SIAM J. SCI. COMPUT.
Vol. 40, No. 2, pp. A848–A866

SYSTEM IDENTIFICATION VIA CUR-FACTORED HANKEL APPROXIMATION*

BORIS KRAMER† AND ALEX A. GORODETSKY‡

Abstract. 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 used 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 of the ERA from cubic to linear, with respect to the Hankel matrix size. Furthermore, our memory requirements scale at the same rate because we never require loading the entire Hankel matrix into memory. These reductions are 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 using a maximum-volume–based cross-approximation scheme that selects a small number of rows and columns to form the row and column space of the approximation. 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.

Key words. CUR decomposition, system identification, model reduction, eigensystem realization algorithm, Hankel matrix

AMS subject classifications. 93B15, 93C05, 65F99, 37E04

DOI. 10.1137/17M1137632

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] that can be used for design, optimization, and control. Building on system-theoretic concepts, 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 antidiagonal are equal.

Historically, the field of system identification from large-scale data 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 [6], Kung [18] in 1978 proposed a system identification algorithm by computing an approximate decomposition of the Hankel matrix through an SVD. A version of this algorithm was later termed the "eigensystem realization algorithm" (ERA) in [16], and it became popular in engineering due to its simplicity and guaranteed stability. The method uses

*Submitted to the journal's Methods and Algorithms for Scientific Computing section July 6, 2017; accepted for publication (in revised form) January 19, 2018; published electronically March 15, 2018.

http://www.siam.org/journals/sisc/40-2/M113763.html

Funding: The first author was supported by the DARPA EQUiPS program award UTA15-001067 (program manager F. Fahroo). The research of the second author was supported by the National Science Foundation through grant IIS-1452019.

†Corresponding author. Massachusetts Institute of Technology, Cambridge, MA 02125 (bokramer@mit.edu, http://web.mit.edu/bokramer/www/).

‡Aerospace Engineering, University of Michigan, Ann Arbor, MI (goroda@umich.edu, www.alexgorodetsky.com).

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 [17], 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 the 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 [11, 2, 4, 22, 20] 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 [23]. 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 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 regard to available computational resources.

Furthermore, we utilize a maximum-volume–based cross-approximation algorithm [9, 10, 21] to compute the CUR decomposition. In contrast to methods that require access to all matrix elements, e.g., methods based on leverage scores [20], this approach does not require even a single pass through all matrix columns. Instead, it provides a deterministic procedure for sampling groups of r columns and rows in order to progressively increase the volume of the matrix formed by their intersection. As such, its complexity scales linearly with the size of the matrix instead of quadratically.

System identification with large data is an active area of research. Hokanson [14] proposes exponential fitting algorithms—determining growth and frequency parameters for exponential signals—for large-scale applications with $\mathcal{O}(n \log n)$ complexity, where n is the number of collected data. Exponential fitting can be viewed as a special case of the partial realization problem introduced below. Antoulas and Ionita [15] derive a linear system representation from data by using the matrix pencil approach, which is easy to implement and works with data both in time- and frequency-domain. However, the matrix pencil approach does not guarantee stability of the resulting model.

Finally, we emphasize that this work is purely data-driven. Thus, in contrast to projection-based model reduction [1], the discussed algorithms do not need access to the mathematical model or computer code. The stability of projection-based meth-

ods has been proved, e.g., by Gugercin and Antoulas [12], for reduced-order models obtained via a mixture of Krylov-based and Gramian-based model reduction. In this work, we obtain a complementary result on stability in the data-driven setting.

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 paper, we let $I_n \in \mathbb{R}^{n\times n}$ denote the identity matrix and shall use I whenever there is no 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 to[1]

$$x_{k+1} = Ax_k + Bu_k, \tag{1}$$

$$y_k = Cx_k + Du_k, \tag{2}$$

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

$$h_k := \begin{Bmatrix} D, & k = 0, \\ CA^{k-1}B, & k = 1, 2, \dots \end{Bmatrix}, \tag{3}$$

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

$$y_k = \sum_{j=0}^{k} h_j \, u_{k-j}. \tag{4}$$

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 ith 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

[1] We 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 a general nonsingular mass matrix at the expense of heavier notation.

each input channel $i = 1, \ldots, m$, we can assemble the Markov parameters columnwise as

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

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

DEFINITION 1 (see [1, Definition 4.46]). *Given the finite set of Markov parameter matrices h_k for $k = 1, 2, \ldots, 2s-1$, $s \in \mathbb{N}^+$, 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:

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

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

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

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*:

$$H = \underbrace{\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{s-1} \end{bmatrix}}_{Observ.} \underbrace{\begin{bmatrix} B & AB & \ldots & A^{s-1}B \end{bmatrix}}_{Contr.}. \tag{6}$$

Observe that the shifted observability matrix satisfies

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

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, 19, 17] 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. In particular, we are able to remove the requirement in [18] that the factorization be in the form of an SVD. Furthermore, we show that the resulting system is stable. These facts enable using matrix factorizations that are faster and more efficient than the SVD.

3.1. ERA with general factorization of $\boldsymbol{H}$. 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 satisfy $\Theta^T\Theta = \Gamma\Gamma^T = S$, where $S = \mathrm{diag}(s_1, \ldots, s_r) \in \mathbb{R}^{r \times r}$, 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 rth order representation of the control input matrix $B_r \in \mathbb{R}^{r \times m}$ such that

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

Denote the first $s-1$ block rows and the last $s-1$ block rows of Θ, respectively, by

$$\Theta^f := \Theta(1\colon(n_r - p), :), \quad \Theta^l := \Theta((p+1)\colon n_r, :). \tag{8}$$

A rank r system matrix A_r is obtained by imposing the shift invariance property (7) on the rank r observability matrix, i.e.,

$$\Theta^f A_r = \Theta^l. \tag{9}$$

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 pseudoinverse [7, 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. To do so, we make an assumption on the impulse response data, which is the same as in Kung [18].

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

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

Clearly, for asymptotically stable dynamical systems, $h_i \to 0$ as $i \to \infty$. However, in the case of the finite data this assumption means that after some finite time, the Markov parameters begin to decay.

THEOREM 3. *Let*

$$H_r = \Theta\Gamma, \quad H_r \in \mathbb{R}^{ps \times ms}$$

with

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

be a factorization of the Hankel matrix.[2] *Let Θ^f, Θ^l be defined as in* (8). *Let Assumption* 2 *hold; then there exists an $N \in \mathbb{N}^+$ such that for $s > N$, the identified reduced-order model given by*

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

is a stable discrete-time dynamical system.

To prove this theorem, we first provide several intermediate results. Define $U_r := \Theta S^{-1/2}$ and $V_r^T := S^{-1/2}\Gamma$ so that $H_r = U_r S V_r^T$, and $U_r^T U_r = V_r^T V_r = I_r$ with $\|U_r\|_2 = 1$. We partition U_r into two different arrangements,

$$U_r = \begin{bmatrix} U_r^f \\ \eta_{s-1} \end{bmatrix} = \begin{bmatrix} \eta_1 \\ U_r^l \end{bmatrix}, \tag{11}$$

where $U_r^f, U_r^l \in \mathbb{R}^{(n_r-p)\times r}$ and $\eta_1, \eta_{s-1} \in \mathbb{R}^{p\times r}$. By construction, $U_r \in \mathbb{R}^{n_r \times r}$ is orthogonal with $r \ll n_r$; therefore, there is a selection of r rows of U_r which are linearly independent. Thus, we can say without loss of generality that U_r^f has (column) rank r.[3]

LEMMA 4. *If U_r^f is constructed as in* (11) *and Theorem* 3, *then the following holds for the Moore–Penrose pseudoinverse:*

$$[U_r^f S^{1/2}]^\dagger = S^{-1/2}[U_r^f]^\dagger. \tag{12}$$

Proof. From [3, Thm. 2.2] we have that for two matrices Y, Z of suitable dimension, $(YZ)^\dagger = Z^\dagger Y^\dagger$ if and only if the range (column) spaces $\mathcal{R}(Y^*YZ) \subseteq \mathcal{R}(Z)$ and $\mathcal{R}(ZZ^*Y^*) \subseteq \mathcal{R}(Y^*)$. Let $Y = U_r^f$ and $Z = S^{1/2}$, and since they are real matrices, we have that $Y^* = Y^T$, $Z^* = Z^T$. Therefore,

$$\mathcal{R}(Y^TYZ) = \mathcal{R}([U_r^f]^T U_r^f S^{1/2}) \subseteq \mathcal{R}(S^{1/2}) = \mathcal{R}(Z). \tag{13}$$

[2]The 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 [18].

[3]Strictly speaking, we can replace U_r^f with $\mathcal{S}U_r \in \mathbb{R}^{n_r-p\times r}$, where $\mathcal{S}$ is a full-rank operator that selects $n_r - p$ rows of U_r^f and has full column rank (similarly for U_r^l), so that the shift-invariance property (7) still holds. This operator exists due to the rank-r property of U_r.

Note that with $S^{1/2}$ being diagonal with positive diagonal elements, the column space $\mathcal{R}(S^{1/2}) = \mathbb{R}^r$. Furthermore,

$$\mathcal{R}(ZZ^TY^T) = \mathcal{R}(S[U_r^f]^T) = \mathcal{R}([U_r^f]^T) = \mathcal{R}(Y^T), \tag{14}$$

because S is diagonal with positive diagonal elements. Since both conditions are met, we obtain our stated result. Alternatively, one could use the fact [7, p. 257] that for a matrix X with full column rank, the pseudoinverse is given via $[X^TX]^{-1}X^T$. Applying this to $X = U_r^f S^{1/2}$ which has full column rank gives the stated result. ∎

LEMMA 5. *Under Assumption* 2 *and with* U_r *segmented as in* (11) *we have*

$$\lim_{s\to\infty} \frac{\|U_r^f\|_2 \|U_r^l\|_2}{1 - \|\eta_{s-1}\|_2^2} < 1.$$

Proof. Consider the SVD $H = U\hat{S}V^T$, so that

$$HV\hat{S}^{-1} = U = [U_r \ *] = \begin{bmatrix} U_r^f & * \\ \eta_{s-1} & * \end{bmatrix}.$$

From (11) we see that η_{s-1} is defined as the last p rows of the U_r matrix, and from (5) we see that $[h_s, \dots, h_{2s-1}]$ is the last p rows of H. Therefore, the first equality above implies $[h_s, \dots, h_{2s-1}]V\hat{S}^{-1} = [\eta_{s-1} \ *]$. Thus, $\lim_{s\to\infty} \|[\eta_{s-1} \ *]\|_2 = \lim_{s\to\infty} \|[h_s, \dots, h_{2s-1}]V\hat{S}^{-1}\|_2$, and as a consequence of Assumption 2 this limit is zero. In particular, $\lim_{s\to\infty} \|\eta_{s-1}\|_2 = 0$, so that $\lim_{s\to\infty} \|U_r^f\|_2 = \lim_{s\to\infty} \|U_r\|_2 = 1$ and $\|U_r^l\|_2 < 1$. Putting these together, we obtain the stated result. ∎

Proof of Theorem 3. To show stability of the discrete-time dynamical system (10), we need to show that all eigenvalues of A_r lie inside the unit circle in $\mathbb{C}$. First, recall that the spectral radius of a matrix, $\rho(A_r) := \max\{|\lambda_1(A_r)|, \dots, |\lambda_r(A_r)|\}$, is bounded by any induced matrix norm of A_r, i.e., $\rho(A_r) \le \|A_r\|$. From (11) we get

$$I_r = U_r^T U_r = [U_r^f]^T U_r^f + \eta_{s-1}^T \eta_{s-1}. \tag{15}$$

Since $\Theta = U_r S^{1/2}$, it follows that $\Theta^f = U_r^f S^{1/2}$ and $\Theta^l = U_r^l S^{1/2}$, and we can rewrite

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

where we used Lemma 4 for the reverse-order law of the pseudoinverse product. With (15) and the above observation that U_r^f has full column rank, we obtain

$$[U_r^f]^\dagger U_r^l = \left([U_r^f]^T U_r^f\right)^{-1} [U_r^f]^T U_r^l = [I_r - \eta_{s-1}^T \eta_{s-1}]^{-1} [U_r^f]^T U_r^l.$$

To show stability, it suffices to consider the eigenvalues of

$$S^{1/2} A_r S^{-1/2} = [U_r^f]^\dagger U_r^l, \tag{16}$$

which, by a similarity transformation, has the same eigenvalues as A_r. Thus,

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

For the last inequality, we used a bound based on the Neumann series [7, Lemma 2.3.3] since, as a submatrix, $\|\eta_{s-1}\|_2 < \|U_r\|_2 = 1$. With the above observation that the spectral radius of a matrix is bounded by any induced matrix norm, we have shown that $\rho(S^{1/2}A_rS^{-1/2}) \leq \frac{\|U_r^f\|_2\|U_r^l\|_2}{1-\|\eta_{s-1}\|_2^2}$. Since A_r and $S^{1/2}A_rS^{-1/2}$ have the same eigenvalues, we therefore also have that $\rho(A_r) \leq \frac{\|U_r^f\|_2\|U_r^l\|_2}{1-\|\eta_{s-1}\|_2^2}$. Using Lemma 5 and the fact that the above inequality holds for all s, we obtain

$$\rho(A_r) \leq \lim_{s\to\infty} \frac{\|U_r^f\|_2\|U_r^l\|_2}{1-\|\eta_{s-1}\|_2^2} < 1,$$

which implies our stated result and completes the proof. ∎

Remark 6. Balanced truncation is inherently related to the eigensystem realization algorithm. We present the 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, Lemmas 4.39 and 5.8; Def. 7.2]. This is important for showing stability of the system in Theorem 3 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 [19]. In fact, using the general factorizations Θ and Γ above within a balancing framework (when the matrices are available) leads to a suboptimal 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 $\mathcal{I} = \{i_1, i_2, \ldots, i_r\}$ denote a set of row indices with $i_k \in \{1, 2, \ldots, n_r\}$ and $\mathcal{J} = \{j_1, j_2, \ldots, j_r\}$ denote a set of column indices with $j_k \in \{1, 2, \ldots, n_c\}$. Then the *CUR factorization* of H is defined as

$$H_r := H(:, \mathcal{J})H(\mathcal{I}, \mathcal{J})^\dagger H(\mathcal{I}, :). \tag{17}$$

If H has finite rank $r < \min(n_r, n_c)$, then we can find $\mathcal{I}$ and $\mathcal{J}$ such that the resulting CUR factorization H_r in (17) is also exact [9].

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 a *submatrix* $H(\mathcal{I}, \mathcal{J})$ that has maximum volume,

i.e., $[\mathcal{I}, \mathcal{J}] = \arg\max_{\hat{\mathcal{I}},\hat{\mathcal{J}}} |\det(H(\hat{\mathcal{I}}, \hat{\mathcal{J}}))|$. If the singular values σ_i of H are ordered decreasingly, then this choice provides the following bounds according to [9, 10]:

$$\|H - H_r\|_C \leq (r+1)\sigma_{r+1},$$
$$\|H - H_r\|_C \leq (r+1)^2 \min_{\mathrm{rank}(X)=r} \|H - X\|_C,$$

where $\|\cdot\|_C$ denotes the Chebyshev norm (maximum in absolute value element of the 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 and Tyrtyshnikov [21] 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 [23, 5] to improve the interpolation point selection of nonlinear functions.

Algorithm 1. `cur-cross`: Cross-approximation of a matrix [21].

Input: Matrix $H \in \mathbb{R}^{n_c \times n_r}$;
Rank upper bound estimate r;
Initial column indices $\mathcal{J} = [j_1, j_2, \ldots, j_r]$;
Stopping tolerance $\delta > 0$;
`maxvol` tolerance ϵ

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

$k = 1$
$Q, R = \mathtt{qr}(H(:, \mathcal{J}))$
$\mathcal{I} = \mathtt{maxvol}(Q, \epsilon)$
$Q, R = \mathtt{qr}((H(\mathcal{I}, :))^T)$
$\mathcal{J} = \mathtt{maxvol}(Q, \epsilon)$
$\hat{Q} = Q(\mathcal{J}, :)$
$H_1 = H(:, \mathcal{J}) \left(Q\hat{Q}^{-1}\right)^T$
repeat
  $Q, R = \mathtt{qr}(H(:, \mathcal{J}))$
  $\mathcal{I} = \mathtt{maxvol}(Q, \epsilon)$
  $Q, R = \mathtt{qr}((H(\mathcal{I}, :))^T)$
  $\mathcal{J} = \mathtt{maxvol}(Q, \epsilon)$
  $\hat{Q} = Q(\mathcal{J}, :)$
  $H_{k+1} = H(:, \mathcal{J}) \left(Q\hat{Q}^{-1}\right)^T$
  $k = k + 1$
until $\|H_{k+1} - H_k\|_F / \|H_k\|_F \leq \delta$

The second approximation seeks 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 result-

ing submatrix is *dominant*.[4] 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 [8], the following bounds are obtained:

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

where H_{maxvol} 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 the reader to [8]. Suppose that $|\det(H_{\text{dom}})| \geq \nu|\det(H_{\text{maxvol}})|$ for $\nu < 1$. Then the componentwise error in H is given in [9] as

$$\|H - H_r\|_C \leq \nu^{-1}(r+1)\sigma_{r+1}. \tag{18}$$

Algorithm 2. `maxvol`: Find dominant submatrix [8].

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

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

$L, U, \mathcal{P} = \texttt{lu}(Q)$ {pivoted LU decomposition; $\mathcal{P}$ is a permutation vector}
$Q_{\mathcal{P}} = Q(\mathcal{P}, :)$
$G = Q_{\mathcal{P}}\,[Q_{\mathcal{P}}(1{:}r, :)]^{\dagger}$
$i^*, j^* = \arg\max_{i,j} |G(i,j)|$
$g = G(i^*, j^*)$
while $|g| > 1 + \epsilon$ **do**
  $\mathcal{P}(j^*) = i^*$
  $\mathcal{P}(i^*) = j^*$
  $Q_{\mathcal{P}} = Q(\mathcal{P}, :)$
  $G = Q_{\mathcal{P}}\,[Q_{\mathcal{P}}(1{:}r, :)]^{\dagger}$
  $i^*, j^* = \arg\max_{i,j} |G(i,j)|$
  $g = G(i^*, j^*)$
end while
$\mathcal{I} = \mathcal{P}(1{:}r)$

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 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,

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

[4] Let 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 $\|G\|_C \leq 1$.

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

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

We can then form the matrix $R\,[H(\mathcal{I}, \mathcal{J})]^{\dagger}\, L = RV_1\Sigma_1^{-1}U_1^T L$ by using the SVD of the submatrix. Taking the SVD of this matrix, we obtain

$$U\Sigma V^T = RV_1\Sigma_1^{-1}U_1^T L.$$

The approximation for the Hankel matrix then becomes

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

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 3.[5] We show as a consequence of Theorem 3 that the identified reduced-order model is stable, since the factorizations have orthogonal columns and rows, respectively, by construction.

COROLLARY 7. *Let* Θ *and* Γ *be computed as above via the* `CUR-ERA` *Algorithm* 3. *Then, the identified reduced-order model* (10) *is stable.*

Algorithm 3. `CUR-ERA`: CUR-based eigensystem realization algorithm.

Input: Markov parameters $h_1, h_2, \ldots, h_{2s-1}$;
Reduced model order r;
Tolerances $\epsilon,\ \delta$;
Initial column indices $\mathcal{J}$

1: Assemble H from $\{h_i\}_{i=1,\ldots,2s-1}$ as in (2.1)
2: $\mathcal{I}, \mathcal{J} = \texttt{cur-cross}(H, r, \mathcal{J}, \delta, \epsilon)$
3: $Q_{col}R = \texttt{qr}(H(:, \mathcal{J}))$
4: $LQ_{row} = \texttt{lq}(H(\mathcal{I}, :))$
5: $U_1\Sigma_1 V_1 = \texttt{svd}(H(\mathcal{I}, \mathcal{J}))$
6: $U\Sigma V^T = \texttt{svd}(RV_1\Sigma_1^{-1}U_1^T L)$
7: $\widehat{U} = Q_{col}U$ and $\widehat{V} = V^T Q_{row}$
8: $\widehat{U}_1 = \widehat{U}(1 : n_c - p, :)$ and $\widehat{U}_2 = \widehat{U}((p+1) : n_c, :)$
9: $A_r = \Sigma^{-1/2}\widehat{U}_1^T\widehat{U}_2\Sigma^{1/2}$
10: $B_r = \Sigma^{1/2}\widehat{V}^T[I_m\ 0]^T$
11: $C_r = [I_p\ 0]\ \widehat{U}\ \Sigma^{1/2}$

Remark 8. We note that the proposed `CUR-ERA` is a direct alternative to the `SVD-ERA` and requires the same assumptions. For data sets which do not satisfy the assumptions, i.e., the convergence-to-zero property, neither of these methods is guaranteed to yield stable reduced-order systems and may yield unsatisfactory results. For example, for mouse-gene data from [24] both `SVD-ERA` and `CUR-ERA` gave unsatisfactory results because the oscillatory nature of the data does not satisfy Assumption 2.

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

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.[6] 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

$$c \leq \left(\log|\det(H_{\text{maxvol}})| - \log|\det(H_{\text{start}})|\right) / \log(1+\epsilon), \tag{19}$$

where H_{start} is the submatrix $Q_{\mathcal{P}}(1{:}r,:) \in \mathbb{R}^{r\times r}$ of line 2 in Algorithm 2 [8].

We now provide the computational cost for cross-approximation.

LEMMA 9. *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*

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

Proof. The computational cost of Algorithm 1 is dominated by two QR decompositions and two calls to `maxvol` 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. The cost of `maxvol` is obtained by calculating the cost of Algorithm 2, which has complexity of $\mathcal{O}(c n_c r^2)$ for the columns and $\mathcal{O}(c n_r r^2)$ 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 $\mathcal{O}\left((n_c + n_r)\,r^2\right)$ (see Appendix A). ∎

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

THEOREM 10. *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 9 we have that the cross-approximation costs $\mathcal{O}\left(\kappa r^3 + \kappa c\,(n_c + n_r)\,r^2\right)$. 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. Considering all of this, Algorithm 3 has the same complexity as Algorithm 1. ∎

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.

[6] Note that we require $n_c > r$, and since the cost of computing the pseudoinverse of $Q_{\mathcal{P}}(1{:}r,:)$ is $\mathcal{O}(r^3)$, it does not contribute to the complexity as measured by $\mathcal{O}$-notation.

PROPOSITION 11. *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*

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

where ν *is as given in* (18)*.*

Proof. Let $\{\tilde{h}_k\}_{k=1}^{2s-1}$ be the entries of the CUR approximation H_r to the Hankel matrix H in (18). For any $1 \le k \le 2s-1$ we have by a simple triangle inequality that

$$\max_{k\in\{1,\dots,2s-1\}} \|h_k - C_r A_r^{k-1} B_r\|_F \le \max_{k\in\{1,\dots,2s-1\}} \left(\|h_k - \tilde{h}_k\|_F + \|\tilde{h}_k - C_r A_r^{k-1} B_r\|_F\right).$$

The first term on the right-hand side can be bounded by applying (18) to the block entry h_i in the approximate Hankel matrix H_r, i.e.,

$$\|h_k - \tilde{h}_k\|_F \le \nu^{-1}\sqrt{mp}(r+1)\sigma_{r+1}(H),$$

and the second term is bounded by

$$\|\tilde{h}_k - C_r A_r^{k-1} B_r\|_F \le \sqrt{r+m+p}\ \sigma_{r+1}(H),$$

which follows from a conservative application of Kung's error bound [18, Thm. 3.2]. Combining the two error bounds gives the stated result. ∎

There are several steps in the proof of the above result that suggest that the error bound is not tight. First, the maximum volume approximation algorithm yields a bound on the maximum elementwise error of the approximate Hankel matrix. However, the above result is over the Frobenius norm of a particular block in the Hankel matrix. Thus, the first part of the error bound conservatively bounds every mp entry in h_k by its maximum elementwise error. Second, the value of ν depends on how close the volume of the submatrix of the CUR approximation is to the maximum volume submatrix. While this may be difficult to determine in practice, we expect this part of the bound to be dominated by the decay of the singular values of the Hankel matrix. In practice, we see the error decay rate follow the decay rate of the singular values; we do not see a dominance of ν. In general, the above theorem is in the same spirit as Kung's original error bound, relating the Hankel singular values to the error in the impulse response approximation. Moreover, it should be noted that in [17] Kung's standard error bound and actual error were computed, and were found to not be sharp either.

5. Numerical results. In this section[7] we numerically compare Algorithm 3, `CUR-ERA`, with Kung's standard `SVD-ERA`.

We consider a mass-spring-damper model from [13, 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

[7] The code for this section is available under GitHub: https://github.com/bokramer/CURERA.

of some selected masses. The number of inputs is equal to the number of outputs, namely $m = p = 30$. This leads to a continuous-time model,

$$\dot{x}(t) = \tilde{A}x(t) + \tilde{B}u(t), \qquad y(t) = \tilde{C}x(t).$$

To apply ERA, we convert the continuous-time matrices to a discrete-time model of the form (1)–(2) with matrices A, B, C via a bilinear transform that maps the left half-plane onto the unit circle, e.g., [17, 12]. In the discrete-time setting, we then generate $2s = 1000$ Markov parameters via (3). The resulting Hankel matrix is of size $15,000 \times 15,000$.[8] As a first step for the 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.

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}

Stability is an important requirement for predictive modeling. For the discrete LTI systems in this work, stability involves the requirement that eigenvalues are contained within the unit circle in the complex plane $\mathbb{C}$. Recall that the CUR-ERA guarantees stability by Theorem 3. 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.

For further comparisons of the results of our CUR-ERA, we map the identified matrices in discrete time, A_r, B_r, C_r, back to continuous time, obtaining $\tilde{A}_r, \tilde{B}_r, \tilde{C}_r$, where we compare simulation results and the transfer functions. The identified reduced-order transfer function for both approaches is

$$G_r(i\omega) := \tilde{C}_r(i\omega I - \tilde{A}_r)^{-1}\tilde{B}_r, \quad \omega \in \mathbb{R},$$

and we compare them with the full-order model transfer function $G(i\omega) := \tilde{C}(i\omega I - \tilde{A})^{-1}\tilde{B}$, $\omega \in \mathbb{R}$. 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

[8] We emphasize that the matrices $\tilde{A}, \tilde{B}, \tilde{C}$ are never used in our algorithm, which only has access to impulse response data.

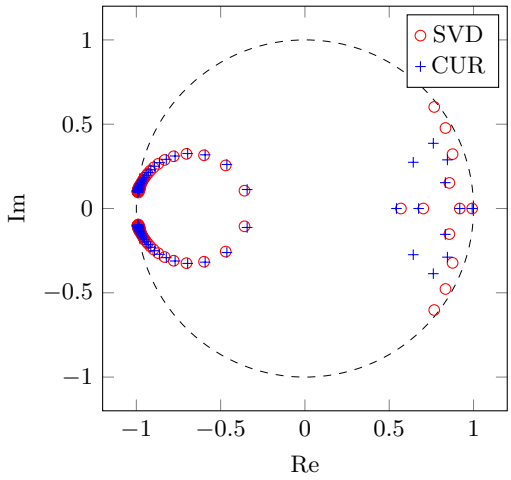

FIG. 1. *Eigenvalues of the identified system matrix A_r via* `CUR-ERA` *and* `SVD-ERA`*, respectively.*

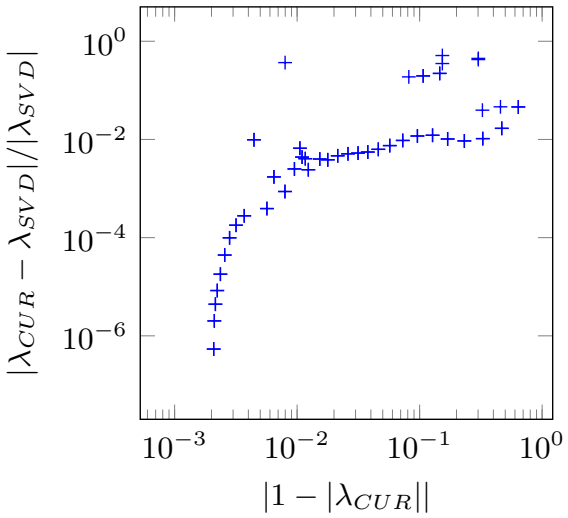

FIG. 2. *Relative accuracy of the eigenvalues $|\lambda_{SVD} - \lambda_{CUR}|/|\lambda_{SVD}|$ with respect to distance from the stability boundary $|1 - |\lambda_{CUR}||$. The relative accuracy of the largest (in magnitude) eigenvalues is significantly higher than for eigenvalues close to the origin,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

$\omega \approx 2Hz$, 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 superiorly here, but one needs to be able to sample the transfer function $G(\cdot)$ at chosen frequencies.

To compare the predictive capabilities of the reduced models from `SVD-ERA` and `CUR-ERA`, we simulate the reduced continuous-time systems as well as the original continuous-time system, from zero to $50s$ with zero initial conditions. The input vector $u(t) \in \mathbb{R}^m$ is chosen as in [13, 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.

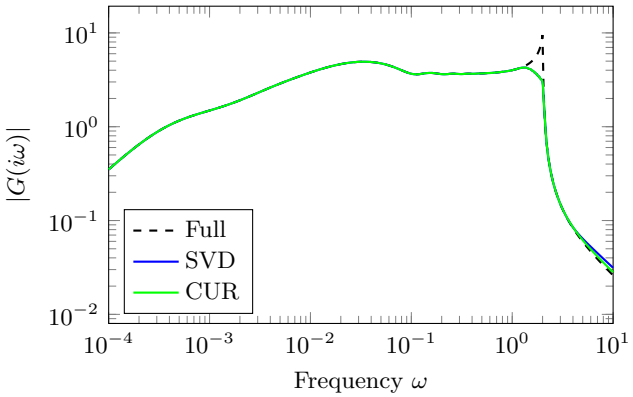

FIG. 3. *Transfer function of the full model and the two reduced-order models identified from* `CUR-ERA` *and* `SVD-ERA`.

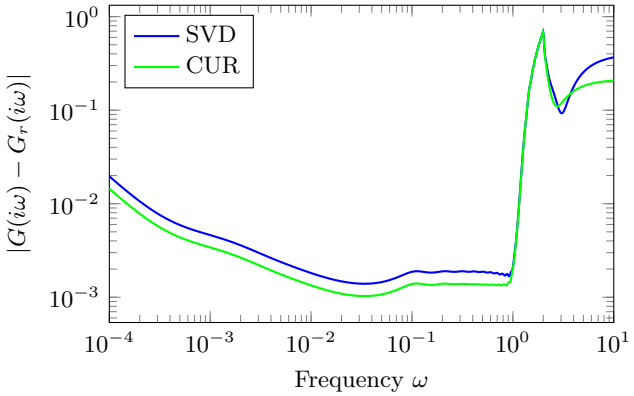

FIG. 4. *Error in transfer function from Figure* 3.

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 and discussions. 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 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 as the optimal rank r SVD approximation.

Appendix A. Fast difference between CUR factorized matrices. Let $H, \widehat{H} \in \mathbb{R}^{n_r \times n_c}$, $C_1, C_2 \in \mathbb{R}^{n_r \times r}$; $R_1, R_2 \in \mathbb{R}^{r \times n_c}$. 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 $\widehat{H} = C_2 R_2$ denote another matrix. A fast computation of $\|H - \widehat{H}\|_F$ can be obtained

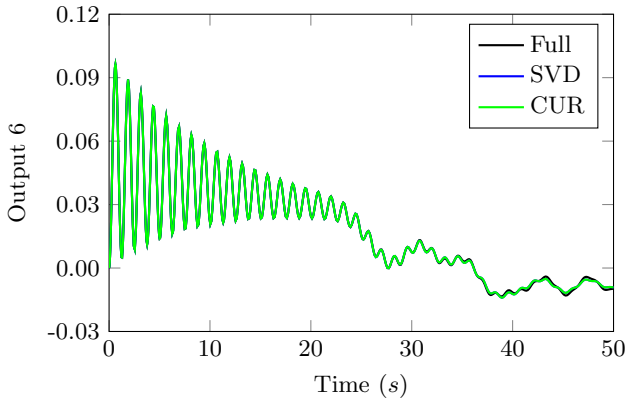

FIG. 5. *Output No. 6 of the continuous-time systems realized using the full and the two identified reduced-order models.*

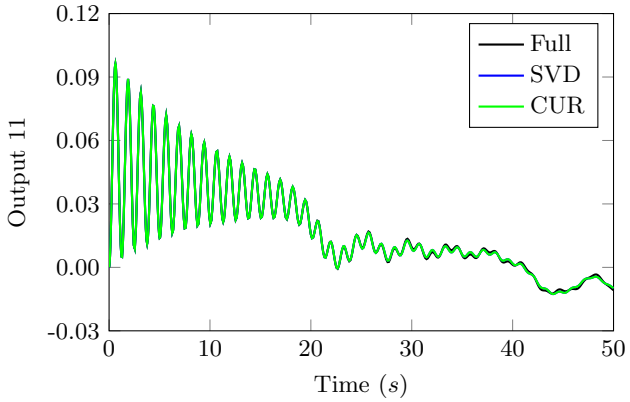

FIG. 6. *Output No. 11 of the continuous-time systems realized using the full and the two identified reduced-order models.*

by rewriting the difference as

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

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

$$E = M \odot M \implies E[i,j] = (M[i,j])(M[i,j]). \tag{20}$$

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

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

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

$$\mathcal{O}((n_r + n_c)r^2), \tag{21}$$

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. We also thank the anonymous referees for their comments, which helped improved this manuscript.

REFERENCES

[1] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, Adv. Des. Control 6, SIAM, Philadelphia, 2005, https://doi.org/10.1137/1.9780898718713.

[2] H. Cheng, Z. Gimbutas, P. G. Martinsson, and V. Rokhlin, *On the compression of low rank matrices*, SIAM J. Sci. Comput., 26 (2005), pp. 1389–1404, https://doi.org/10.1137/030602678.

[3] D. S. Djordjević and N. Č. Dinčić, *Reverse order law for the Moore–Penrose inverse*, J. Math. Anal. Appl., 361 (2010), pp. 252–261.

[4] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, *Relative-error CUR matrix decompositions*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 844–881, https://doi.org/10.1137/07070471X.

[5] Z. Drmač and S. Gugercin, *A new selection operator for the discrete empirical interpolation method—improved a priori error bound and extensions*, SIAM J. Sci. Comput., 38 (2016), pp. A631–A648, https://doi.org/10.1137/15M1019271.

[6] G. Golub and C. Reinsch, *Singular value decomposition and least squares solutions*, Numer. Math., 14 (1970), pp. 403–420.

[7] G. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1996, pp. 374–426.

[8] S. Goreinov, I. Oseledets, D. Savostyanov, E. Tyrtyshnikov, and N. Zamarashkin, *How to find a good submatrix*, in Matrix Methods: Theory, Algorithms and Applications, World Scientific, Hackensack, NJ, 2010, pp. 247–256.

[9] S. Goreinov and E. Tyrtyshnikov, *The maximal-volume concept in approximation by low-rank matrices*, in Structured Matrices in Mathematics, Computer Science, and Engineering, I (Boulder, CO, 1999), Contemp. Math. 280, AMS, Providence, RI, 2001, pp. 47–51.

[10] S. Goreinov and E. Tyrtyshnikov, *Quasi-optimality of a skeleton approximation of a matrix in the Chebyshev norm*, Dokl. Math., 83 (2011), pp. 374–375.

[11] S. A. Goreinov, E. Tyrtyshnikov, and N. Zamarashkin, *A theory of pseudoskeleton approximations*, Linear Algebra Appl., 261 (1997), pp. 1–21.

[12] S. Gugercin and A. C. Antoulas, *Model reduction of large-scale systems by least squares*, Linear Algebra Appl., 415 (2006), pp. 290–321.

[13] S. Gugercin, R. Polyuga, C. Beattie, and A. Van Der Schaft, *Structure-preserving tangential interpolation for model reduction of port-Hamiltonian systems*, Automatica J. IFAC, 48 (2012), pp. 1963–1974.

[14] J. Hokanson, *Numerically Stable and Statistically Efficient Algorithms for Large Scale Exponential Fitting*, Ph.D. thesis, Rice University, Houston, TX, 2013.

[15] A. C. Ionita and A. C. Antoulas, *Matrix pencils in time and frequency domain system identification*, in Developments in Control Theory towards Glocal Control, IET Control Eng. Ser. 76, IET, London, 2012, pp. 79–88.

[16] J.-N. Juang and R. Pappa, *An eigensystem realization algorithm for modal parameter identification and model reduction*, J. Guidance Control Dynam., 8 (1985), pp. 620–627.

[17] B. Kramer and S. Gugercin, *Tangential interpolation-based eigensystem realization algorithm for MIMO systems*, Math. Comput. Model. Dyn. Syst., 22 (2016), pp. 282–306.

[18] S.-Y. Kung, *A new identification and model reduction algorithm via singular value decomposition*, in Proceedings of the 12th Asilomar Conference on Circuits, Systems, and Computers, Pacific Grove, CA, 1978, pp. 705–714.

[19] Z. Ma, S. Ahuja, and C. W. Rowley, *Reduced-order models for control of fluids using the eigensystem realization algorithm*, Theoret. Comput. Fluid Dynam., 25 (2011), pp. 233–247.

[20] M. W. Mahoney and P. Drineas, *CUR matrix decompositions for improved data analysis*, Proc. Natl. Acad. Sci. USA, 106 (2009), pp. 697–702.

[21] I. Oseledets and E. Tyrtyshnikov, *TT-cross approximation for multidimensional arrays*, Linear Algebra Appl., 432 (2010), pp. 70–88.

[22] V. Rokhlin, A. Szlam, and M. Tygert, *A randomized algorithm for principal component analysis*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 1100–1124, https://doi.org/10.1137/080736417.

[23] D. C. Sorensen and M. Embree, *A DEIM induced CUR factorization*, SIAM J. Sci. Comput., 38 (2016), pp. A1454–A1482, https://doi.org/10.1137/140978430.

[24] B. Zhu, Q. Zhang, Y. Pan, E. Mace, B. York, A. Antoulas, C. Dacso, and B. W. O'Malley, *A cell-autonomous mammalian* 12 *hr clock coordinates metabolic and stress rhythms*, Cell Metabolism, 25 (2017), pp. 1305–1319.