \left[\begin{array}{c|c|c} \hat{R}(i) & \hat{E}_o(i) & A^T\mathcal{O}_e(i) \end{array}\right],$$

where $\hat{E}_c(i)$ and $\hat{E}_o(i)$ are the neglected parts at iteration i in the algorithm, and the matrices $\mathcal{C}_e(i)$ and $\mathcal{O}_e(i)$ are defined as follows,

$$\mathcal{C}_e(i) \doteq \left[\begin{array}{c|c|c} \hat{E}_c(i-1) & \dots & A^{i-1}\hat{E}_c(0) \end{array}\right], \mathcal{O}_e(i)^T \doteq \left[\begin{array}{c|c|c} \hat{E}_o(i-1) & \dots & \left(A^T\right)^{i-1}\hat{E}_o(0) \end{array}\right].$$

Proof. We just show the proof for $V^{(i)}$, the other being similar. At each step, there exists an orthogonal matrix $V = \left[\begin{array}{c|c} V_1 & V_2 \end{array}\right]$ such that

$$\left[\begin{array}{c|c} B & A\hat{S}(i) \end{array}\right] V = \left[\begin{array}{c|c} \hat{S}(i+1) & \hat{E}_c(i+1) \end{array}\right].$$

For $i = 0$ we have $\mathcal{C}_0 = \left[\begin{array}{c|c} \hat{S}(0) & \hat{E}_c(0) \end{array}\right]$, and so $V^{(0)} = I$. We prove the general result by induction. Suppose that there exists an orthogonal matrix $V^{(i)}$ such that

$$\mathcal{C}_i V^{(i)} = \left[\begin{array}{c|c|c|c|c} \hat{S}(i) & \hat{E}_c(i) & A\hat{E}_c(i-1) & \dots & A^{i-1}\hat{E}_c(0) \end{array}\right].$$

Since $\mathcal{C}_{i+1} = \left[\begin{array}{c|c} B & A\mathcal{C}_i \end{array}\right]$, we choose

$$V^{(i+1)} = \begin{bmatrix} I_m & 0 \\ 0 & V^{(i)} \end{bmatrix} \begin{bmatrix} V & 0 \\ 0 & I_{im} \end{bmatrix},$$

from which it follows that

$$\begin{aligned}
\mathcal{C}_{i+1}V^{(i+1)} &= \left[\begin{array}{c|c} B & A\mathcal{C}_i \end{array}\right] \begin{bmatrix} I_m & 0 \\ 0 & V^{(i)} \end{bmatrix} \begin{bmatrix} V & 0 \\ 0 & I_{im} \end{bmatrix} \\
&= \left[\begin{array}{c|c} B & A\mathcal{C}_i V^{(i)} \end{array}\right] \begin{bmatrix} V & 0 \\ 0 & I_{im} \end{bmatrix} \\
&= \left[\begin{array}{c|c|c|c|c} B & A\hat{S}(i) & A\hat{E}_c(i) & \dots & A^{i-1}\hat{E}_c(0) \end{array}\right] \begin{bmatrix} V & 0 \\ 0 & I_{im} \end{bmatrix} \\
&= \left[\begin{array}{c|c|c|c|c} \hat{S}(i+1) & \hat{E}_c(i+1) & A\hat{E}_c(i) & \dots & A^{i}\hat{E}_c(0) \end{array}\right] \\
&= \left[\begin{array}{c|c|c} \hat{S}(i+1) & \hat{E}_c(i+1) & A\mathcal{C}_e(i+1) \end{array}\right]. \qquad \square
\end{aligned}$$

As a consequence of this theorem we have the following result which give us an approximation of the original Hankel matrix $\mathcal{H}_i$.

THEOREM 5.3. *At each iteration i, there exist unitary matrices $V^{(i)} \in \mathbb{R}^{(n+im)\times(n+im)}$ and $U^{(i)} \in \mathbb{R}^{(n+ip)\times(n+ip)}$, such that*

$$\left(U^{(i)}\right)^T \mathcal{H}_i V^{(i)} = \left[\begin{array}{c|c|c} \hat{R}(i)^T\hat{S}(i) & 0 & \hat{R}(i)^T A\mathcal{C}_e(i) \\ \hline 0 & \hat{E}_o(i)^T\hat{E}_c(i) & \hat{E}_o(i)^T A\mathcal{C}_e(i) \\ \hline \mathcal{O}_e(i)A\hat{S}(i) & \mathcal{O}_e(i)A\hat{E}_c(i) & \mathcal{O}_e(i)A^2\mathcal{C}_e(i) \end{array}\right]. \tag{5.3}$$

Proof. First we have the relationship between the Hankel matrices, the controllability and observability matrices $\mathcal{H}_i \doteq \mathcal{O}_i\mathcal{C}_i$, and from the previous theorem, there exist two unitary matrices $V^{(i)} \in \mathbb{R}^{(n+im)\times(n+im)}$ and $U^{(i)} \in \mathbb{R}^{(n+ip)\times(n+ip)}$, such that

$$\left(U^{(i)}\right)^T \mathcal{H}_i V^{(i)} \doteq \left(U^{(i)}\right)^T \mathcal{O}_i\mathcal{C}_i V^{(i)} = \left[\begin{array}{c} \hat{R}(i)^T \\ \hline \hat{E}_o(i)^T \\ \hline \mathcal{O}_e(i)A \end{array}\right] \left[\begin{array}{c|c|c} \hat{S}(i) & \hat{E}_c(i) & A\mathcal{C}_e(i) \end{array}\right].$$

The final result then follows easily. ∎

This result enables us to evaluate the quality of our approximations by using the Hankel matrices (and so the Hankel map) without having to pass by Gramians, which can be very unsuitable in some cases (especially when the original system is poorly balanced). The procedure yields two matrices $\hat{S}(n)$ and $\hat{R}(n)$ of full rank n. Using these matrices, we can approximate the Gramians $\mathcal{G}_c$ and $\mathcal{G}_o$ of the original model by $\hat{S}(n)\hat{S}(n)^T$ and $\hat{R}(n)\hat{R}(n)^T$, respectively. The differences between the approximate low-rank Gramians and the exact Gramians

$$\mathcal{E}_c(i) \doteq \mathcal{G}_c(i) - \hat{\mathcal{P}}_i, \quad \mathcal{E}_o(i) \doteq \mathcal{G}_o(i) - \hat{\mathcal{Q}}_i$$

remains bounded for large i, as indicated in the following theorem.

THEOREM 5.4. *Let $\mathcal{P}$ and $\mathcal{Q}$ be the solutions of*

$$\mathcal{P} = A\mathcal{P}A^T + I, \quad \mathcal{Q} = A^T\mathcal{Q}A + I,$$

respectively. Then

$$\|\mathcal{E}_c(i)\|_2 \le \eta_c^2\|\mathcal{P}\|_2 \le \eta_c^2\frac{\kappa(A)^2}{1-\rho(A)^2}, \quad \|\mathcal{E}_o(i)\|_2 \le \eta_o^2\|\mathcal{Q}\|_2 \le \eta_o^2\frac{\kappa(A)^2}{1-\rho(A)^2},$$

where $\eta_c \doteq \max_i \|\hat{E}_c(i)\|_2$ and $\eta_o \doteq \max_i \|\hat{E}_o(i)\|_2$.

Proof. It follows from Theorem 5.2 that

$$\mathcal{E}_c(i+1) = A\mathcal{E}_c(i)A^T + \hat{E}_c(i)\hat{E}_c(i)^T, \quad \mathcal{E}_o(i+1) = A^T\mathcal{E}_o(i)A + \hat{E}_o(i)\hat{E}_o(i)^T.$$

We can also consider the equations:

$$\begin{aligned}\mathcal{X}_c(i+1) &= A\mathcal{X}_c(i)A^T + (\eta_c^2 I - \hat{E}_c(i)\hat{E}_c(i)^T), \quad \mathcal{X}_c(0) = 0,\\ \mathcal{X}_o(i+1) &= A^T\mathcal{X}_o(i)A + (\eta_o^2 I - \hat{E}_o(i)\hat{E}_o(i)^T), \quad \mathcal{X}_o(0) = 0.\end{aligned}$$

Their iterates $\mathcal{X}_c(i)$ and $\mathcal{X}_o(i)$ are clearly positive semi-definite and hence converge to the solutions $\mathcal{X}_c$ and $\mathcal{X}_o$, respectively, which are also positive semi-definite. Moreover, by linearity we have

$$\begin{aligned}\mathcal{E}_c(i+1) + \mathcal{X}_c(i+1) &= A(\mathcal{E}_c(i) + \mathcal{X}_c(i))A^T + \eta_c^2 I,\\ \mathcal{E}_o(i+1) + \mathcal{X}_o(i+1) &= A^T(\mathcal{E}_o(i) + \mathcal{X}_o(i))A + \eta_o^2 I.\end{aligned}$$

It then follows that

$$\lim_{i\to\infty} \mathcal{E}_c(i) + \mathcal{X}_c(i) = \eta_c^2\mathcal{P}, \quad \lim_{i\to\infty} \mathcal{E}_o(i) + \mathcal{X}_o(i) = \eta_o^2\mathcal{Q},$$

and we obtain $\|\mathcal{E}_c(i)\|_2 \le \eta_c^2\|\mathcal{P}\|_2$, and $\|\mathcal{E}_o(i)\|_2 \le \eta_o^2\|\mathcal{Q}\|_2$. The second bound follows from the eigendecomposition of A. ∎

THEOREM 5.5. *Using the first n columns $U_1^{(i)}$ of $U^{(i)}$ and $V_1^{(i)}$ of $V^{(i)}$, we obtain a rank n approximation of the Hankel map*

$$\mathcal{H} - U_1^{(i)}\hat{R}(i)^T\hat{S}(i)\left(V_1^{(i)}\right)^T = \mathcal{E}_H(i),$$

for which we have the error bound

$$\|\mathcal{E}_H(i)\|_2 \le \frac{\kappa(A)}{\sqrt{1-\rho(A)^2}}\max\{\eta_c\|\hat{R}^TA\|_2, \eta_o\|A\hat{S}\|_2\} + \frac{\kappa(A)^2}{1-\rho(A)^2}\eta_o\eta_c.$$

ETNA
Kent State University
http://etna.math.kent.edu

Proof. This follows directly from the bounds of Theorem 5.3 that can be used to bound the blocks in the form in (5.3) different from the $(1,1)$ block. More explicitly, from (5.3) we have

$$\mathcal{H}_i = U^{(i)} \left[\begin{array}{c|cc} \hat{R}(i)^T\hat{S}(i) & 0 & \hat{R}(i)^T A\mathcal{C}_e(i) \\ \hline 0 & E_o(i)^T E_c(i) & E_o(i)^T A\mathcal{C}_e(i) \\ \mathcal{O}_e(i)A\hat{S}(i) & \mathcal{O}_e(i)AE_c(i) & \mathcal{O}_e(i)A^2\mathcal{C}_e(i) \end{array}\right] \left(V^{(i)}\right)^T$$

and so $\mathcal{E}_H(i) = \mathcal{E}_H^{(1)}(i) + \mathcal{E}_H^{(2)}(i)$, where

$$\mathcal{E}_H^{(1)}(i) = U^{(i)} \left[\begin{array}{c|cc} 0 & 0 & \hat{R}(i)^T A\mathcal{C}_e(i) \\ \hline 0 & 0 & 0 \\ \mathcal{O}_e(i)A\hat{S}(i) & 0 & 0 \end{array}\right] \left(V^{(i)}\right)^T$$

and

$$\mathcal{E}_H^{(2)}(i) = U^{(i)} \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & \mathcal{E}_e \end{array}\right] \left(V^{(i)}\right)^T, \qquad \mathcal{E}_e = \left[\begin{array}{cc} E_o(i)^T E_c(i) & E_o(i)^T A\mathcal{C}_e(i) \\ \mathcal{O}_e(i)AE_c(i) & \mathcal{O}_e(i)A^2\mathcal{C}_e(i) \end{array}\right],$$

and thus

$$\|\mathcal{E}_H(i)\|_2 \leq \max\{\|\hat{R}(i)^T A\mathcal{C}_e(i)\|_2, \|\mathcal{O}_e(i)A\hat{S}(i)\|_2\} + \|\mathcal{E}_e\|_2. \qquad \square$$

REMARK 5.6. One obtains an approximate rank factorization of a Hankel map with i block columns and rows at each instant i. The bounds obtained in Theorems 5.4 and 5.5 are moreover independent of i. As i grows larger one can expect that reasonable approximations of η_c and η_o are in fact given by the neglected parts of the last iteration, i.e., $\eta_c \approx \|E_c(i)\|_2$ and $\eta_o \approx \|E_c(i)\|_2$, which will give much tighter bounds in these theorems. In fact, as we remarked before, η_c and η_o are function of the initialization instant and one can write

$$\eta_c(k) = \max_{k\leq i\leq\infty} \|E_c(i)\|_2, \quad \eta_o(k) = \max_{k\leq i\leq\infty} \|E_o(i)\|_2.$$

Since $\eta_c(i)$ and $\eta_o(i)$ are typically decreasing, we can replace them by the maximum over the last iteration steps.

REMARK 5.7. We can make the same convergence study, as for the RLRG algorithm, to conclude that the RLRH algorithm has a unique fixed point which is $\{A, B\}$ invariant and $\{A, C\}$ invariant at the same time. This leads to the conclusion that the fixed point in this case is the dominant part of the common "balanced" Gramian. This property will also imply a very nice result for the reduced model that we show in the following section.

REMARK 5.8. If a bilinear transformation T is applied to the system $\{A, B, C\}$ to get a new system $\{T^{-1}AT, T^{-1}B, CT\}$, the corresponding controllability and observability matrices and the Hankel map, respectively, will be

$$\hat{\mathcal{C}} = T^{-1}\mathcal{C}, \quad \hat{\mathcal{O}} = \mathcal{O}T, \quad \hat{\mathcal{H}} = \hat{\mathcal{O}}\hat{\mathcal{C}} = \mathcal{O}\mathcal{C} = \mathcal{H}.$$

This transformation will not affect the Hankel map (which is taken into account for the balancing). This means that for any realization of the system, RLRH-ABT will do as good as for a balanced realization. And so it is a powerful very cheap method to produce a good balanced approximation to a linear system.

ETNA
Kent State University
http://etna.math.kent.edu

6. Approximate balanced truncation using the RLRG and RLRH algorithms. Using the two previous algorithms, RLRG and RLRH, we can use the idea of approximate balanced truncation (see Section 3) to obtain a reduced order model. The idea here is to use low-rank approximations of the Gramians, obtained via RLRG or RLRH, instead of the original Cholesky factors of the Gramians in the balanced truncation algorithm. The implemented algorithms are given by Algorithms 6.1 and 6.2.

ALGORITHM 6.1. *RLRG Approximate Balanced Truncation (RLRG_ABT).*

1: ***procedure*** RLRG_ABT(A, B, C, n, tol)
2: *Run RLRG (Algorithm 4.1) to get low-rank approximations $S, R \in \mathbb{R}^{N\times n}$ of the Cholesky factors of the Gramians $\mathcal{G}_c$ and $\mathcal{G}_o$, respectively.*
3: *Calculate the singular value decomposition $S^T R = U\Sigma V^T$.*
4: *Let $X = SU\Sigma^{-1/2}$, and $Y = RV\Sigma^{-1/2}$.*
5: *The order n approximate truncated balanced realization is given by*
$$\tilde{A} = Y^* AX, \quad \tilde{B} = Y^* B, \quad \tilde{C} = CX.$$
6: ***end procedure***

ALGORITHM 6.2. *RLRH Approximate Balanced Truncation (RLRH_ABT).*

---

procedure RLRH_ABT(A, B, C, n, tol)
Run RLRH (Algorithm 5.1) to get low-rank approximations $\hat{S}, \hat{R} \in \mathbb{R}^{N\times n}$ of the Cholesky factors of the Gramians $\mathcal{G}_c$ and $\mathcal{G}_o$, respectively.
Let $X = S\Sigma^{-1/2}$, and $Y = R\Sigma^{-1/2}$.
The order n approximate truncated balanced realization is given by
$$\hat{A} = Y^* AX, \quad \hat{B} = Y^* B, \quad \hat{C} = CX.$$
end procedure

In Algorithm 6.1, we use the SVD in Line 3 to "balance" the projection matrices. This is crucial because we approximate the Gramians independently. In practice, if the system has poles close to the unit circle, one or both Gramians are not well approximated. This is not the case in Algorithm 6.2, because the product of the two low-rank approximations is already equal to a diagonal matrix of nonnegative values. This is the first advantageous property of the RLRH approximate balanced truncation method.

These two approximate balanced truncation algorithms have some very desirable properties that we show below.

THEOREM 6.3. *Both the algorithms RLRG_ABT and RLRH_ABT lead to a balanced stable reduced model.*

Before giving the proof of this result, we will need to show the following lemma.

LEMMA 6.4. *Let $X, Y \in \mathbb{R}^{k\times l}$ and $Z \in \mathbb{R}^{k\times r}$. If $X^T Z = 0$ and $X^T Y$ is full rank then $Y^T Z = 0$.*

Proof. Firstly, as $X^T Y$ is full rank, the columns of X and those of Y span the same subspace of $\mathbb{R}^{k\times l}$. Secondly, as $X^T Z = 0$, the columns of Z span a subspace of $\mathbb{R}^{k\times r}$ that is orthogonal to the subspace spanned by the columns of X into $\mathbb{R}^{k\times(l+r)}$. Then the subspace spanned by the columns of Y is also orthogonal to the subspace spanned by the columns of Z, i.e., $Y^T Z = 0$. $\square$

Proof of Theorem 6.3. We will prove the theorem for the RLRG_ABT algorithm; the proof for the other algorithm is similar. Let S and R be the fixed points of the RLRG algorithm applied to the system $\{A, B, C\}$ (2.1), i.e.,

$$\begin{bmatrix} S \mid E_c \end{bmatrix} = \begin{bmatrix} AS \mid B \end{bmatrix} V_c, \quad \begin{bmatrix} R \mid E_o \end{bmatrix} = \begin{bmatrix} A^T R \mid C^T \end{bmatrix} V_o, \tag{6.1}$$

where $V_c \in \mathbb{R}^{(n+m)\times(n+m)}$ and $V_o \in \mathbb{R}^{(n+p)\times(n+p)}$ are unitary matrices. It follows that

$$SS^T + E_c E_c^T = ASS^T A^T + BB^T, \quad RR^T + E_o E_o^T = A^T RR^T A + C^T C. \tag{6.2}$$

Recall that the projection matrices are

$$\pi_l = RV\Sigma^{-\frac{1}{2}}, \quad \pi_r = SU\Sigma^{-\frac{1}{2}}, \quad S^T R = U\Sigma V^T, \quad \text{where} \quad U, \Sigma, V \in \mathbb{R}^{n\times n}.$$

Now, using these projection matrices we can project both equations (6.2), and we obtain, respectively,

$$\pi_l^T \left(SS^T + E_c E_c^T\right) \pi_l = \pi_l^T ASS^T A^T \pi_l^T + \pi_l^T BB^T \pi_l, \tag{6.3}$$

$$\pi_l^r \left(RR^T + E_o E_o^T\right) \pi_r = \pi_r^T A^T RR^T A\pi_r + \pi_r^T C^T C\pi_r. \tag{6.4}$$

By definition we have $\pi_l^T \pi_r = I_n$, and by construction we have $S^T E_c = 0$ and $R^T E_o = 0$. Moreover, we have

$$\pi_l^T S = \Sigma^{-\frac{1}{2}} V^T R^T S = \Sigma^{\frac{1}{2}} U, \quad \pi_r^T R = \Sigma^{-\frac{1}{2}} U^T S^T R = \Sigma^{\frac{1}{2}} V^T.$$

Applying the previous lemma yields that $\pi_l^T E_c = 0$ and $\pi_r^T E_o = 0$. Then equations (6.3) and (6.4) become

$$\begin{aligned} \pi_l^T SS^T \pi_l &= \pi_l^T ASS^T A^T \pi_l^T + \pi_l^T BB^T \pi_l, \\ \pi_r^T RR^T \pi_r &= \pi_r^T A^T RR^T A\pi_r + \pi_r^T C^T C\pi_r. \end{aligned}$$

We can check easily that (as U and V are unitary matrices)

$$\pi_l^T SS^T \pi_l = \pi_r^T RR^T \pi_r = \Sigma, \quad SS^T = \pi_r \Sigma \pi_r^T, \quad \text{and} \quad RR^T = \pi_l \Sigma \pi_l^T.$$

Finally, we obtain the Stein equations

$$\Sigma = \pi_l^T A\pi_r \Sigma \pi_r^T A^T \pi_l^T + \pi_l^T BB^T \pi_l \quad \text{and} \quad \Sigma = \pi_r^T A^T \pi_l \Sigma \pi_l^T A\pi_r + \pi_r^T C^T C\pi_r.$$

These two equations prove that the reduced model $\{\pi_l^T A\pi_r, \pi_l^T B, C\pi_r\}$ has a balanced Gramian Σ. This Gramian is by construction positive definite and the solution of the last two Stein equations, from which we conclude that the reduced system is stable. ☐

The next result concerns the convergence of the computed Hankel singular values.

THEOREM 6.5. *Let σ_i and $\hat{\sigma}_i$ be the Hankel singular values of the original model and the reduced model via either RLRG_ABT or RLRH_ABT respectively: $\sigma_i^2 = \lambda(\mathcal{G}_c\mathcal{G}_o)$ and $\sigma_i^2 = \lambda(SS^T RR^T)$. Then*

$$\sum_{i=1}^{n} \sigma_i^2 - \sum_{i=1}^{n} \hat{\sigma}_i^2 \leq \frac{\kappa(A)^2}{1-\rho(A)^2} \left(m\eta_c^2 \operatorname{trace}(\mathcal{G}_o) + p\eta_o^2 \operatorname{trace}(\mathcal{G}_c)\right),$$

where m is the number of inputs, p is the number of outputs, and η_c and η_o are the corresponding noise levels.

Proof. We have

$$\begin{aligned}\sum_{i=1}^{n}\sigma_i^2-\sum_{i=1}^{n}\hat{\sigma}_i^2 &= \mathrm{trace}(\mathcal{G}_c\mathcal{G}_o)-\mathrm{trace}(SS^TRR^T)\\ &= \mathrm{trace}(\mathcal{G}_c\mathcal{G}_o-SS^TRR^T)\\ &= \mathrm{trace}\left((\mathcal{G}_c-SS^T)\mathcal{G}_o+SS^T(\mathcal{G}_o-RR^T)\right)\\ &\le \mathrm{trace}(\mathcal{G}_c-SS^T)\mathrm{trace}(\mathcal{G}_o)+\mathrm{trace}(SS^T)\mathrm{trace}(\mathcal{G}_o-RR^T).\end{aligned}$$

And using previous results in this paper we obtain finally

$$\sum_{i=1}^{n}\sigma_i^2-\sum_{i=1}^{n}\hat{\sigma}_i^2 \le \frac{\kappa(A)^2}{1-\rho(A)^2}\left(m\eta_c^2\mathrm{trace}(\mathcal{G}_o)+p\eta_o^2\mathrm{trace}(\mathcal{G}_c)\right). \qquad \square$$

Here, it should be mentioned that for RLRG_ABT the noise levels η_c and η_o could be not of the same order as the Gramians are approximated independently. This could affect the quality of the previous bound. On the other hand, for the RLRH_ABT, we have $\eta = \eta_c \simeq \eta_o$, which yields

$$\sum_{i=1}^{n}\sigma_i^2-\sum_{i=1}^{n}\hat{\sigma}_i^2 \le \frac{\kappa(A)^2\eta^2}{1-\rho(A)^2}\left(m\ \mathrm{trace}(\mathcal{G}_o)+p\ \mathrm{trace}(\mathcal{G}_c)\right).$$

Another result for the Hankel singular values is obtained using the perturbation theory for the singular values; see [19, Page 449] and Theorem 5.5.

THEOREM 6.6. *Let σ_i and $\hat{\sigma}_i$ be the Hankel singular values of the original model and the reduced model via RLRH_ABT. Then for $i = 1,\ldots,n$*

$$|\sigma_i-\hat{\sigma}_i| \le \frac{\kappa(A)}{\sqrt{1-\rho(A)^2}}\eta\hat{\sigma}_1\|A\|_2+\frac{\kappa(A)^2}{1-\rho(A)^2}\eta^2,$$

where η is the noise level.

Proof. We apply [19, Corollary 8.6.2] and Theorem 5.5. We also use the fact that $\eta = \eta_c \simeq \eta_o, \quad \|\hat{S}\|_2 = \|\hat{R}\|_2 = \hat{\sigma}_1.$ $\square$

7. Further discussion.

7.1. Quality of the bounds. All our bounds are a function of $\frac{\kappa(A)^2}{1-\rho(A)^2}$ (or its square root). At first sight this appeared to be a disappointing property of our algorithms. It suggests that our proposed algorithms do not work unless the problem at hand is very well conditioned and the spectral radius of A is far enough from 1. But, one also should notice that in every term where the expression $\frac{\kappa(A)^2}{1-\rho(A)^2}$ appears, it is multiplied by the square of one of the noise levels (either η_c or η_o). These noise levels are of the order of the machine epsilon and in most cases will make these terms small. This is illustrated in our numerical examples.

7.2. Stopping criterion. Since our iterative method computes successive approximations to the solution of a Lyapunov equation, a practical test is needed to determine when to stop the iteration. Ideally this test should measure the distance of the last iterate to the true solution (the Gramian), but this is not possible as the true solution is unknown. Instead, various other metrics are used, typically involving the residual (noise level) or reached fixed point. The following stopping criteria could be considered:

- *Maximal number of iteration steps*. The iteration is stopped after a certain number $i_{\max}$ of iterations steps. Obviously, no additional computations need to be performed to evaluate it. The drawback of this stopping criterion is that it is not related to the attainable accuracy of the delivered low-rank Gramian.
- *Stagnation of the canonical angles*. The iteration is stopped when stagnation of $\sphericalangle(S(i-1), S(i))$ is detected. Roughly speaking, these angles are considered as "stagnating" when no noticeable decrease is observed in consecutive iteration steps. This criterion works well in practice. It requires the computation of an SVD, which gives the cosines of these angles.
- *Stagnation of* Σ_c. We predefine a tolerance ϵ_m and test if $\|\Sigma_c(i) - \Sigma_c(i-1)\| \leq \epsilon_m$ for several iterations, in the 2-norm or the Frobenius norm.
- *Smallness of the noise* η_c. We predefine a tolerance ϵ_m and test if $\eta_c \leq \epsilon_m$ for several iterations. Loosely speaking, this means the following. When η_c and consequently $\|E_c(i)\|$ become smaller than ϵ_m , then the "contribution" from the following iterations is not needed as it will not ameliorate the quality of the approximation.

In general, the three last criteria are affected by round-off errors, which is why we should wait a few more steps before stopping of the algorithm. Note that the delay between stagnation and stopping of the algorithm can be changed; in our algorithms we consider a delay of 10 steps. In practice, the second and third stopping criteria are combined to have a good low-rank approximation of the Gramians; see the discussion following Theorem 4.11. The two last stopping criteria could be considered as equivalent, as a stagnation of Σ_c means that the noise levels are very small and negligible.

7.3. The choice of n. So far we have only considered the case where the reduced order n is constant and fixed from the beginning by the user. But actually, if one wants to choose a convenient value for n one has to do an explicit thorough analysis of the whole Hankel operator (or matrix) involved and strive for some sort of singular value ranking. For large-scale dimensions this pre-treatment is prohibitive.

The current situation is that we can choose dynamically the reduced order by choosing the number of vectors kept during the iterations of the algorithm, i.e., $n = n_i$ is variable. This is very cheap as we already pass through the whole matrix with a kind of a sliding window which sorts locally the singular values. And so one can adapt n_i as soon as the information "unveiled" by the sliding window is relevant to the approximation. One should notice that as we are using SVD-based algorithms, the quality of the approximations will be a function of the existence and the size of the gap between what we keep and what we neglect [19]. Here, one can adopt many strategies using some ad-hoc specification, e.g.,

- *Absolute tolerance strategy*. In this case, one has to predefine a tolerance value ς_a and ask the algorithm to neglect all singular values which are smaller than this tolerance, i.e., $n_i = \min\{j : \sigma_j(S) < \varsigma_a\}$.
- *Relative tolerance strategy*. This strategy is more dynamic and suitable. Typically, the user can define an interval $[n_{min}, n_{max}] \subset \mathbb{N}$ and the algorithm has to find the optimal value for n_i, such that $n_{min} \leq n_i \leq n_{max}$. By optimal, we mean the smallest n_i such that the quality of the approximations is acceptable. Let ς_r be a pre-specified tolerance value. At each iteration we apply our algorithm and we check for all computed singular values $\sigma_j(S)$, $j = 1 : n_i + m$, the quotient $\sigma_j(S)/\sigma_1(S)$, for $j = 1 : n_i + m$. The first j for which we will have $\sigma_j(S)/\sigma_1(S) \leq \varsigma_r$, is compared to n_i; if this j is smaller than n_i then we take the next n_{i+1} equal to n_i (i.e., $n_{i+1} = n_i$), otherwise we take $n_{i+1} = j$, and so on.
- Another strategy can be adopted for the choice of n_i. It is based on the fact that the quality of the approximation depends on the gap between the retained values and the neglected

ETNA
Kent State University
http://etna.math.kent.edu

ones. So one can detect the gaps between singular values in each window, and adapt n as for the relative tolerance strategy.

In the strategies above, because $n_{min} \leq n_1 \leq n_2 \leq \cdots \leq n_{max}$, if one keeps in memory all values of n_i, we can choose at the end the low-rank approximation only from the n-rank approximation, which will embed all other n_i-rank approximations. Of course, the pre-specification of ς_a or ς_r will be crucial.

8. Numerical examples. In this section we apply our algorithms to four different dynamical systems: a building model, a CD player model, and two International Space Station models. These benchmarks are described in more detail in [10, 11, 20]. These models are continuous, so we discretize each system using a bilinear transformation with parameter $\xi = 2$ [3]. In Table 8.1 we give the order of the system (N), the number of inputs (m), and outputs (p), the order of reduced system (n), and the corresponding tolerance value. We show also in this table the spectral radii and the condition numbers of the matrices A.

TABLE 8.1
Summary of data of the benchmark models.

	N	m	p	n	tol.value	$\rho(A)$	$\kappa(A)$
build model	48	1	1	10	0.16	0.4997	$8.0478.10^3$
CD player model	120	2	2	24	$2.8.10^{-7}$	0.5266	$1.7793.10^4$
ISS 1R model	270	3	3	32	2.10^{-3}	0.7338	$9.6802.10^3$
ISS 12A model	1412	3	3	195	65.10^{-4}	0.8310	$5.7728.10^3$

TABLE 8.2
H_∞ norm of benchmark models, and the error systems.

model	$\|\mathcal{S}\|_\infty$	$\frac{\|\mathcal{S}-\mathcal{S}_{opt}\|_\infty}{\|\mathcal{S}\|_\infty}$	$\frac{\|\mathcal{S}-\hat{\mathcal{S}}_1\|_\infty}{\|\mathcal{S}\|_\infty}$	$\frac{\|\mathcal{S}-\hat{\mathcal{S}}_2\|_\infty}{\|\mathcal{S}\|_\infty}$
Building	0.0053	0.1143	0.4301	0.4320
CD player	$2.3198.10^6$	$8.0704.10^{-8}$	$6.8931.10^{-6}$	$1.7.10^{-6}$
ISS 1R	0.1159	0.0013	0.1023	0.0979
ISS 12A	0.0107	0.0071	0.9697	0.9390

TABLE 8.3
Noise levels $\mu_\bullet$ for benchmark models.

model	μ_c^{RLRG}	μ_o^{RLRG}	μ_c^{RLRH}	μ_o^{RLRH}
Building	$6.5063.10^{-15}$	$4.3799.10^{-12}$	$7.7292.10^{-15}$	$3.2445.10^{-11}$
CD player	$4.0575.10^{-20}$	$6.0341.10^{-20}$	$1.6867.10^{-14}$	$2.8846.10^{-13}$
ISS 1R	$1.5063.10^{-8}$	$1.1641.10^{-10}$	$6.2550.10^{-8}$	$1.6270.10^{-9}$
ISS 12A	$7.1973.10^{-25}$	$1.1751.10^{-27}$	$7.5093.10^{-22}$	$4.7292.10^{-25}$

The first remark is that because we work directly on the Hankel map (with RLRH method) we do not need to “balance” (using an SVD) the projection matrices to obtain a convenient reduced-order model.

ETNA
Kent State University
http://etna.math.kent.edu

TABLE 8.4
CPU time for different algorithms.

model	BT	RLRG_ABT	RLRH_ABT
Building	0.3750	0.3380	0.0810
CD player	0.7970	0.7340	0.7030
ISS 1R	11.6720	4.7350	2.5470
ISS 12A	$1.1327.10^3$	$0.1029.10^3$	$0.0282.10^3$

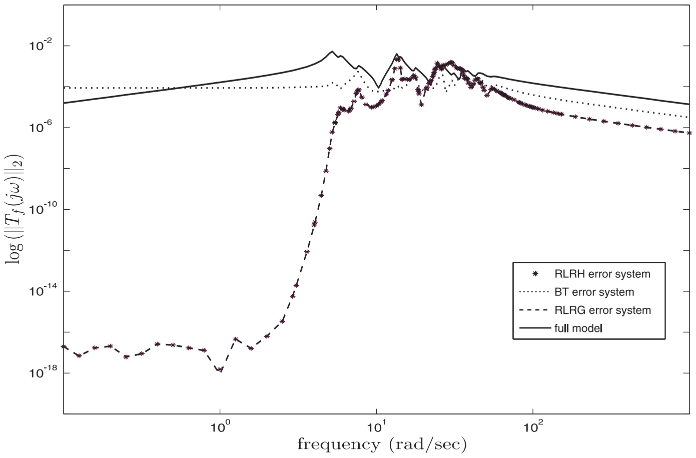

FIG. 8.1. $\sigma_{\max}$-*plot of the frequency responses for the building model.*

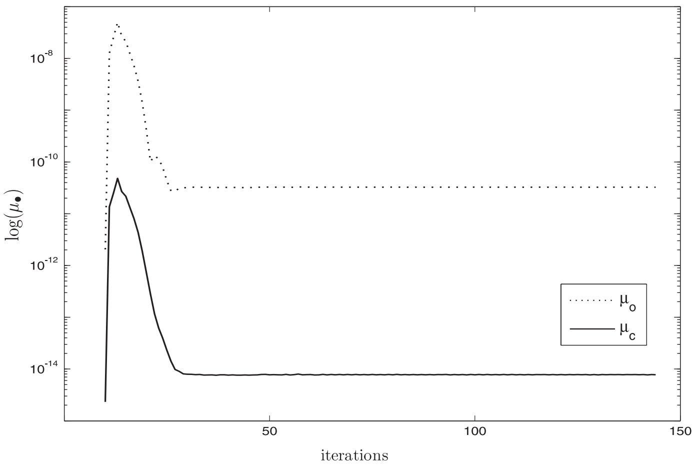

FIG. 8.2. *Evolution of the values of the noise levels for the building model.*

For each example, the relative $\mathcal{H}_\infty$ norms of the full system $\mathcal{S}$ and the error systems are tabulated in Table 8.2, and the σ_{max}-plot of the full order and the corresponding error system are shown in Figures 8.1, 8.3, 8.5, and 8.7. We use the notations $\mathcal{S}_{opt}$ for the reduced order model by balanced truncation, $\hat{\mathcal{S}}_1$ for the reduced order model by RLRG_ABT algorithm, and $\hat{\mathcal{S}}_2$ for the reduced order model by RLRH_ABT algorithm.

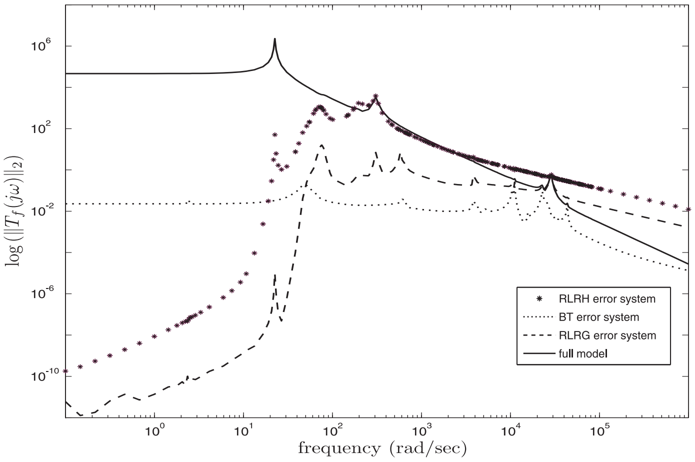

FIG. 8.3. $\sigma_{\max}$*-plot of the frequency responses for the CD player model.*

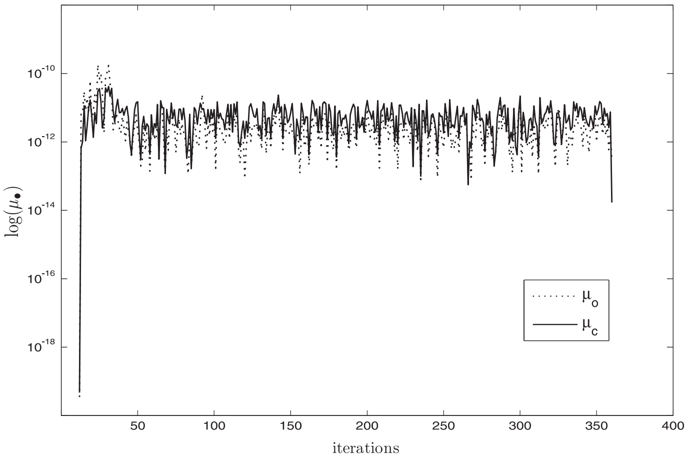

FIG. 8.4. *Evolution of the values of the noise levels for the CD player model.*

It can be seen from Figures 8.1, 8.3, 8.5, and 8.7 that we obtain with the RLRH approximation results which are close to those obtained via BT. These results are also close of those of RLRG approximation, but we have applied the RLRG algorithm to the controllability and observability matrices with a $\hat{n}$, where $\hat{n} > n$, and we have balanced the projection matrices using an SVD to keep only n projection matrices. These operations make the RLRG more expensive, and so the RLRH algorithm is less expensive and the results are as good as those obtained using the RLRG approximation.

Figures 8.2, 8.4, 8.6, and 8.8 show the noise levels η_c and η_o. Notice that the noise levels shown must be interpreted also in a special way as it was done for the RLRG algorithm. The noise levels must be multiplied by the corresponding power of the spectral radius of A to obtain the real values of the noise level at the end, i.e., the real noise level $\tilde{\mu}_\bullet$ is obtained as $\tilde{\mu}_\bullet(i) \doteq \rho(A)^{\tau-i}\mu_\bullet(i)$, where τ is the number of iteration. Therefore, the values of noise levels considered in the previous theorems will be taken in the last obtained values, which will be very small. We notice here also that for poorly balanced systems the resulting

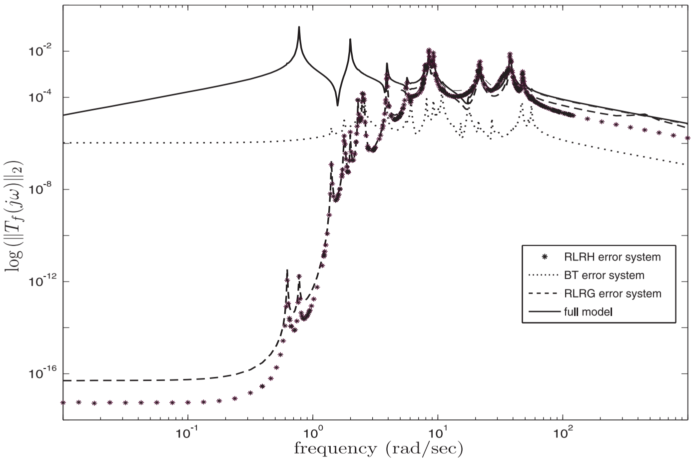

FIG. 8.5. $\sigma_{\max}$*-plot of the frequency responses for the ISS 1R model.*

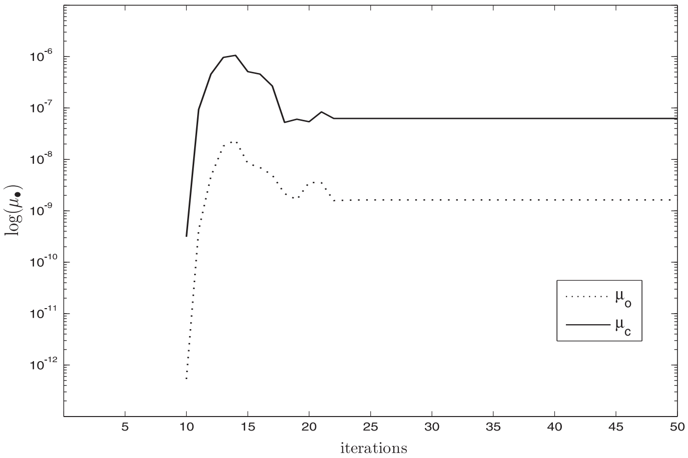

FIG. 8.6. *Evolution of the values of the noise levels for the ISS 1R model.*

noise levels are not of the same order as for well balanced systems. This is still the case for the CD player model. We remark also that for "close balanced" systems, like the CD player model ($\kappa(T) = 40.7341$, where T is the balancing transformation) RLRG yields better results. But, RLRH is as least better for "poorly balanced" systems. This is the case for the Building model ($\kappa(T) = 347.0781$) and more clearly for the International space station ($\kappa(T) = 7.4018.10^5$). Of course, RLRH is always faster and cheaper as we do not need to balance the approximations at the end of the algorithm (by computing the SVD of a product of two tall and skinny matrices).

9. Concluding remarks. In this paper, we proposed two recursive approximate balanced truncation model reduction methods based on the Gramians and the Hankel map. Subsequently the approaches for computing approximate Gramians and Hankel map were derived. These approaches provide results close to those obtained by balanced truncation, considered to be optimal, with lower computational cost. Unlike all other methods in the literature, the reduced order model produced by our methods are guaranteed to be stable and

ETNA
Kent State University
http://etna.math.kent.edu

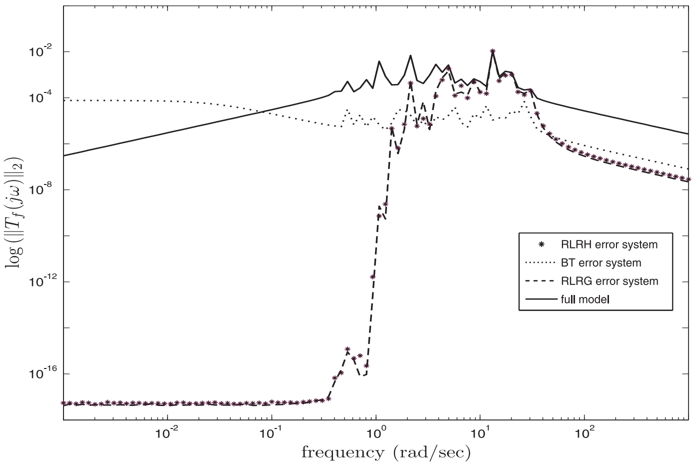

FIG. 8.7. $\sigma_{\max}$*-plot of the frequency responses for the ISS 12A model.*

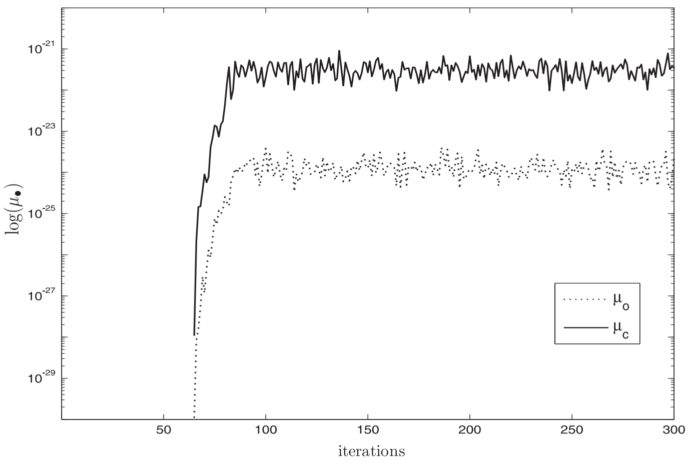

FIG. 8.8. *Evolution of the values of the noise levels for the ISS 12A model.*

balanced. Bounds on the quality of the approximation are given with some numerical examples. The RLRH algorithm is the best algorithm for approximating the balanced truncation in terms of accuracy and computational cost. Its cost is $O(N(n+m)(n+p))$, which is only linear in the large dimension N, unlike balanced truncation which has a cost which is cubic in the large dimension, i.e., $O(N^3)$. The numerical examples show that this algorithm has very good properties in term of stability, convergence rate and the quality of the approximation.

Despite the obviously desirable features of the Hankel map approach proposed here, many open questions remain. There are a number of refinements with respect to performance, convergence, and accuracy which require more theoretical and algorithmic analysis. There is one particularly interesting feature concerning the comparison between the original Hankel map and the Hankel map of the reduced order model. For instance, we just compared the original Hankel map and its dominant block approximation. To compare the two Hankel maps we still need a better understanding of the algorithm and its features.

Acknowledgements. I gratefully acknowledge the helpful remarks and suggestions, by Nicholas J. Higham, Paul Van Dooren and the two anonymous reviewers, which significantly improved the presentation and the quality of this paper.

REFERENCES

[1] A. ANTOULAS, *Approximation of Large-Scale Dynamical Systems*, SIAM, Philadelphia, 2005.
[2] A. ANTOULAS, D. SORENSON, AND Y. ZHOU, *On the decay rate of Hankel singular values and related issues*, Systems Control Lett., 46 (2002), pp. 323–342.
[3] K. ÅSTRÖM AND B. WITTENMARK, *Computer-Controlled Systems: Theory and Design*, Prentice Hall, 1997.
[4] P. BENNER, M. CASTILLO, E. QUINTANA-ORTÍ, AND V. HERNÁNDEZ, *Parallel partial stabilizing algorithms for large linear control systems*, J. Supercomput., 615 (2000), pp. 193–206.
[5] P. BENNER, E. QUINTANA-ORTÍ, AND G. QUINTANA-ORTÍ, *Balanced truncation model reduction of large-scale dense systems on parallel computers*, Math. Comput. Model. Dyn. Syst., 6 (2000), pp. 383–405.
[6] ———, *Numerical solution of discrete stable linear matrix equations on multicomputers*, Parallel Algorithms Appl., 17 (2002), pp. 127–146.
[7] Å. BJÖRCK AND G. GOLUB, *Numerical methods for computing angles between linear subspaces*, Math. Comp., 27 (1973), pp. 579–594.
[8] Y. CHAHLAOUI, *Low-Rank Approximations and Model Reduction*, PhD Thesis, num. 14/2003, Universite Catholique de Louvain, Louvain-La-Neuve, 2003.
[9] ———, *A posteriori error bounds for discrete balanced truncation*, MIMS Eprints 2009.12, available at http://eprints.ma.man.ac.uk/1224, to appear in Linear Algebra Appl., 2011.
[10] Y. CHAHLAOUI AND P. VAN DOOREN, *A collection of benchmark examples for model reduction of linear time invariant dynamical systems*, SLICOT Working Note 2002-2, 2002, available from ftp://wgs.esat.kuleuven.ac.be/pub/WGS/REPORTS/SLWN2002-2.ps.Z.
[11] ———, *Benchmark examples for model reduction of linear time invariant dynamical systems*, 45 (2005), pp. 379–392.
[12] ———, *Model reduction of time-varying systems*, 45 (2005), pp. 131–148.
[13] S. CHANDRASEKARAN, P. DEWILDE, M. GU, T. PALS, X. SUN, A. J. VAN DER VEEN, AND D. WHITE, *Some fast algorithms for sequentially semiseparable representations*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 341–364.
[14] S. CHANDRASEKARAN AND M. GU, *Fast and stable eigendecomposition of symmetric banded plus semi-separable matrices algorithms for banded plus semi-separable matrices*, Linear Algebra Appl., 313 (2000), pp. 107–114.
[15] ———, *Fast and stable algorithms for banded plus semi-separable matrices*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 373–384.
[16] ———, *A Divide-and-Conquer algorithm for the eigendecomposition of symmetric block-diagonal plus semiseparable matrices*, Numer. Math., 96 (2004), pp. 723–731.
[17] P. DEWILDE AND A.-J. VAN DER VEEN, *Time-Varying Systems and Computations*, Kluwer Academic Publishers, Boston, 1998.
[18] K. GLOVER, *All optimal Hankel norm approximations of linear multivariable systems and their $\mathcal{L}^\infty$-error bounds*, Internat. J. Control, 39 (1984), pp. 1115–1193.
[19] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd edition, Johns Hopkins Press, Baltimore, 1996.
[20] S. GUGERCIN, A. ANTOULAS, AND N. BEDROSSIAN, *Approximation of the international space station 1r and 12a flex models*, in Proceedings of the 40th IEEE Conference on Decision and Control, 2001, pp. 1515–1516.
[21] S. GUGERCIN, D. SORENSON, AND A. ANTOULAS, *A modified low-rank Smith method for large-scale Lyapunov equations*, Numer. Algorithms, 32 (2003), pp. 27–55.
[22] N. HIGHAM, *Functions of Matrices: Theory and Computation*, SIAM, Philadelphia, 2008.
[23] D. HU AND L. REICHEL, *Krylov-subspace methods for the Sylvester equation*, Linear Algebra Appl., 172 (1992), pp. 283–313.
[24] I. JAIMOUKHA AND E. KASENALLY, *Krylov subspace methods for solving large Lyapunov equations*, SIAM J. Numer. Anal., 31 (1994), pp. 227–251.
[25] T. KAILATH, *Linear Systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1980.
[26] J. ORTEGA AND W. RHEINBOLDT, *Iterative Solution of Nonlinear Equations in Several Variables*, SIAM, Philadelphia, 2000.
[27] T. PENZL, *Numerical solution of generalized Lyapunov equations*, Adv. Comput. Math., 8 (1998), pp. 33–48.
[28] ———, *Algorithms for model reduction of large dynamical systems*, Technical Report SFB393/99-40, Sonderforschungsbereich 393 Numerische Simulation auf massiv parallelen Rechnern, TU Chemnitz, 1999, available at http://www.tu-chemnitz.de/sfb393/sfb99pr.html.

[29] ———, *A cyclic low-rank Smith method for large sparse Lyapunov equations*, SIAM J. Sci. Comput., 21 (2000), pp. 1404–1418.
[30] ———, *Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case*, Systems Control Lett., 40 (2000), pp. 139–144.
[31] Y. SAAD, *Numerical solutions of large Lyapunov equations*, in Signal Processing, Scattering, Operator Theory, and Numerical Methods, M. A. Kaashoek, J. H. Van Schuppen, and A. C. Ran, eds., Birkhäuser, Basel, 1990, pp. 503–511.
[32] A. SCOTTEDWARD HODEL, *Numerical Methods for the Solution of Large and Very Large, Sparse Lyapunov Equations*, PhD thesis, University of Illinois at Urbana-Champaign, Champaign, Illinois, 1989.
[33] R. SMITH, *Matrix equation $XA + BX = C$*, SIAM J. Appl. Math., 16 (1968), pp. 198–201.
[34] G. STEWART, *Matrix Algorithms*, vol. 1, SIAM, Philadelphia, 1998.
[35] ———, *Matrix Algorithms*, vol. 2, SIAM, Philadelphia, 2001.
[36] G. STEWART AND J. SUN, *Matrix Perturbation Theory*, Academic Press, San Diego, 1990.
[37] A.-J. VAN DER VEEN AND P. DEWILDE, *On low-complexity approximation of matrices*, Linear Algebra Appl., 203 (1992), pp. 1145–1201.
[38] A. VARGA, *Balancing related methods for minimal realization of periodic systems*, Systems Control Lett., 36 (1999), pp. 339–349.
[39] E. WACHSPRESS, *Iterative solution of the Lyapunov matrix equation*, Appl. Math. Lett., 1 (1988), pp. 87–90.