

System Identification via CUR-factored Hankel Approximation

ACDL Technical Report TR16-5

Boris Kramer* Alex A. Gorodetsky†

October 7, 2016

Subspace-based system identification for dynamical systems is a sound, system-theoretic way to obtain linear, time-invariant system models from data. The interplay of data and systems theory is reflected in the Hankel matrix, a block-structured matrix whose factorization is necessary for system identification. For systems with many inputs, many outputs, or large time-series of system-response data, established methods based on the singular value decomposition (SVD)—such as the eigensystem realization algorithm (ERA)—are prohibitively expensive. In this paper, we propose an algorithm to reduce the complexity, with respect to Hankel matrix size, of the eigensystem realization algorithm (ERA) from cubic to quadratic. This reduction is realized by replacing the SVD with a CUR decomposition that directly seeks a low-rank approximation of the Hankel matrix. The CUR decomposition is obtained by selecting a small number rows and columns using a maximum-volume-based cross approximation scheme. We present a worst-case error bound for our resulting system identification algorithm, and we demonstrate its computational advantages and accuracy on a numerical example. The example demonstrates that the resulting identification yields almost indistinguishable results compared with the SVD-based ERA, yet comes with significant computational savings.

*Corresponding author. Massachusetts Institute of Technology, Cambridge, MA 02125, (bokramer@mit.edu, <http://web.mit.edu/bokramer/www/>). The research of this author was supported by the DARPA EQUiPS program award number UTA15-001067 (program manager F. Fahroo).

†Massachusetts Institute of Technology, Cambridge, MA (alex@alexgorodetsky.com, www.alexgorodetsky.com). The research of this author was supported by National Science Foundation through grant IIS-1452019.

1. Introduction

We are motivated to enable subspace-based system identification from a large amount of system-response data. In particular, we are interested in using experimental or simulated data to compute reduced-order models [1], which can then be used for design, optimization, and control. Building on system-theoretic concepts, the system identification involves computing a decomposition of a large matrix—the Hankel matrix. A Hankel matrix is a block-structured matrix, where the data on parallels to the main anti-diagonal are equal.

Historically, the field of model reduction for large-scale systems heavily relied on advances in numerical linear algebra to tackle both scalability and computational feasibility of algorithms. After the introduction of a computationally feasible and stable singular value decomposition (SVD) by Golub and Reinsch in 1970 [5], Kung [14] proposed in 1979 a system identification algorithm by computing an approximate decomposition of the Hankel matrix through an SVD. A version of this algorithm was later termed “Eigen-system Realization Algorithm (ERA)” in [12], and it became popular in engineering, due to its simplicity and guaranteed stability. The method uses impulse response data to construct a linear, time-invariant system representation—of any order—that best matches the data, by using system theoretic concepts.

One step of this widely used algorithm involves computing the SVD of the Hankel matrix. This operation is computationally infeasible for large Hankel matrices often encountered in practice, due to its cubic complexity with respect to matrix size. In prior work [13], we were able to mitigate this computational bottleneck, specifically for Hankel matrices that arose from large input/output dimensions, using a suitable tangential interpolation of the data.

In this work, we propose a more general, and faster, method to combat the expense of SVD-based ERA for systems with large Hankel matrices. Namely, we use the CUR decomposition to directly seek low-rank factorizations of Hankel matrices, which can then be used in an ERA-based framework. We show that this results in a system identification algorithm with costs that are linear in the row and column dimension of the Hankel matrix, versus cubic, for SVD-based methods. Moreover, we obtain error bounds on the approximation of the impulse response sequence by the identified reduced model.

The CUR decomposition [10, 2, 3, 18, 16] has emerged as a successful alternative to the SVD in the area of big data and search/compression algorithms. Recently, Sorensen and Embree provided a new row and column selection that leads to accurate matrix approximations, and provided a thorough error analysis in [19]. In this paper, we extend the CUR to system-theoretic approaches to system identification.

Algorithms for computing the CUR gain their advantage by assuming, rather than discovering, low-rank structure of a matrix. In other words, while the SVD must compute all of the eigenvalues to determine their decay (and resulting matrix rank), algorithms for computing CUR decompositions often directly seek a factorization of a given rank. Intuitively, this constraint allows for computational gains. While the requirement of knowing the matrix rank ahead of time may seem like a barrier, rank adaptation schemes

exist. Furthermore, since our focus is reduced-order modeling, the desired/allowable model order is often prescribed *a priori*. This prescription typically follows engineering requirements and limitations with regards to available computational resources.

In Section 2 we review and introduce the notational setting. In Section 3, we show that any factorization of the Hankel matrix into two matrices with orthogonal columns, and rows, respectively, can be used in the ERA, and that the resulting reduced-order models remain stable. In Section 4, we integrate the CUR factorization into ERA-based system identification. We present associated error bounds and give the computational cost of the algorithm. The numerical study in Section 5 demonstrates that the algorithm enjoys speedups of two orders of magnitude when compared with the standard SVD-based ERA. Furthermore, our algorithm yields comparable results in terms of accuracy of the resulting reduced-order models.

Throughout this manuscript, we let $I_n \in \mathbb{R}^{n \times n}$ denote the identity matrix, and shall use I whenever there is not confusion about the dimension. Given a vector $x \in \mathbb{R}^n$ and $r \leq n$, the vector $x(1:r)$ denotes the first r components. Similarly, for a matrix $A \in \mathbb{R}^{n \times n}$, we denote by $A(1:r, 1:r)$ the leading $r \times r$ submatrix of A .

2. System Identification

The state-space form of a linear time-invariant (LTI) discrete-time system is described by the 4-tuple of matrices (A, B, C, D) , where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$ and $D \in \mathbb{R}^{p \times m}$ are state-to-state, state-to-input, state-to-output, and feed-through system matrices, respectively. The states $x_k \in \mathbb{R}^n$, the inputs $u_k \in \mathbb{R}^m$, and the outputs $y_k \in \mathbb{R}^p$ evolve according to¹

$$x_{k+1} = Ax_k + Bu_k, \quad (1)$$

$$y_k = Cx_k + Du_k, \quad (2)$$

where $k \in \mathbb{N}_0^+$ corresponds to time $t_k = k\Delta t$, for uniform sampling time Δt . The initial condition x_0 is assumed to be zero—the system is excited through external disturbances.

2.1. Hankel Matrix and Partial Realization

The *Markov parameters* $h_k \in \mathbb{R}^{p \times m}$ are defined according to

$$h_k := \begin{cases} D, & k = 0 \\ CA^{k-1}B, & k = 1, 2, \dots \end{cases}, \quad (3)$$

so the output response equation for system (1)–(2) is fully determined by the Markov parameters according to

$$y_k = \sum_{j=0}^k h_j u_{k-j}. \quad (4)$$

¹We follow the original ERA notation and assume a standard state-space, i.e., the mass matrix is the identity. This makes the notation involving Markov parameters and the Hankel matrix much simpler. The theory can be extended to the general case at the expense of heavier notation.

This equation is known as the external description of the system.

In practice, the matrices (A, B, C, D) are often unavailable. However, the Markov parameters can be obtained by observing the reaction of the system to “delta inputs”. In particular, we sample the outputs y_k when the i th input component is $[u_k]_i = 1$ at all sampling instances $k \in \mathbb{N}_0^+$, and the other inputs are set to zero. Doing this for each input channel $i = 1, \dots, p$ we can assemble the Markov parameters column-wise as

$$h_k(1:p, i) = y_k - y_{k-1}, \quad \text{for } [u_k]_i = 1.$$

Note, that this requires p simulations or experiments of the system. The *partial realization problem* addresses the inference of a LTI system from a finite sequence of Markov parameters.

Definition 2.1. [1, Definition 4.46] Given the finite set of Markov parameters matrices h_k , for $k = 1, 2, \dots, 2s - 1, s \in \mathbb{N}^+$, the partial realization problem consists of finding a positive integer n and constant matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, and $D \in \mathbb{R}^{p \times m}$, such that (3) holds.

The *Hankel matrix*, denoted by H , plays a key role in system identification, and it is constructed from the $2s - 1$ Markov parameters:

$$H := \begin{bmatrix} h_1 & h_2 & \dots & h_s \\ h_2 & h_3 & \dots & h_{s+1} \\ \vdots & \vdots & \ddots & \vdots \\ h_s & h_{s+1} & \dots & h_{2s-1} \end{bmatrix} \in \mathbb{R}^{n_r \times n_c}. \quad (5)$$

To simplify notation, let $n_r = ps$ be the row dimension of H , and $n_c = ms$ its column dimension. The Hankel matrix can become large ($\dim(H) > 10^5$)—and hence the SVD prohibitively expensive—due to one or more of the following three parameters:

1. **Input dimension m :** The number of input terminals can be large $m \approx \mathcal{O}(10-100)$ in applications where many actuators are installed.
2. **Output dimension p :** The number of outputs of a system can in some cases be of the size of the state space, which is often very large (e.g., fluid dynamical applications $p \geq \mathcal{O}(10^5)$).
3. **Sample size s :** For slowly decaying dynamics, and other practical reasons, the recorded sample size s can be large, $s \approx \mathcal{O}(10^3 - 10^5)$.

To motivate the system identification procedure, assume for a moment that the (A, B, C, D) matrices are known, so that the Hankel matrix reads as

$$H = \begin{bmatrix} CB & CAB & \dots & CA^{s-1}B \\ CAB & CA^2B & \dots & CA^sB \\ \vdots & \vdots & \ddots & \vdots \\ CA^{s-1}B & CA^sB & \dots & CA^{2s-1}B \end{bmatrix}.$$

It is well known (e.g., [1, Lemma 4.39]) that for a realizable impulse response sequence, the Hankel matrix can be factored into the product of the *observability matrix* and the *controllability matrix*:

$$H = \underbrace{\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{s-1} \end{bmatrix}}_{\text{Observ.}} \underbrace{[B \ AB \ \dots \ A^{s-1}B]}_{\text{Contr.}}. \quad (6)$$

Observe, that the shifted observability matrix satisfies

$$\begin{bmatrix} C \\ \vdots \\ CA^{s-2} \end{bmatrix} A = \begin{bmatrix} CA \\ \vdots \\ CA^{s-1} \end{bmatrix}. \quad (7)$$

If the observability matrix in (6) is available, then the above shift-invariance formula naturally leads to a least-squares problem for the matrix A , since everything except A is known. Truncating the last block row of the observability matrix gives the data for the left side of (7), and truncating the first block row yields the right-hand-side of (7).

The computational challenge is to factorize the Hankel matrix into its observability and controllability matrices by solving the least squares problem (7) with data provided by the Markov parameters. The common observation [1, 15, 13] that the Hankel matrix is of numerical low rank is a key enabler for many state-of-the-art rank-revealing matrix factorizations developed by the computer science and numerical linear algebra communities.

3. Factorization of Hankel Matrix

In this section, we show that *any* decomposition that provides an orthogonal basis for the range and column space for H can be used for the purpose of system identification. Furthermore, we show that the resulting system is stable. These facts enable using matrix factorizations other than the SVD for identification. We also require the following assumption about the convergence of the impulse response sequence, analogously to [14], in the proof of Theorem 3.2 to prove that the resulting system is stable.

Assumption 3.1. *Assume that $2s - 1$ Markov parameters are given and that the given impulse response sequence is convergent in the sense that*

$$h_i \rightarrow 0 \quad \text{for } i > s.$$

Clearly, for asymptotically stable dynamical systems, $h_i \rightarrow 0$ as $i \rightarrow \infty$. However, in the case of the finite data, this assumption means that $\|h_1\| \gg \|h_{s+1}\|$; in other words, the Markov parameters have decayed significantly after s steps.

3.1. ERA with General Factorization of H

This section describes the main steps of the ERA. For the sake of generality, let the Hankel matrix be approximated by a rank $r \leq \min(n_r, n_c)$ (recall $H \in \mathbb{R}^{n_r \times n_c}$, $n_r = ps$, $n_c = ms$) factorization of the form

$$H_r = \Theta \Gamma \approx H,$$

where $\Theta \in \mathbb{R}^{n_r \times r}$ and $\Gamma \in \mathbb{R}^{r \times n_c}$. Moreover, let the matrices Θ and Γ have orthogonal columns and rows, respectively, such that the Gramian matrices $\Theta^T \Theta = \Gamma \Gamma^T = S$, where $S = \text{diag}(s_1, \dots, s_r) \in \mathbb{R}^{r \times r}$, and $s_i \in \mathbb{R}^+$.

In light of (6), if Θ is the rank r reduced order observability matrix, then its first block row can be used to estimate a reduced state-to-output matrix $C_r \in \mathbb{R}^{p \times r}$ such that

$$C_r = [I_p \ 0] \Theta,$$

where I_p is the $p \times p$ identity matrix. Similarly, the first block column of Γ yields an r th order representation of the control input matrix $B_r \in \mathbb{R}^{r \times m}$ such that

$$B_r = \Gamma \begin{bmatrix} I_m \\ 0 \end{bmatrix}.$$

Let

$$\Theta^f = \Theta(1:(n_r - p), :), \quad \Theta^l = \Theta((p + 1):n_r, :) \quad (8)$$

denote the first $s - 1$ block rows, and the last $s - 1$ block rows of Θ , respectively. Define $\Gamma^f, \Gamma^l \in \mathbb{R}^{r \times (n_c - m)}$ analogously through truncation of the first and last block column of Γ . A rank r system matrix A_r is obtained by imposing the shift invariance property (7) on the rank r controllability and observability matrices, i.e.,

$$\Theta^f A_r = \Theta^l, \quad A_r \Gamma^f = \Gamma^l. \quad (9)$$

Either equality can be used to solve the least squares problem for A_r . Without loss of generality, we focus on the first equality in (9) involving the observability matrix. The solution of $\min_{A_r \in \mathbb{R}^{r \times r}} \|\Theta^f A_r - \Theta^l\|_F$ is obtained by using the Moore-Penrose pseudo-inverse [6, Ch. 5], denoted by $[\cdot]^\dagger$, given by

$$A_r = [\Theta^f]^\dagger \Theta^l.$$

3.2. Stability of system matrix

We now show that the resulting reduced-order model defined via A_r, B_r, C_r is a stable dynamical system.

Theorem 3.2. *Let Assumption 3.1 hold and let*

$$H_r = \Theta \Gamma,$$

with

$$\Theta^T \Theta = \Gamma \Gamma^T = S = \text{diag}(s_1, \dots, s_r), \quad s_i > 0$$

be a factorization of the Hankel matrix.² Let Θ^f, Θ^l be defined as in (8). Then the identified reduced-order model given by

$$A_r = [\Theta^f]^\dagger \Theta^l, \quad B_r = \Gamma [I_m \ 0]^T, \quad C_r = [I_p \ 0] \Theta, \quad (10)$$

is a stable discrete-time dynamical system.

Proof. To show stability of the discrete time dynamical system A_r, B_r, C_r we need to show that the spectrum of A_r is contained inside the unit circle in \mathbb{C} . First, recall that the spectral radius of a matrix, $\rho(A_r) = \|A_r\|_2$, is bounded by any induced matrix norm of A_r . To see this, let λ be any eigenvalue of A_r , and let $x \neq 0$ be its corresponding eigenvector, so that $A_r x = \lambda x$. Then,

$$|\lambda| \|x\| = |\lambda x| = |A_r x| \leq \|A_r\| \cdot \|x\|, \quad \Rightarrow \quad \rho(A_r) \leq \|A_r\|,$$

for any induced norm $\|\cdot\|$, since the inequality holds for all eigenvalues of A_r . Next, since we have $\Theta^T \Theta = S$, define $U_r = \Theta S^{-1/2} \in \mathbb{R}^{n_r \times r}$, so that $U_r^T U_r = I_r$. The orthonormal basis U_r can be divided into

$$U_r = \begin{bmatrix} U_r^f \\ u_{s-1} \end{bmatrix} \quad (11)$$

with $u_{s-1} \in \mathbb{R}^{p \times r}$ and $U_r^f \in \mathbb{R}^{(s-1)p \times r}$, and recall $n_r = ps$. Consequently,

$$I_r = U_r^T U_r = [U_r^f]^T U_r^f + u_{s-1}^T u_{s-1}. \quad (12)$$

We can then rewrite

$$A_r = [\Theta^f]^\dagger \Theta^l = S^{-1/2} [U_r^f]^\dagger U_r^l S^{1/2}.$$

To show stability, we consider the similarity transformation

$$S^{1/2} A_r S^{-1/2} = [U_r^f]^\dagger U_r^l, \quad (13)$$

which has the same eigenvalues than A_r . Therefore, the pseudo-inverse in (13) can be rewritten as

$$\begin{aligned} [U_r^f]^\dagger U_r^l &= \left([U_r^f]^T U_r^f \right)^{-1} [U_r^f]^T U_r^l, \\ &= [I_r - u_{s-1}^T u_{s-1}]^{-1} [U_r^f]^T U_r^l. \end{aligned}$$

Using the above, and the similarity transformation of A_r , we have

$$\begin{aligned} \|A_r\|_2 &= \|S^{1/2} A_r S^{-1/2}\|_2 = \|[U_r^f]^\dagger U_r^l\|_2 \\ &= \|[I_r - u_{s-1}^T u_{s-1}]^{-1} [U_r^f]^T U_r^l\|_2 \\ &\leq \|[U_r^f]^\dagger U_r^l\|_2 \\ &\leq \|U_r^f\|_2 \|U_r^l\|_2 \\ &\leq 1. \end{aligned}$$

²The entries s_i do *not* have to be the singular values of H . In the special case where $s_i = \sigma_i(H)$ are the leading r singular values of H , the factorization is the standard ERA from [14].

where the first two equalities are due to (13) and the third equality is due to (12). The first inequality follows from Assumption 3.1, and the last two inequalities are a simple application of Cauchy-Schwarz and the orthogonality of U_r . In sum, we showed that the spectral radius $\rho(A_r) = \|A_r\|_2 \leq 1$, which completes the proof. \square

Remark 3.3. Balanced truncation is inherently related to the eigensystem realization algorithm. We present ERA in a more general form though, not requiring that Θ and Γ are the optimal balancing modes, but only that they define principal-axis balancing transformations [1, Lemmata 4.39, 5.8, Def. 7.2]. This is important for showing stability of the system in Theorem 3.2 above. By not requiring the factorizations to come from an SVD approximation of the Hankel matrix, we lose optimality bounds with respect to the transfer function error [1, Thm. 7.10] and the “theoretical equivalence” notion to snapshot-based balanced truncation [15]. In fact, using the general factorizations Θ and Γ above within a balancing framework (when the matrices are available) leads to a sub-optimal error bound. Nevertheless, we show below that allowing for a general factorization can save much computational effort, and yield approximations with the same order-of-magnitude error as expensive SVD based methods.

4. CUR-based System Identification

In this section, we look at the specific case of obtaining the decomposition of the Hankel matrix via a CUR factorization. Furthermore, we provide theoretical bounds relating the approximation error of the CUR decomposition to the singular values of the matrix. These results are integral to stating an error bound for the approximate Markov parameters which are obtained by CUR-based system identification. The numerical results in Section 5 demonstrate that the errors are small in practice. Finally, we show that the advantages of the CUR factorization include storage requirements of $r^2 + (n_r + n_c)r$ elements, and computational complexity that scales linearly with the number of rows and columns of the Hankel matrix, and quadratically with its rank.

4.1. CUR Approximation of Hankel Matrix

Let $\mathcal{I} = \{i_1, i_2, \dots, i_r\}$ denote a set of row indices with $i_k \in \{1, 2, \dots, n_r\}$ and $\mathcal{J} = \{j_1, j_2, \dots, j_r\}$ denote a set of column indices with $j_k \in \{1, 2, \dots, n_c\}$. Then the *CUR factorization* of H is defined as

$$H_r := H(:, \mathcal{J})H(\mathcal{I}, \mathcal{J})^\dagger H(\mathcal{I}, :). \quad (14)$$

If H has finite rank $r < \min(n_r, n_c)$, i.e., there exists a rank r matrix $H_r^{opt} \in \mathbb{R}^{n_r \times n_c}$ such that $\|H_r^{opt} - H\|_F = 0$, then we can find \mathcal{I} and \mathcal{J} such that the resulting CUR factorization H_r in (14) is also exact [8].

In practice, however, matrices are often only *approximately* low-rank. In other words, they can be decomposed into the sum of a low-rank matrix and a matrix with small norm, $H = H_r + E$ such that $\|E\|_F \leq \epsilon$. In these cases, the indices \mathcal{I} and \mathcal{J} must be chosen such that the error of the approximation $\|H - H_r\|$ is small in some norm. Certain choices of

index sets lead to better approximations, and one ideal choice that we seek is to find a *submatrix* $H(\mathcal{I}, \mathcal{J})$ that has maximum volume, i.e., $[\mathcal{I}, \mathcal{J}] = \arg \max_{\hat{\mathcal{I}}, \hat{\mathcal{J}}} |\det(H(\hat{\mathcal{I}}, \hat{\mathcal{J}}))|$. This choice provides the following bounds according to [8, 9]:

$$\begin{aligned} \|H - H_r\|_C &\leq (r + 1)\sigma_{r+1}, \\ \|H - H_r\|_C &\leq (r + 1)^2 \min_{\text{rank}(X)=r} \|H - X\|_C, \end{aligned}$$

where $\|\cdot\|_C$ denotes the Chebyshev norm (maximum in absolute value element of the matrix) and σ_{r+1} denotes the $(r + 1)$ th singular value of the Hankel matrix.

In practice, finding a maximum volume submatrix is infeasible because it is a combinatorial optimization problem. Therefore, we make two approximations. First, we use the cross approximation scheme of Oseledets [17] that alternates between searching over rows and columns; this algorithm is reproduced below by Algorithm 1. The main idea of cross approximation is to sequentially increase the volume of a submatrix by alternating a search over rows and columns. The algorithm we use avoids visiting every element of H and is thus fast to compute. Similar maximal volume algorithms were applied recently in the Discrete Empirical Interpolation Framework [19, 4] to improve the interpolation point selection of nonlinear functions.

Algorithm 1 Cross approximation of a matrix [17]

Input: Matrix $H \in \mathbb{R}^{n_c \times n_r}$;

Rank upper bound estimate r ;

Initial column indices $\mathcal{J} = [j_1, j_2, \dots, j_r]$;

Stopping tolerance $\delta > 0$;

maxvol tolerance ϵ

Output: \mathcal{I}, \mathcal{J} such that $H(\mathcal{I}, \mathcal{J})$ has “large” volume

- 1: $k = 1$
 - 2: $Q, R = \text{qr}(H(:, \mathcal{J}))$
 - 3: $\mathcal{I} = \text{maxvol}(Q, \epsilon)$
 - 4: $Q, R = \text{qr}((H(\mathcal{I}, :))^T)$
 - 5: $\mathcal{J} = \text{maxvol}(Q, \epsilon)$
 - 6: $\hat{Q} = Q(\mathcal{J}, :)$
 - 7: $H_1 = H(:, \mathcal{J}) \left(Q\hat{Q}^{-1}\right)^T$
 - 8: **repeat**
 - 9: $Q, R = \text{qr}(H(:, \mathcal{J}))$
 - 10: $\mathcal{I} = \text{maxvol}(Q, \epsilon)$
 - 11: $Q, R = \text{qr}((H(\mathcal{I}, :))^T)$
 - 12: $\mathcal{J} = \text{maxvol}(Q, \epsilon)$
 - 13: $\hat{Q} = Q(\mathcal{J}, :)$
 - 14: $H_{k+1} = H(:, \mathcal{J}) \left(Q\hat{Q}^{-1}\right)^T$
 - 15: $k = k + 1$
 - 16: **until** $\|H_{k+1} - H_k\|_F / \|H_k\|_F \leq \delta$
-

The second approximation to seek a *dominant* submatrix instead of the maximum volume submatrix when searching over rows or columns. For a fixed set of rows (columns) Algorithm 2 is used to find a new set of columns (rows) such that the resulting submatrix is *dominant*³. These two algorithmic ingredients, cross approximation and seeking a dominant submatrix, allow us to *approximately* find the maximum-volume submatrix. Indeed, the volume of a dominant submatrix can bound the maximum volume submatrix. Suppose we are considering submatrices of size $r \times r$ of the matrix H , then according to [7], the following bounds are obtained

$$|\det(H_{\max\text{vol}})| \geq |\det(H_{\text{dom}})| \geq \frac{|\det(H_{\max\text{vol}})|}{r^{r/2}},$$

where $H_{\max\text{vol}}$ is a submatrix of H with maximum volume amongst all submatrices, and H_{dom} is any dominant submatrix of H . For more details we refer to [7]. Suppose that $|\det(H_{\text{dom}})| \geq \nu |\det(H_{\max\text{vol}})|$ for $\nu < 1$. Then the componentwise error in H is given in [8] as

$$\|H - H_r\|_C \leq \nu^{-1}(r+1)\sigma_{r+1}. \quad (15)$$

Algorithm 2 maxvol: Find dominant submatrix [7]

Input: Matrix $Q \in \mathbb{R}^{n_c \times r}$;

Convergence tolerance ϵ

Output: Row indices $\mathcal{I} = (i_1, i_2, \dots, i_r)$ such that $Q(\mathcal{I}, :)$ is approximately dominant;

- 1: $L, U, \mathcal{P} = \mathbf{1u}(Q)$ {pivoted LU decomposition; \mathcal{P} are pivots}
 - 2: $Q_{\mathcal{P}} = Q(\mathcal{P}, :)$
 - 3: $G = Q_{\mathcal{P}} [Q_{\mathcal{P}}(1:r, :)]^\dagger$
 - 4: $i^*, j^* = \arg \max_{i,j} |G(i, j)|$
 - 5: $g = G(i, j)$
 - 6: **while** $|g| > 1 + \epsilon$ **do**
 - 7: $\mathcal{P}(j^*) = i^*$
 - 8: $\mathcal{P}(i^*) = j^*$
 - 9: $Q_{\mathcal{P}} = Q(\mathcal{P}, :)$
 - 10: $G = Q [Q_{\mathcal{P}}(1:r, :)]^\dagger$
 - 11: $i^*, j^* = \arg \max_{i,j} |G(i, j)|$
 - 12: $g = G(i, j)$
 - 13: **end while**
 - 14: $\mathcal{I} = \mathcal{P}(1:r)$
-

4.2. Using CUR in the Eigensystem Realization Algorithm

After obtaining the CUR decomposition of H it is necessary to find an orthogonal basis for the row and column space. To this end, let the SVD of the submatrix $H(\mathcal{I}, \mathcal{J})$ be

³Let H be an $n \times r$ matrix with $n > r$ and let $\text{rank}(H) = r$. Specify an index set \mathcal{J} such that $|\mathcal{J}| = r$. The submatrix $H(\mathcal{J}, :)$ is dominant if $G = H[H(\mathcal{J}, :)]^\dagger$ has elements such that $\|G\|_C \leq 1$.

defined as

$$U_1 \Sigma_1 V_1^T = H(\mathcal{I}, \mathcal{J}),$$

and recall that it may be obtained with $\mathcal{O}(r^3)$ operations. Next, a basis for the column space is obtained by taking a QR decomposition of the combination of the columns found from Algorithm 1

$$Q_{col} R = H(:, \mathcal{J}),$$

using $\mathcal{O}(nr^2)$ operations. A basis for the row space is similarly obtained by taking the LQ decomposition of the rows

$$L Q_{row} = H(\mathcal{I}, :).$$

We can then form the matrix $R[H(\mathcal{I}, \mathcal{J})]^\dagger L = R V_1 \Sigma_1^{-1} U_1^T L$ by using the SVD of the submatrix. Taking the SVD of this matrix we obtain

$$U \Sigma V^T = R V_1 \Sigma_1^{-1} U_1^T L.$$

The approximation for the Hankel matrix then becomes

$$H \approx H_r = \underbrace{Q_{col} U \Sigma^{1/2}}_{=: \Theta} \Sigma^{1/2} V^T \underbrace{Q_{row}}_{=: \Gamma}$$

where, in analogy to (6) above, Θ, Γ are approximations of the observability and controllability matrices, respectively. A modified ERA with CUR-factored Hankel approximation is given by Algorithm 3⁴. We show as a consequence of Theorem 3.2 that the identified reduced-order model is stable, since the factorizations are have orthogonal columns, and rows, respectively, by construction.

Corollary 4.1. *Let Θ and Γ be computed as above via the CUR-ERA Algorithm 3. Then, the identified reduced-order model (10) is stable.*

4.3. Computational Complexity of the CUR-based ERA

The computational complexity of the CUR-based ERA algorithm is defined by the computation of the CUR decomposition. Algorithm 2, `maxvol`, requires the LU factorization of an $n_c \times r$ matrix. This factorization requires $\mathcal{O}(n_c r^2)$ operations. Furthermore, the dominant cost of each iteration is the multiplication of an $n_c \times r$ matrix and an $r \times r$ matrix, for a cost of $\mathcal{O}(n_c r^2)$ operations. If $c \in \mathbb{N}^+$ denotes the number of iterations performed, then the computational complexity of `maxvol` is $\mathcal{O}(c n_c r^2)$.

A conservative bound on c is

$$c \leq (\log|\det(H_{\maxvol})|) - \log|\det(H_{\text{start}})|) / \log(1 + \epsilon), \quad (16)$$

where H_{start} is the submatrix $Q_{\mathcal{P}}(1:r, :) \in \mathbb{R}^{r \times r}$ of Line 2 in Algorithm 2 [7].

We now provide the computational cost for cross approximation.

⁴The initial indices \mathcal{J} in Algorithm 3 are chosen from the uniform distribution on the discrete set of column indices.

Algorithm 3 CUR-ERA: CUR-based Eigensystem Realization Algorithm

Input: Markov parameters $h_1, h_2, \dots, h_{2s-1}$;

Reduced model order r ;

Tolerances ϵ, δ ;

Initial column indices \mathcal{J}

- 1: Assemble H from $\{h_i\}_{i=1, \dots, 2s-1}$ as in (2.1)
 - 2: $\mathcal{I}, \mathcal{J} = \text{cur-cross}(H, r, \mathcal{J}, \delta, \epsilon)$
 - 3: $Q_{col}R = \text{qr}(H(:, \mathcal{J}))$
 - 4: $LQ_{row} = \text{lq}(H(\mathcal{I}, :))$
 - 5: $U_1 \Sigma_1 V_1 = \text{svd}(H(\mathcal{I}, \mathcal{J}))$
 - 6: $U \Sigma V^T = \text{svd}(R V_1 \Sigma_1^{-1} U_1^T L)$
 - 7: $\hat{U} = Q_{col}U$ and $\hat{V} = V^T Q_{row}$
 - 8: $\hat{U}_1 = \hat{U}(1 : n_c - p, :)$ and $\hat{U}_2 = \hat{U}((p + 1) : n_c, :)$
 - 9: $A_r = \Sigma^{-1/2} \hat{U}_1^T \hat{U}_2 \Sigma^{1/2}$
 - 10: $B_r = \Sigma^{1/2} \hat{V}^T [I_m \ 0]^T$
 - 11: $C_r = [I_p \ 0] \hat{U} \Sigma^{1/2}$
-

Lemma 4.2. *Let $\kappa \in \mathbb{N}^+$ denote the number of iterations required for Algorithm 1, and let $c \in \mathbb{N}^+$ denote the maximum number of iterations required for `maxvol`. Then the total computational cost of Algorithm 1 is*

$$\mathcal{O}(\kappa r^3 + \kappa c(n_c + n_r)r^2).$$

Proof. The computational cost of Algorithm 1 is dominated by two QR decompositions, two calls to `maxvol` and the computation of the two determinants of an $r \times r$ matrix, at every iteration. The two QR decompositions are obtained for the $n_c \times r$ and $n_r \times r$ matrices, at a cost of $\mathcal{O}((n_c + n_r)r^2)$ overall. Obtaining the determinant of an $r \times r$ matrix is a $\mathcal{O}(r^3)$ operation. The cost of `maxvol` is obtained by calculating the cost of Algorithm 2, which has complexity of $\mathcal{O}(cn_c r^2)$ for the columns and $\mathcal{O}(cn_r r^2)$ for the rows. The computational complexity of computing the difference between the Hankel matrices of successive iterations, i.e., the stopping criterion in Line 16, is $\mathcal{O}((n_c + n_r)r^2)$ (see Appendix A). \square

The overall computational cost of the CUR-based system identification CUR-ERA is given by the following theorem.

Theorem 4.3. *Let $\kappa \in \mathbb{N}$ denote the number of iterations required for Algorithm 1, and let $c \in \mathbb{N}$ denote the maximum number of iterations required for `maxvol`. The computational complexity of CUR-ERA is*

$$\mathcal{O}(\kappa r^3 + \kappa c(n_c + n_r)r^2).$$

Proof. The computational complexity of CUR-ERA (Algorithm 3) for system identification is dominated by decomposing the Hankel matrix into a CUR approximation. From

Lemma 4.2 we have that the cross approximation costs $\mathcal{O}(\kappa r^3 + \kappa c(n_c + n_r)r^2)$. Additionally, two QR decompositions have to be computed, at cost $\mathcal{O}((n_c + n_r)r^2)$. The SVD of the small $r \times r$ matrix requires once more $\mathcal{O}(r^3)$ operations. The two matrix multiplications in Line 7 require also $(n_r + n_c)r^2$ computations. All other matrix multiplications are dominated by the above cost. Taken together, Algorithm 3 has the same complexity as Algorithm 1. \square

4.4. Error Bound for CUR-based System Identification

In this section, we establish a bound for the maximum approximation error in approximating the Markov parameters. The error bound below incorporates the first neglected singular value of the Hankel matrix.

Proposition 4.4. *Let $\{h_k\}_{k=1}^{2s-1}$ be the original sequence of Markov parameters, where $h_k \in \mathbb{R}^{p \times m}$ and let $\{C_r A_r^{k-1} B_r\}_{k=1}^{2s-1}$ be the r th order identified sequence via CUR-ERA in Algorithm 3. The worst-case error in approximating the Markov parameters is*

$$\max_{k \in \{1, \dots, 2s\}} \|h_k - C_r A_r^{k-1} B_r\|_F \leq \nu^{-1} m p (r+1)^2 \sigma_{r+1}(H)^2,$$

where ν is given in (15).

Proof. The bound follows from applying (15) to the block entry h_i in the approximate Hankel matrix H_r . \square

5. Numerical Results

In this section⁵ we numerically compare Algorithm 3, CUR-ERA, with Kung's standard SVD-ERA.

We consider a mass-spring-damper model from [11, Sec. 6]. The model is a system of ordinary differential equations of dimension $n = 1000$, and it is equivalent to 500 mass-spring-damper elements. All masses are $m_i = 4$, the spring constants are $k_i = 4$ and the damping coefficients are $c_i = 0.1$ for $i = 1, 2, \dots, 500 = n/2$. The states are the displacement and momentum of the masses, and the outputs are the velocities of some selected masses. The number of inputs is equal to the number of outputs, namely $m = p = 30$. We have access to $2s = 1000$ Markov parameters.

The resulting Hankel matrix is of size $15,000 \times 15,000$. As a first step for ERA, a low-rank factorization of order $r = 80$ is computed via SVD and CUR, respectively. The tolerances for the CUR decomposition were set to $\delta = 10^{-4}$ and $\epsilon = 2 \times 10^{-2}$ in Algorithm 1. The corresponding CPU times and relative Frobenius norm error in the approximation of the Hankel matrix are given in Table 1. Since the CUR decomposition is initialized randomly, we averaged the results over 20 test runs.

Stability is an important requirement for predictive modeling. For the discrete LTI systems in this work, stability involves the requirement that eigenvalues are contained

⁵The code for this section is available under GitHub: <https://github.com/bokramer/CURERA>.

Table 1: CPU times to compute SVD and CUR decomposition of Hankel matrix. Solved on a desktop computer with Intel®Core i7-3770 CPU @ 3.40GHz × 8, with 32GB of DDR3 RAM and MATLAB 2015b.

	SVD-ERA	CUR-ERA
CPU time	1216.8s	47.72s
$\ H - H_r\ _F / \ H\ _F$	1.56×10^{-5}	4.35×10^{-5}

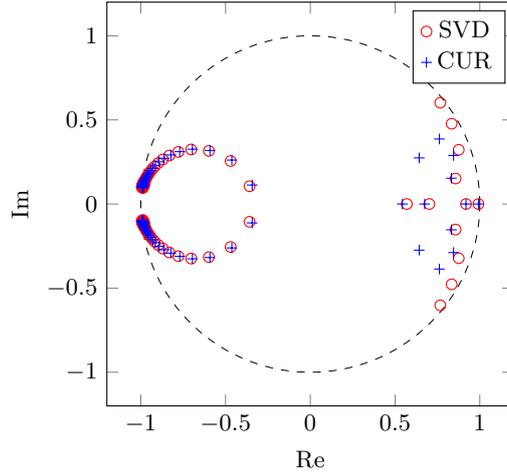


Figure 1: Eigenvalues of the identified system matrix A_r via CUR-ERA and SVD-ERA, respectively.

within unit circle in the complex plane \mathbb{C} . Recall, that the CUR-ERA guarantees stability by Theorem 3.2. Figure 1 shows the eigenvalues of the identified reduced-order matrices A_r from CUR-ERA and SVD-ERA.

Figure 2 demonstrates that the eigenvalues close to the critical unit circle – which are most important for system stability – are accurately approximated. Let λ_{SVD} denote the eigenvalue of the reduced order matrix A_r obtained from the SVD-ERA algorithm. Let λ_{CUR} denote the eigenvalue of the corresponding matrix obtained from using the CUR-ERA algorithm. The figure plots $|\lambda_{SVD} - \lambda_{CUR}| / |\lambda_{SVD}|$ against the difference from the eigenvalues to the stability boundary $|1 - |\lambda_{CUR}||$. Eigenvalues close to the stability boundary are approximated within 10^{-6} in accuracy, whereas eigenvalues further away from the boundary of the unit circle have larger errors.

Next, we look at the identified, reduced-order transfer functions

$$G_r(\omega) := C_r(\omega I - A_r)^{-1} B_r, \quad \omega \in \mathbb{C},$$

for both approaches, and we compare them with the full-order model transfer function

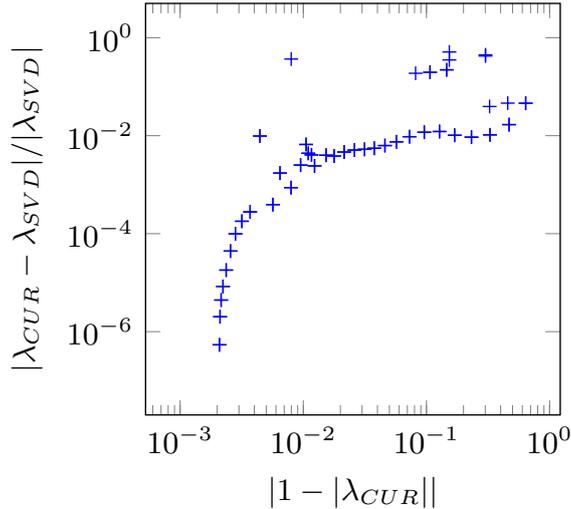


Figure 2: Relative accuracy of the eigenvalues $|\lambda_{SVD} - \lambda_{CUR}|/|\lambda_{SVD}|$ with respect to distance from the stability boundary $|1 - |\lambda_{CUR}||$. The relative accuracy of the largest (in magnitude) eigenvalues is significantly higher than for eigenvalues close to the origin, as can also be seen in Figure 1.

$G(\omega) := C(\omega I - A)^{-1}B$, $\omega \in \mathbb{C}$. Note that the transfer function $G(\cdot)$ is computed in these test problems for comparison purposes, but is, of course, not available in practice.

Figure 3 shows this comparison, whereas Figure 4 shows an error Bode plot. The errors in the transfer function are almost identical in the critical region of resonance $\omega \approx 2Hz$, yet both fail to approximate the spike fully. This error behavior comes from the fact, that system identification via Markov parameters is equivalent to sampling the transfer function at infinite frequency, and hence one cannot expect interpolation at finite frequencies. Rational interpolation based model reduction schemes would perform superior here, but one needs to be able to sample the transfer function $G(\cdot)$ at chosen frequencies.

5 To compare the predictive capabilities of the reduced models from SVD-ERA and CUR-ERA, we simulate the reduced systems as well as the full model, from zero to 50s with zero initial conditions. The input vector $u(t) \in \mathbb{R}^m$ is chosen as in [11, Ex.6.3] with equal components in all input channels as $e^{-0.05t} \sin(5t)$. Figures 5 and 6 show the sixth and eleventh outputs. Both plots demonstrate that the models predict the same qualitative behavior, and the computationally much faster CUR-ERA system identification matches the traditional SVD-ERA method extremely well.

6. Conclusions

We have shown that using the CUR decomposition together with the classic eigensystem realization algorithm leads to the inference of accurate reduced-order models from

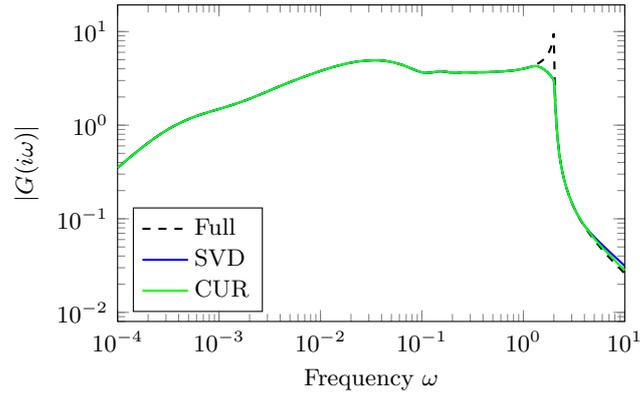


Figure 3: Transfer function of the full model and the two reduced-order models identified from CUR-ERA and SVD-ERA.

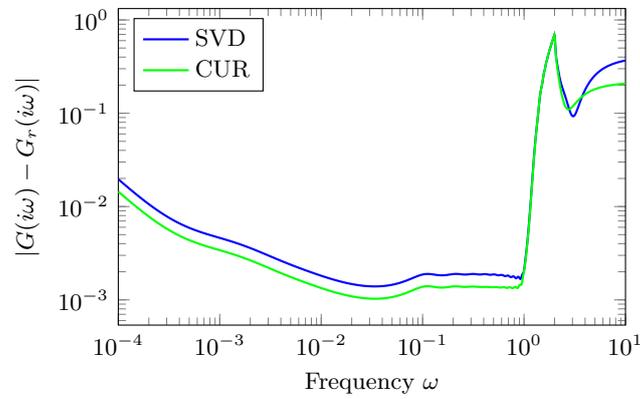


Figure 4: Error in transfer function from Figure 3.

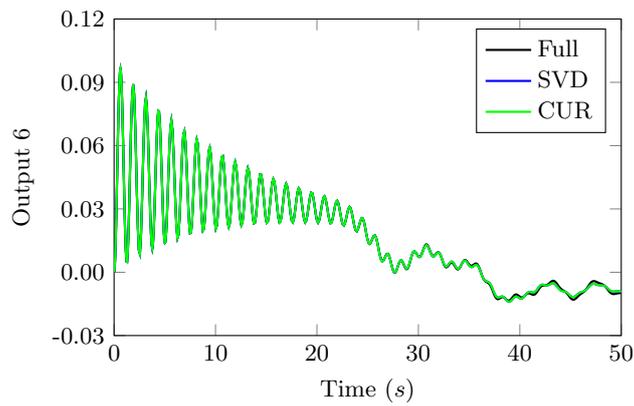


Figure 5: Output No.6 of the continuous-time systems realized using the full and the two identified reduced-order models.

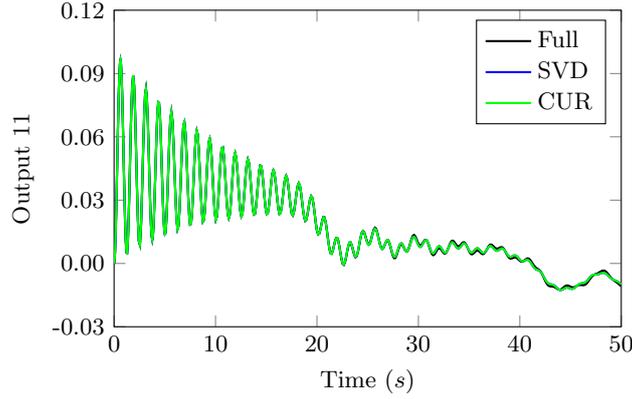


Figure 6: Output No.11 of the continuous-time systems realized using the full and the two identified reduced-order models.

the data-stream of Markov parameters. We showed that generic factorizations with orthogonal columns and rows, respectively, of the Hankel matrix can be used within the ERA framework. Furthermore, we were able to achieve a speedup of 25 compared with the SVD-based ERA, an algorithm that is widely used in engineering practice. At the same time, the approximation error of the Hankel matrix via CUR remained in the same order-of-magnitude than the optimal rank r SVD approximation.

A. Fast difference between CUR factorized matrices

Let $H, \hat{H} \in \mathbb{R}^{n_r \times n_c}$, $C_1, C_2 \in \mathbb{R}^{n_r \times r}$; $R_1, R_2 \in \mathbb{R}^{r \times n_c}$. Let $H = C_1 R_1$ represent the CUR factorization of H where $C_1 = H(:, \mathcal{J})H(\mathcal{I}, \mathcal{J})^{-1}$ is the product of the column matrix and the submatrix of the CUR. The decomposition requires $\mathcal{O}(n_c r^2)$ operations. Let $\hat{H} = C_2 R_2$ denote another matrix. A fast computation of $\|H - \hat{H}\|_F$ can be obtained by rewriting the difference as

$$\begin{aligned}
 M &= H - \hat{H} \\
 &= C_1 R_1 - C_2 R_2 \\
 &= \begin{bmatrix} C_1 & -C_2 \end{bmatrix} \begin{bmatrix} R_1 \\ R_2 \end{bmatrix} \\
 &= CR.
 \end{aligned}$$

Note that $C \in \mathbb{R}^{n_r \times 2r}$ and $R \in \mathbb{R}^{2r \times n_c}$. In practice, these matrices do not have to be explicitly formed, and, therefore, do not impact the computational expense of the algorithm. Let the elementwise product be defined as

$$E = M \odot M \implies E[i, j] = (M[i, j])(M[i, j]). \quad (17)$$

Then

$$\begin{aligned}
E[i, j] &= C[i, :]R[:, j]C[i, :]R[:, j] \\
&= C[i, :]R[:, j] \otimes C[i, :]R[:, j] \\
&= [C[i, :] \otimes C[i, :]] [R[:, j] \otimes R[:, j]].
\end{aligned}$$

The Kronecker product of the rows $C[i, :]$ requires $(2r)^2 = 4r^2$ operations. Similarly, the Kronecker product of the columns $R[:, j]$ requires $4r^2$ operations. Define

$$\begin{aligned}
\langle M, M \rangle &= \sum_{i,j=1}^{n_r, n_c} E[i, j] \\
&= \sum_{i,j=1}^{n_r, n_c} [C[i, :] \otimes C[i, :]] [R[:, j] \otimes R[:, j]] \\
&= \left[\sum_{i=1}^{n_r} [C[i, :] \otimes C[i, :]] \right] \left[\sum_{j=1}^{n_c} [R[:, j] \otimes R[:, j]] \right] \\
&= \Gamma_1 \Gamma_2.
\end{aligned}$$

Computing Γ_1 requires summing the Kronecker products of n_r row vectors of C , which requires $2n_r r$ operations. Computing n_r Kronecker products of C requires $4n_r r^2$ operations. Thus the total number of operations for computing Γ_1 is $4n_r r^2 + 2n_r r$. Similarly the cost of computing Γ_2 is $4n_c r^2 + 2n_c r$. Note that $\Gamma_1 \in \mathbb{R}^{1 \times 2r}$ and $\Gamma_2 \in \mathbb{R}^{2r \times 1}$, thus computing their product requires $4r$ operations. The complexity of computing $\langle M, M \rangle$ is then

$$\mathcal{O}((n_r + n_c)r^2), \quad (18)$$

i.e., linear scaling in the number of columns and rows of the matrix H instead of the usual quadratic complexity of computing the Frobenius norm. We obtain the desired result as

$$\|M\|_F = \sqrt{\langle M, M \rangle}.$$

Acknowledgments

The authors thank Professors Karen Willcox and Serkan Gugercin for helpful comments and suggestions.

References

- [1] A. C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. Advances in Design and Control. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2005.

- [2] H. Cheng, Z. Gimbutas, P.-G. Martinsson, and V. Rokhlin. On the compression of low rank matrices. *SIAM Journal on Scientific Computing*, 26(4):1389–1404, 2005.
- [3] P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2):844–881, 2008.
- [4] Z. Drmac and S. Gugercin. A new selection operator for the discrete empirical interpolation method—improved a priori error bound and extensions. *SIAM Journal on Scientific Computing*, 38(2):A631–A648, 2016.
- [5] G. Golub and C. Reinsch. Singular value decomposition and least squares solutions. *Numerische mathematik*, 14(5):403–420, 1970.
- [6] G. Golub and C. F. Van Loan. Matrix computations. *Johns Hopkins University, Press, Baltimore, MD, USA*, pages 374–426, 1996.
- [7] S. Goreinov, I. Oseledets, D. Savostyanov, E. Tyrtyshnikov, and N. Zamarashkin. How to find a good submatrix. *Matrix methods: theory, algorithms and applications*, pages 247–256, 2010.
- [8] S. Goreinov and E. Tyrtyshnikov. The maximal-volume concept in approximation by low-rank matrices. *Contemporary Mathematics*, 280:47–52, 2001.
- [9] S. Goreinov and E. Tyrtyshnikov. Quasioptimality of skeleton approximation of a matrix in the Chebyshev norm. In *Doklady Mathematics*, volume 83, pages 374–375. Springer, 2011.
- [10] S. A. Goreinov, E. Tyrtyshnikov, and N. Zamarashkin. A theory of pseudoskeleton approximations. *Linear Algebra and its Applications*, 261(1):1–21, 1997.
- [11] S. Gugercin, R. Polyuga, C. Beattie, and A. Van Der Schaft. Structure-preserving tangential interpolation for model reduction of port-Hamiltonian systems. *Automatica*, 48(9):1963–1974, 2012.
- [12] J.-N. Juang and R. Pappa. An eigensystem realization algorithm for modal parameter identification and model reduction. *Journal of Guidance, Control, and Dynamics*, 8(5):620–627, 1985.
- [13] B. Kramer and S. Gugercin. Tangential interpolation-based eigensystem realization algorithm for mimo systems. *Mathematical and Computer Modelling of Dynamical Systems*, 22(4):282–306, 2016.
- [14] S.-Y. Kung. A new identification and model reduction algorithm via singular value decomposition. In *Proc. 12th Asilomar Conf. Circuits, Syst. Comput., Pacific Grove, CA*, pages 705–714, 1978.

- [15] Z. Ma, S. Ahuja, and C. W. Rowley. Reduced-order models for control of fluids using the eigensystem realization algorithm. *Theoretical and Computational Fluid Dynamics*, 25(1-4):233–247, 2011.
- [16] M. W. Mahoney and P. Drineas. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009.
- [17] I. Oseledets and E. Tyrtyshnikov. TT-cross approximation for multidimensional arrays. *Linear Algebra and its Applications*, 432(1):70–88, 2010.
- [18] V. Rokhlin, A. Szlam, and M. Tygert. A randomized algorithm for principal component analysis. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1100–1124, 2009.
- [19] D. Sorensen and M. Embree. A DEIM induced CUR factorization. *SIAM Journal on Scientific Computing*, 38(3):A1454–A1482, 2016.