NUMERICAL METHODS FOR SYSTEM IDENTIFICATION

MESHAL ALHARBI*

Abstract. In this project, we study some of the numerical methods of the identification of linear time-invariant (LTI) systems. LTI systems play an important role in many engineering disciplines as many systems are linear or can be approximated effectively by linear dynamics. The process of retrieving the structure of dynamical systems from observable data is known as system identification, and a popular algorithm for that is the eigensystem realization algorithm (ERA). We study a variant of ERA that uses randomized matrix decomposition approaches to improve numerical efficiency.

Key words. System identification, eigensystem realization algorithm, randomized low-rank approximation.

1. Introduction.

System identification is the process of recovering the structure of dynamical systems from observed, and possibly noisy data [15]. The problem might be limited to identifying the input-output relationships for specific inputs of interest, or the full underlying dynamical model might be desired [4]. Due to its applicability in various engineering challenges, system identification has garnered a vast and extensive body of literature. This project focus on the identification of linear time-invariant (LTI) discrete-time systems.

The identification of LTI systems adhere to well established principles from control and system theories, see [14, 16] and the references within. Many of the numerical methods for system identification rely on a low-rank approximation of the Hankel matrices assembled from observed data [9, 5]. A particular algorithm of interest is the eigensystem realization algorithm (ERA), which requires a singular value decomposition (SVD) of Hankel matrices [10]. Because of its simplicity and proven stability, ERA has seen widespread adaptation in many engineering disciplines.

Given that the size of Hankel matrices for many realistic applications can be in the order of $\mathbb{R}^{10000\times 10000}$ or more, the factorization of such matrices is computationally expensive. Recently, many variations of the ERA have been proposed to address computational cost of factorizing large Hankel matrices [11]. In particular, randomized approaches for matrix decomposition has been used to reduce the computation complexity of ERA by orders of magnitude [13]. This project aim to study the numerical properties of these methods and compare them with the original ERA.

The remainder of this report is organized as follows. Section 2 provides the necessary background about LTI systems and introduces the ERA. Section 3 presents the the modifications introduced to ERA to enhance its efficiency. Section 4 analytically compares between the studied ERA variants. Section 5 describes the datasets, the experimentation, and the numerical results. Finally, Section 6 presents the conclusions of this report.

2. Background.

A LTI discrete-time system with order n, m inputs, and l outputs is governed by the following state-space representation:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k \\ \boldsymbol{y}_k &= \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{D}\boldsymbol{u}_k \end{aligned} \tag{2.1}$$

*MIT Computational Science and Engineering (meshal@mit.edu).

where $\boldsymbol{x}_k \in \mathbb{R}^n$ is the state vector, $\boldsymbol{u}_k \in \mathbb{R}^m$ is the input vector, and $\boldsymbol{y}_k \in \mathbb{R}^l$ is the output vector at sampling time k. The matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is known as the state matrix, $\boldsymbol{B} \in \mathbb{R}^{n\times m}$ is the input matrix, $\boldsymbol{C} \in \mathbb{R}^{l\times n}$ is the output matrix, and $\boldsymbol{D} \in \mathbb{R}^{l\times m}$ is the input-output matrix.

The goal of system identification to to estimate the matrices $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ up to a similarity transformation $(\boldsymbol{TAT}^{-1}, \boldsymbol{TB}, \boldsymbol{CT}^{-1}, \boldsymbol{D})$. Note that the inevitable matrix $\boldsymbol{T} \in \mathbb{R}^{n\times n}$ does not change the relationship between $\{\boldsymbol{u}_k\}$ and $\{\boldsymbol{y}_k\}$. Moreover, if we assume that the system start from initial state $\boldsymbol{x}_0 = \boldsymbol{0}$, then the outputs $\{\boldsymbol{y}_k\}$ can be characterized by an equation known as the external description of the system:

$$\boldsymbol{y}_k = \sum_{j=0}^{k} \boldsymbol{h}_j \boldsymbol{u}_{k-j}, \quad k = 0, 1, \dots \tag{2.2}$$

where the matrices $\boldsymbol{h}_k \in \mathbb{R}^{l\times m}$ are called *Markov parameters* and are described by:

$$\boldsymbol{h}_k = \begin{cases} \boldsymbol{D}, & k = 0 \\ \boldsymbol{CA}^{k-1}\boldsymbol{B}, & k = 1, 2, \dots, 2s-1 \end{cases} \tag{2.3}$$

Markov parameters can be estimated by exciting the system with a series of impulse responses, or they can be obtained from general input-output data [6]. Note that $\boldsymbol{D}$ is immediately available from $\boldsymbol{h}_0$, thus, the identification task will focus on recovering the remaining matrices $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$.

ERA requires that Markov parameters $\{\boldsymbol{h}_k\}$ are available, and use them to form a block Hankel matrix as follows:

$$\begin{aligned} \boldsymbol{\mathcal{H}}_s &= \begin{bmatrix} \boldsymbol{h}_1 & \boldsymbol{h}_2 & \cdots & \boldsymbol{h}_s \\ \boldsymbol{h}_2 & \boldsymbol{h}_3 & \cdots & \boldsymbol{h}_{s+1} \\ \vdots & \vdots & \iddots & \vdots \\ \boldsymbol{h}_s & \boldsymbol{h}_{s+1} & \cdots & \boldsymbol{h}_{2s-1} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{CB} & \boldsymbol{CAB} & \cdots & \boldsymbol{CA}^{s-1}\boldsymbol{B} \\ \boldsymbol{CAB} & \boldsymbol{CA}^2\boldsymbol{B} & \cdots & \boldsymbol{CA}^{s}\boldsymbol{B} \\ \vdots & \vdots & \iddots & \vdots \\ \boldsymbol{CA}^{s-1}\boldsymbol{B} & \boldsymbol{CA}^{s}\boldsymbol{B} & \cdots & \boldsymbol{CA}^{2s-2}\boldsymbol{B} \end{bmatrix} \end{aligned} \tag{2.4}$$

thus, $\boldsymbol{\mathcal{H}}_s \in \mathbb{R}^{sl\times sm}$. Under the standard observability and controllability assumptions [1], it is known that this block Hankel matrix can be factorized into:

$$\boldsymbol{\mathcal{H}}_s = \boldsymbol{\mathcal{O}}\boldsymbol{\mathcal{C}} = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \vdots \\ \boldsymbol{CA}^{(s-1)} \end{bmatrix} \begin{bmatrix} \boldsymbol{B} & \boldsymbol{AB} & \cdots & \boldsymbol{A}^{s-1}\boldsymbol{B} \end{bmatrix} \tag{2.5}$$

where $\boldsymbol{\mathcal{O}}$ and $\boldsymbol{\mathcal{C}}$ are known as observability and controllability matrices, respectively. Note that if observability matrix $\boldsymbol{\mathcal{O}}$ is available, one can form a least-square problem for recovering $\boldsymbol{A}$ as follows:

$$\begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \vdots \\ \boldsymbol{CA}^{(s-2)} \end{bmatrix} A = \begin{bmatrix} \boldsymbol{CA} \\ \boldsymbol{CA}^2 \\ \vdots \\ \boldsymbol{CA}^{(s-1)} \end{bmatrix} \tag{2.6}$$

Algorithm 2.1 Eigensystem Realization (ERA)

```
Inputs: Markov parameters $\{\boldsymbol{h}_k\}$, target rank $1 \leq r \leq n$
Outputs: reduced system matrices $\boldsymbol{A}_r, \boldsymbol{B}_r, \boldsymbol{C}_r$
1: Form block Hankel matrix $\boldsymbol{\mathcal{H}}_s$ from $\{\boldsymbol{h}_k\}$ using (2.4)
2: Compute truncated SVD factorization $\boldsymbol{\mathcal{H}}_s \approx \boldsymbol{U}_r \boldsymbol{\Sigma}_r \boldsymbol{V}_r^T$
3: Partition $\boldsymbol{U}_r = \begin{bmatrix} \boldsymbol{\Upsilon}_f^{(r)} & * \end{bmatrix}^T = \begin{bmatrix} * & \boldsymbol{\Upsilon}_l^{(r)} \end{bmatrix}^T$ s.t. $\boldsymbol{\Upsilon}_f^{(r)}, \boldsymbol{\Upsilon}_l^{(r)} \in \mathbb{R}^{(s-1)l \times r}$
4: Compute $\boldsymbol{A}_r = [\boldsymbol{\Upsilon}_f^{(r)}]^\dagger \boldsymbol{\Upsilon}_l^{(r)}$
5: Compute $\boldsymbol{B}_r = \boldsymbol{\Sigma}_r \boldsymbol{V}_r^T \begin{bmatrix} \boldsymbol{I}_m & \boldsymbol{0} \end{bmatrix}^T$
6: Compute $\boldsymbol{C}_r = \begin{bmatrix} \boldsymbol{I}_l & \boldsymbol{0} \end{bmatrix} \boldsymbol{U}_r$
7: return $\boldsymbol{A}_r, \boldsymbol{B}_r, \boldsymbol{C}_r$
```

Notice that except for $\boldsymbol{A}$, all other matrices are known (i.e., they can be obtained by slicing $\boldsymbol{\mathcal{O}}$). ERA uses this fact, combined with the SVD of $\boldsymbol{\mathcal{H}}_s$ to estimate $\boldsymbol{\mathcal{O}}$, for reduced-order system identification as described in Algorithm 2.1. From here onward, we refer to the ERA that uses the vanilla SVD by SVD-ERA.

3. Efficient ERA.

As discussed briefly in Section 1, the size of Hankel matrices for many realistic applications is very large. More concretely, in practice m and l are usually in $\mathcal{O}(10 - 100)$ while the number of samples s is $\mathcal{O}(10^3 - 10^5)$. This will give rise to Hankel matrices with dimensions $\mathcal{O}(10^4 - 10^7)$, making the SVD factorization in ERA computationally prohibitive.

Luckily, as can be seen from (2.4), the block Hankel matrix $\boldsymbol{\mathcal{H}}_s$ is very structured; Thus, one might hope that its singular value decay quickly, which would facilitate randomized approaches to reduce the complexity of SVD factorization in ERA. The authors in [13] utilized this fact and used the procedure described in Algorithm 3.1 for randomized SVD factorization.

The algorithm start by generating a random Gaussian matrix $\boldsymbol{\Omega}$ to be used to approximate the action of the matrix $\boldsymbol{X}$ to be factorized. In particular, lines 2-6 describe a procedure for $q + 1$ subspace iteration with orthogonalization in-between each iteration. This procedure is also know as normalized power iterations, which is know to improve the floating-point numerical stability in case of large matrices [12]. Finally, the matrix $\boldsymbol{X}$ is projected into a smaller subspace where we perform a vanilla SVD decomposition. Note that for a given target rank r, the projection is done into a larger rank $(r + \rho)$ where $\rho \leq 20$ is oversampling parameter. Oversampling is known to exponentially reduce the probability of failure in capturing the optimal reduced subspace (i.e., failure has probability $\mathcal{O}(\rho^{-\rho})$ [8]).

Note that we can improve the efficiency of ERA further by not forming the block Hankel matrix $\boldsymbol{\mathcal{H}}_s$ explicitly and instead rely on Markov parameters $\{\boldsymbol{h}_k\}$ for matrix-vector and matrix-matrix products. This can be achieved as block Hankel matrix can be viewed as special cases of circulant matrices where a fast matrix-vector product can be obtained by using fast Fourier transforms (FFT) [7]. We do not pursue such modifications in this project as the major gains are in term of memory complexity, and they are mostly significant in the cases of very massive Hankel matrices.

The outputs of Algorithm 3.1 can be used to replace line 2 in Algorithm 2.1 to achieve a more efficient ERA. From here onward, we to refer to the algorithm that uses RandSVD in combination with ERA for system identification by RandSVD-ERA and it will be our main algorithm of interest in this project.

Algorithm 3.1 RandSVD

```
Inputs: matrix $\boldsymbol{X} \in \mathbb{R}^{M\times N}$, target rank $r$, oversampling parameter $\rho$ such that
        $r + \rho \leq \min\{M, N\}$, number of subspace iterations $q \geq 0$
Outputs: truncated approximate factorization $\boldsymbol{X} \approx \hat{\boldsymbol{U}}_r\hat{\boldsymbol{\Sigma}}_r\hat{\boldsymbol{V}}_r^T$
Draw standard Gaussian random matrix $\boldsymbol{\Omega} \in \mathbb{R}^{N\times(r+\rho)}$
Form $\boldsymbol{Y}_0 = \boldsymbol{X}\boldsymbol{\Omega}$ and compute its QR factorization $\boldsymbol{Y}_0 = \boldsymbol{Q}_0\boldsymbol{R}_0$
for $j = 1, 2, ..., q$ do
    Form $\tilde{\boldsymbol{Y}}_j = \boldsymbol{X}^*\boldsymbol{Q}_{j-1}$ and compute its QR factorization $\tilde{\boldsymbol{Y}}_j = \tilde{\boldsymbol{Q}}_j\tilde{\boldsymbol{R}}_j$
    Form $\boldsymbol{Y}_j = \boldsymbol{X}\tilde{\boldsymbol{Q}}_j$ and compute its QR factorization $\boldsymbol{Y}_j = \boldsymbol{Q}_j\boldsymbol{R}_j$
end for
Form $\boldsymbol{B} = \boldsymbol{Q}_q^*\boldsymbol{X}$
Calculate SVD $\boldsymbol{B} = \boldsymbol{U}_B\boldsymbol{\Sigma}_B\boldsymbol{V}_B^T$
Set $\hat{\boldsymbol{U}}_r = \boldsymbol{Q}_q\boldsymbol{U}_B(:, 1:r)$, $\hat{\boldsymbol{\Sigma}}_r = \boldsymbol{\Sigma}_B(1:r, 1:r)$, and $\hat{\boldsymbol{V}}_r = \boldsymbol{V}_B(:, 1:r)$
return $\hat{\boldsymbol{U}}_r, \hat{\boldsymbol{\Sigma}}_r, \hat{\boldsymbol{V}}_r^T$
```

4. Analytical Comparison. In this section, we compare between SVD-ERA and RandSVD-ERA in term of their asymptotic number of operations, numerical accuracy, and stability.

4.1. Computational Cost. Given that the SVD calculation dominates the computations for both SVD-ERA and RandSVD-ERA, we mainly compare the two algorithms in terms of the asymptotic number of operations required to obtain the SVD factorization. For a general dense matrix in $\mathbb{R}^{M\times N}$, it is known that SVD factorization requires $\mathcal{O}(MN\min\{M,N\})$. Thus, asymptotic cost of SVD-ERA is:

$$\text{Cost SVD-ERA} = \mathcal{O}(s^3 lm \min\{l, m\}) \tag{4.1}$$

On the other hand, the cost of randomized SVD factorization consists of two parts: the cost of subspace iterations and the cost of the deterministic SVD. Let T_{mul} be the cost of matrix-matrix product, then for a dense matrix in $\mathbb{R}^{M\times N}$ with $M \geq N$, one can show that the cost of randomized SVD is [13]:

$$\text{Cost RandSVD} = (2q+1)(r+\rho)T_{\text{mul}} + \mathcal{O}(r^2(M+N)) \tag{4.2}$$

Then, for RandSVD-ERA, the asymptotic cost is given by:

$$\text{Cost RandSVD-ERA} = \mathcal{O}(rs^2 lm + r^2 s(l+m)) \tag{4.3}$$

Note that the cost does not depend on q is it is a constant that does not scale with the problem size. In practice, $s \gg r, l$, or m. Thus, RandSVD-ERA achieve an order of magnitude improvement in asymptotic cost over SVD-ERA.

4.2. Error Analysis. The error bounds for randomized algorithms are well studied in the literature, see [8] for a formal treatment of the subject. Thus, this section will focus on how errors in SVD factorization induce errors in the subsequent system identification step.

Given that system identification is achieved up to a similarity transform, the main notion of accuracy we study is in term of the eigenvalues of state matrix $\boldsymbol{A}_r$. Let $\boldsymbol{A}_r$, and $\hat{\boldsymbol{A}}_r$ be the matrices recovered by SVD-ERA and RandSVD-ERA, respectively. If

$\psi(\cdot)$ donate the spectrum of a matrix, then the spectral variation between between $\boldsymbol{A}_r$ and $\hat{\boldsymbol{A}}_r$ is defined as [3]:

$$\mathrm{SV}\left(\psi(\boldsymbol{A}_r), \psi(\hat{\boldsymbol{A}}_r)\right) = \max_{1\leq j\leq n} \min_{1\leq i\leq n} \left|\lambda_i(\boldsymbol{A}_r) - \lambda_j(\hat{\boldsymbol{A}}_r)\right| \tag{4.4}$$

Let $\mathcal{R}(\boldsymbol{U}_r)$ and $\mathcal{R}(\hat{\boldsymbol{U}}_r)$ be the column spaces generated by SVD-ERA and RandSVD-ERA. Then, the spectral variation between $\boldsymbol{A}_r$ and $\hat{\boldsymbol{A}}_r$ is bounded by:

$$\mathrm{SV}\left(\psi(\boldsymbol{A}_r), \psi(\hat{\boldsymbol{A}}_r)\right) \leq \kappa(\boldsymbol{W})\eta \left(1 + \frac{\sqrt{2}\left\|[\boldsymbol{\Upsilon}_f^{(r)}]^\dagger\right\|_2}{1-\eta}\right) \tag{4.5}$$

$$\eta = 2\sin\theta_{\max}\left\|[\boldsymbol{\Upsilon}_f^{(r)}]^\dagger\right\|_2 \tag{4.6}$$

where $\boldsymbol{W}$ is the left eigenvectors of $\boldsymbol{A}_r$, $\kappa(\cdot)$ is the condition number, and $\eta < 1$ is a factor that depends on the maximum canonical angel $\theta_{\max}$ between $\mathcal{R}(\boldsymbol{U}_r)$ and $\mathcal{R}(\hat{\boldsymbol{U}}_r)$. See [13] for a detailed proof.

There are multiple factors that affect the bound in (4.5); The dependence on the condition number of $\boldsymbol{W}$ and the norm of pseudoinverse of $\boldsymbol{\Upsilon}_f^{(r)}$ capture natural sensitively in original linear system and the ERA's least-square fitting, respectively. On the other hand, η arises from the mismatch between the optimal column space $\mathcal{R}(\boldsymbol{U}_r)$ and the approximate column space $\mathcal{R}(\hat{\boldsymbol{U}}_r)$.

4.3. Stability. Compared to SVD-ERA, RandSVD-ERA mainly requires additional QR factorizations, which can be done using unitary transformations that do not affect the numerical stability. It is equally important to discuss the *stability of the dynamical systems* identified by SVD-ERA and RandSVD-ERA.

We say that a LTI system is stable of spectral radius of its eigenvalues $\rho(\boldsymbol{A}_r) < 1$. For SVD-ERA, the observability and controllability assumptions grantee such notion of stability [1]. From (4.5), we see that RandSVD-ERA can shift the eigenvalues outside the unit circle; Thus, for $\hat{\boldsymbol{A}}_r$ to be stable, RandSVD-ERA must achieve:

$$\kappa(\boldsymbol{W})\eta \left(1 + \frac{\sqrt{2}\left\|[\boldsymbol{\Upsilon}_f^{(r)}]^\dagger\right\|_2}{1-\eta}\right) < 1 - \rho(\boldsymbol{A}_r) \tag{4.7}$$

This introduce an upper bound on η and the allowable discrepancy between the subspaces $\mathcal{R}(\boldsymbol{U}_r)$ and $\mathcal{R}(\hat{\boldsymbol{U}}_r)$.

5. Numerical Experiment.

We now compare SVD-ERA and RandSVD-ERA through a numerical example. Both algorithms have been implemented using Julia and its Standard Library. The experiments have done on a computer with an AMD Ryzen 9 5900HX CPU and 16GB of DDR4 RAM.

5.1. Dataset. We use a dataset that is part of Oberwolfach Benchmark Collection, which is a stable benchmark for model order reduction and system identification techniques. In particular, the dataset is generated from ordinary differential equations for discretized heat transfer problem of steel profiles. See [2] for data creation details.

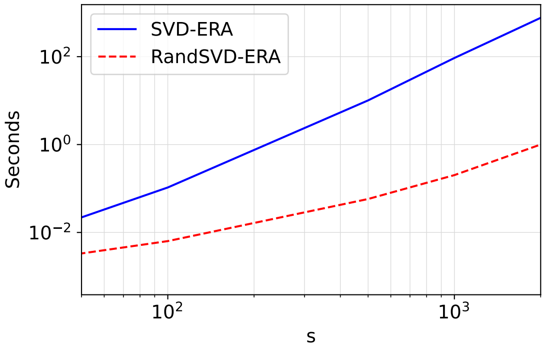

FIG. 1. *Timing comparison between SVD-ERA and RandSVD-ERA for multiple values of s.*

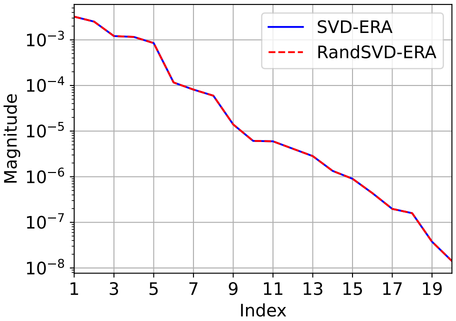

FIG. 2. *Singular values of $\boldsymbol{\mathcal{H}}_s$ and $\hat{\boldsymbol{\mathcal{H}}}_s$.*

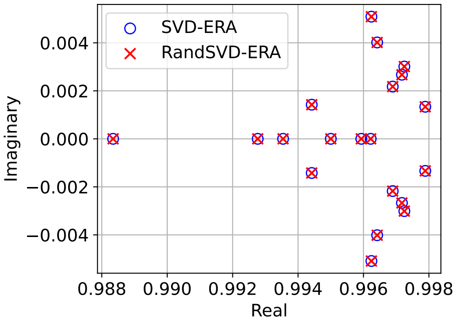

FIG. 3. *Eigenvalues of $\boldsymbol{A}_r$ and $\hat{\boldsymbol{A}}_r$.*

As the heat transfer dataset describe a continuous time problem, we used the `c2d` function from `ControlSystems` library to convert the problem to discrete time. The problem has $l = 6$ inputs and $m = 7$ outputs, while the number of samples s can be chosen freely. We fix the reduced model order $r = 20$ on all the results shown.

5.2. Timing and Memory. To test the computational complexity of the two algorithms, we varied the number of samples s from 50 to 2000. This resulted in a matrix in the largest case of size $\mathbb{R}^{12000\times 14000}$.

Figure 1 shows the timing each method spent in the calculation of the SVD factorization. We used Julia macro `@btime` from `BenchmarkTools` library to produce theses results. As predicted by the analysis in Section 4, RandSVD-ERA offers an order of magnitude improvement in computation times when compared to SVD-ERA.

Memory requirement (measured using allocated memory from `@btime`) has shown similar trends; For example, for the case of $s = 2000$, SVD-ERA used 6.80 GiB of RAM while RandSVD-ERA only used 78.65 MiB.

5.3. Accuracy. We now discuss the accuracy of RandSVD-ERA when compared to SVD-ERA. We have chosen $q = 1$ as the number of subspace iterations. Figure 2 shows the singular values of Hankel matrices generated from the two SVD-ERA and RandSVD-ERA. We can notice that the singular values of the Hankel matrices decay quickly and there is a good agreement between the two algorithms.

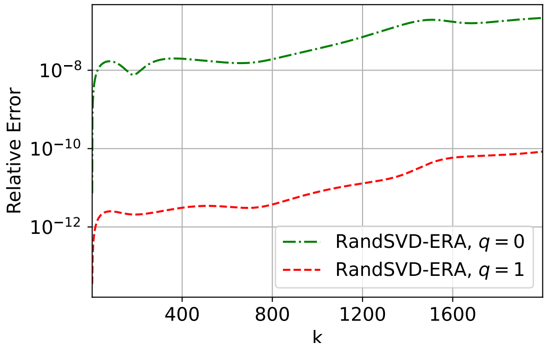

FIG. 4. *Relative errors in the Markov parameters of RandSVD-ERA for multiple values of q.*

Figure 3 shows the eigenvalues of the recovered state matrices $\boldsymbol{A}_r$ and $\hat{\boldsymbol{A}}_r$. Inspecting the figure, we can see that RandSVD-ERA where able to recover accurate state matrix $\hat{\boldsymbol{A}}_r$ in term of its eigenvalues. A more quantitative result is in term of the spectral variation as defined in 4.4, which we measured to be:

$$\mathrm{SV}\left(\psi(\boldsymbol{A}_r), \psi(\hat{\boldsymbol{A}}_r)\right) \approx 2.9720 \times 10^{-13}$$

To investigate the effects of the number of subspace iterations q, we study the accuracy of the Markov parameters generated by RandSVD-ERA when compared to SVD-ERA. We define this accuracy in term of norm of relative errors as:

$$E_k = \frac{\left\| \boldsymbol{C}_r \boldsymbol{A}_r^k \boldsymbol{B}_r - \hat{\boldsymbol{C}}_r \hat{\boldsymbol{A}}_r^k \hat{\boldsymbol{B}}_r \right\|_2}{\left\| \boldsymbol{C}_r \boldsymbol{A}_r^k \boldsymbol{B}_r \right\|_2}, \quad k = 1, 2, \ldots, 2s - 1 \tag{5.1}$$

Figure 4 shows this relative error for the cases of $q = 0$ and $q = 1$. We can notice that multiple subspace iterations (i.e., $q = 1$) leads to smaller relative errors which is in the order of 10^{-11}. We also tested values $q > 1$ and noticed that the errors do not improve further.

Additionally, from Figure 4, we can see an increasing trend in errors as the sample index k increases. This can be attributed to the recursive manner in the definition of the Markov parameters (Equation 2.3), which can lead to a compounding of the errors as the sample index k increases.

6. Conclusion. In this project, we studied the numerical properties of the eigensystem realization algorithm and some of its recent variants. In particular, we considered a variant called RandSVD-ERA that utilizes randomized techniques to deal with the SVD factorization of large block Hankel matrices. When compared to the original SVD-ERA, RandSVD-ERA exhibits appealing properties as its computational complexity is an order of magnitude lower. Additionally, we investigated the accuracy of RandSVD-ERA theoretically and empirically. We showed that both of the eigenvalues and Markov parameters recovered by RandSVD-ERA are accurate up to $\mathcal{O}(10^{-12})$ when compared to the ones recovered by SVD-ERA.

Nonetheless, there are cases where one might prefer SVD-ERA over RandSVD-ERA. In particular, if A_r is ill-conditioned and its spectral radius $\rho(\boldsymbol{A}_r)$ is very close to one. In these cases, the bound shown in Equation 4.5 might not hold, and RandSVD-ERA is not guaranteed to recover a stable dynamical system. Other cases include contexts where the recovered system will be used to simulate very long trajectories. In this setting, the increasing error trend of Markov parameters of RandSVD-ERA might be undesirable.

As discussed briefly in Section 3, further improvements over RandSVD-ERA include utilizing the block Hankel structure for efficient matrix-vector products. Doing this has the potential of reducing the dependency on s in the computational complexity of RandSVD-ERA from $\mathcal{O}(s^2)$ to $\mathcal{O}(s \log(s))$. It will also have more drastic impacts on memory requirements as the entire Hankel matrix does not need to be formed.

REFERENCES

[1] A. C. Antoulas, *Approximation of large-scale dynamical systems*, Society for Industrial and Applied Mathematics, 2005, https://doi.org/10.1137/1.9780898718713.

[2] P. Benner, V. Mehrmann, and D. C. Sorensen, *Dimension reduction of large-scale systems*, vol. 45, Springer, 2005, https://doi.org/10.1007/3-540-27909-1_19.

[3] R. Bhatia, *Matrix analysis*, vol. 169, Springer Science & Business Media, 2013, https://doi.org/10.1007/978-1-4612-0653-8.

[4] B. N. Datta, *Numerical methods for linear control systems*, vol. 1, Academic Press, 2004, https://doi.org/10.1016/B978-012203590-6/50013-6.

[5] M. Fazel, T. K. Pong, D. Sun, and P. Tseng, *Hankel matrix rank minimization with applications to system identification and realization*, SIAM Journal on Matrix Analysis and Applications, 34 (2013), pp. 946–977, https://doi.org/10.1137/110853996.

[6] M. S. Fledderjohn, M. S. Holzel, H. J. Palanthandalam-Madapusi, R. J. Fuentes, and D. S. Bernstein, *A comparison of least squares algorithms for estimating markov parameters*, in Proceedings of the 2010 American Control Conference, 2010, pp. 3735–3740, https://doi.org/10.1109/ACC.2010.5530673.

[7] G. H. Golub and C. F. Van Loan, *Matrix computations*, JHU press, 2013.

[8] N. Halko, P. G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Review, 53 (2011), pp. 217–288, https://doi.org/10.1137/090771806.

[9] M. Ishteva, K. Usevich, and I. Markovsky, *Factorization approach to structured low-rank approximation with applications*, SIAM Journal on Matrix Analysis and Applications, 35 (2014), pp. 1180–1204, https://doi.org/10.1137/130931655.

[10] J.-N. Juang and R. S. Pappa, *An eigensystem realization algorithm for modal parameter identification and model reduction*, Journal of guidance, control, and dynamics, 8 (1985), pp. 620–627, https://doi.org/10.2514/3.20031.

[11] B. Kramer and A. A. Gorodetsky, *System identification via cur-factored hankel approximation*, SIAM Journal on Scientific Computing, 40 (2018), pp. A848–A866, https://doi.org/10.1137/17M1137632.

[12] P.-G. Martinsson, A. Szlam, M. Tygert, et al., *Normalized power iterations for the computation of svd*, Manuscript., Nov, (2010).

[13] R. Minster, A. K. Saibaba, J. Kar, and A. Chakrabortty, *Efficient algorithms for eigensystem realization using randomized svd*, SIAM Journal on Matrix Analysis and Applications, 42 (2021), pp. 1045–1072, https://doi.org/10.1137/20M1327616.

[14] E. Reynders, *System identification methods for (operational) modal analysis: review and comparison*, Archives of Computational Methods in Engineering, 19 (2012), pp. 51–124, https://doi.org/10.1007/s11831-012-9069-x.

[15] P. Van den Hof, *Closed-loop issues in system identification*, Annual Reviews in Control, 22 (1998), pp. 173–186, https://doi.org/10.1016/S1367-5788(98)00016-9.

[16] M. Verhaegen and V. Verdult, *Filtering and system identification: a least squares approach*, Cambridge university press, 2007, https://doi.org/10.1017/CBO9780511618888.