



**HAL**  
open science

# A Data-Driven Krylov Model Order Reduction for Large-Scale Dynamical Systems

M. Hamadi, Khalide Jbilou, A. Ratnani

► **To cite this version:**

M. Hamadi, Khalide Jbilou, A. Ratnani. A Data-Driven Krylov Model Order Reduction for Large-Scale Dynamical Systems. *Journal of Scientific Computing*, 2023, 95, pp.1-26. 10.1007/s10915-023-02122-8 . hal-04412653

**HAL Id: hal-04412653**

**<https://hal.science/hal-04412653>**

Submitted on 7 May 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A DATA-DRIVEN KRYLOV MODEL ORDER REDUCTION FOR LARGE-SCALE DYNAMICAL SYSTEMS

M.A. HAMADI <sup>†\*</sup>, K. JBILOU<sup>\*†</sup>, AND A. RATNANI<sup>\*</sup>

**Abstract.** Dynamical systems that involve non-linearity of the dynamics is a major challenge encountered in learning these systems. Similarly, the lack of adequate models for phenomena that reflect the governing physics can be an obstacle to an appropriate analysis. Nonetheless, some numerically or experimentally measured data can be found. Based on this data, and using a data-driven method such as the Loewner framework, it is possible to manage this data to derive a high fidelity reduced dynamical system that mimics the behaviour of the original data. In this paper, we tackle the issue of large amount of data presented by samples of transfer functions in a frequency-domain. The main step in this framework consists in computing singular value decomposition (SVD) of the Loewner matrix which provides accurate reduced systems. However, the large amount of data prevents this decomposition from being computed properly. We exploit the fact that the Loewner and shifted Loewner matrices, the key tools of Loewner framework, satisfy certain large scale Sylvester matrix equations. Using an extended block Krylov subspace method, a good approximation in a factored form of the Loewner and shifted Loewner matrices can be obtained and a minimal computation cost of the SVD is ensured. This method facilitates the process of a large amount of data and guarantees a good quality of the inferred model at the end of the process. Accuracy and efficiency of our method are assessed in the final section.

**Key words.** Interpolation, Krylov subspaces, Loewner framework, Sylvester equation.

**1. Introduction.** Over the last years, Model Order Reduction (MOR) techniques have gained substantial attention beyond the scientific computing and computational engineering communities, and this is due to **their** reputation as a powerful tool for addressing what the MOR communities refer to "the curse of dimensionality". MOR is a tool capable to overcome this problem by providing lower order models that are suitable for reproducing the input-output behaviour of the original dynamical systems at a small marginal cost. These reduced models are characterized by its ease of manipulation and control, unlike the original high order systems of a complex dynamics. One of the mainstream methods in MOR is the Krylov subspace projection method, for more details on this method, we refer the readers to [5, 9, 16, 20]. On the other hand, the Balanced Truncation (BT) method [15, 23, 24], which is based on discarding the states that are both difficult to reach and to observe. Other materials and details about MOR techniques can be found in [2, 3, 7] and the references therein.

To construct high fidelity reduced models using the above mentioned methods, access to the entire structure matrices of the original models has always been necessary. Those methods are known as intrusive methods. On the contrary, we have the non-intrusive methods which are the main subject of this paper, and the main goal is to build a low-order model exclusively from data (i.e., measurements), which means that the whole process does not require any knowledge about the governing equations that describe the original dynamical systems.

The Loewner-based approach is among many widely used approaches for deriving the input-output behaviour of the original dynamical system only from data. This approach was first proposed in [22] to construct a generalized realization state space problem, coupled with tangential interpolation data. We refer to "data" as samples of transfer functions in a frequency-domain, or as the output measured for an input in a time-domain. In a frequency domain, a data-driven Loewner framework to construct reduced models was addressed in [12] and an extension to time-domain data was suggested in [26]. We follow the Loewner process which is based on splitting data into left and right data, building the data-matrices, and finally the key tool of this framework which is the construction of two matrices  $\mathbb{L}, \mathbb{L}_s \in \mathbb{C}^{N \times N}$  named respectively Loewner and shifted Loewner matrices. Further details about the fundamental role of these matrices in the proposed approach will be tackled in the next section. We refer to  $N$  by the amount of data that we collect. After setting up the key tool of the Loewner framework, we come to the final step, which is the computation of the singular value decomposition

---

\*Université Mohammed VI polytechnique, Lot 660, Hay Moulay Rachid, Ben Guerir, 43150 Maroc; amine.hamadi@um6p.ma; ahmed.ratnani@um6p.ma

<sup>†</sup>Université du Littoral Côte d'Opale, 50 Rue F. Buisson, BP 699-62228 Calais cedex, France; jbilou@univ-littoral.fr

(SVD) of  $\mathbb{L}$  or a linear combination of ( $\mathbb{L}$  and  $\mathbb{L}_s$ ) [21], and then inferred a lower order model by truncating the SVD followed by a projection. Those are the main steps of Loewner framework for model reduction as it is described in [21, 22]. The authors in [25] have exploited the Cauchy-like structure of  $\mathbb{L}$  and  $\mathbb{L}_s$  in order to avoid the explicit allocation of the two matrices, in particular when the amount of data  $N$  is too large. In [8], the authors proposed an heuristic methodology to collect the data for learning dynamical systems. A reformulation of the classic Balanced Truncation (BT) model reduction method has been addressed in [11], where the authors shown how to compute the product of the Gramians, the key tool of BT, just from data by using some quadrature techniques. In the last two mentioned papers, the authors obtained good results when the amount of data  $N$  is low or medium, but for large amount of data, no numerical experiments have been reported.

In this paper, we are interested in the field of model reduction for frequency response problems, and especially, we tackled the issue of large amount of data, since for large  $N$ , the SVD of the Loewner and shifted Loewner matrices  $\mathbb{L}, \mathbb{L}_s \in \mathbb{C}^{N \times N}$  may require a prohibitive computational cost due to the dense structure (*i.e.*, a matrix with many non-zero components) of such large matrices. To solve this issue, we exploit the fact that  $\mathbb{L}$  and  $\mathbb{L}_s$  satisfy two large scale Sylvester equations with low-rank right-hand sides expressed as  $BB^T$ , where  $B \in \mathbb{C}^{n \times s}$  and  $s \ll n$ . We propose to solve those large scale matrix equations by means of an extended bloc Arnoldi method, which will allow us to construct a good approximation of  $\mathbb{L}$  and  $\mathbb{L}_s$  in a factored form. This factorization would help us to reduce storage and computation requirements, also it guarantees a proper computation of the SVD of  $\mathbb{L}$  in order to get a reduced model that can represent accurately the data. The ADI-Galerkin method proposed in [6], is another algebraic numerical tool used in [18] to build approximate solutions to large scale Sylvester equations. The author has shown the effectiveness of using the ADI-Galerkin method in the computation of the SVD of  $\mathbb{L}$  when dealing with large  $N$ . However, no results regarding the effectiveness of the constructed reduced model have been reported. A comparison with this method is presented in the numerical section.

In this work, we propose to use an extended bloc Arnoldi method to find approximate solutions for large and sparse Sylvester matrix equations of the form

$$M\mathbb{L} - \Lambda\mathbb{L} = BB^T, \quad (1.1)$$

where  $M, \Lambda$  and  $B$  are matrices of appropriate dimension. In [17, 27], the authors have shown the performance of this method for solving large-scale Riccati and Lyapunov matrix equations. As we mentioned before, in order to construct reduced models that can represent accurately the data, an SVD of  $\mathbb{L} \in \mathbb{C}^{N \times N}$  should be established. Our main goal is to find an approximate solution  $\tilde{\mathbb{L}}$  to  $\mathbb{L}$  and use it instead in the computation of the SVD, which will then be used to build the reduced model. Assume that an approximate solution  $\tilde{\mathbb{L}}$  has been computed using a numerical Krylov-subspace method, then according to [1, 19], the approximate solution has the following form

$$\mathbb{L} \approx \tilde{\mathbb{L}} = \mathcal{V}_m Y_m \mathcal{V}_m^*,$$

where  $\mathcal{V}_m \in \mathbb{C}^{N \times 2ms}$  is a basis-matrix of the extended bloc Krylov subspace,  $Y_m \in \mathbb{C}^{2ms \times 2ms}$  is a small matrix ( $m \ll N$ ) and  $s$  is the column number of  $B$ . To establish an SVD of  $\mathbb{L} \in \mathbb{C}^{N \times N}$  using our technique, we will need only to compute the SVD of the small matrix  $Y_m$  (*i.e.*,  $[P, \Sigma, Q] = \text{svd}(Y_m)$ ), and then we get

$$\mathbb{L} \approx \mathcal{V}_m P \Sigma (W_m Q)^*. \quad (1.2)$$

From the above, it appears that we have achieved a reduction in the computational complexity of the SVD of  $\mathbb{L}$  from  $O(N^3)$  to  $O(4Nm^2s^2)$  where  $m$  is the order of the small matrix  $Y_m$ , and  $s$  is the column number of the low-rank right-hand side of (1.1). This enables us to process vast amount of data effectively as it is shown in the section of numerical examples.

We propose the following organization of the manuscript by sections. In Section 2, we give a brief review on the Loewner framework and we describe the main steps of the process. A presentation

of the extended block Krylov subspace method is shown in Section 3 for solving the two large scale Sylvester equations associated to the Lowener and shifted Lowener matrices. Finally, in Section 4, the accuracy and efficiency of our method is evaluated and compared to other known methods.

**2. The Loewner framework.** A brief introduction to the Loewner framework and its main steps are described in this section, for more details, we refer the reader to [21, 22] and the references therein. The Lowener framework is a data-driven and non-intrusive model order reduction method. In particular, it does not require any governing equations or system matrices that describe the model. Unlike the classical intrusive methods that require the whole system matrices of the original model to construct a reduced one.

**2.1. Krylov-based methods for intrusive models.** We first give a brief description of a Krylov subspace based model order reduction method; see [9, 14, 20] for more details. Consider a linear time invariant (LTI) dynamical system given by

$$(\Sigma) : \begin{cases} E \dot{x}(t) &= Ax(t) + Bu(t), & x(0) = 0, \\ y(t) &= Cx(t) + Du(t), \end{cases} \quad (2.1)$$

where  $x(t) \in \mathbb{R}^n$  are the variables (or states if  $E$  is non-singular),  $u(t) \in \mathbb{R}^q$  and  $y(t) \in \mathbb{R}^p$  are the inputs and outputs, respectively. The matrices  $A, E \in \mathbb{R}^{n \times n}$  are large and sparse,  $B \in \mathbb{R}^{n \times q}$  and  $C^T \in \mathbb{R}^{n \times p}$  are supposed to have small number of columns i.e.,  $q, p \ll n$ ,  $D \in \mathbb{R}^{p \times q}$ . This approach seeks to construct the following reduced model

$$(\hat{\Sigma}) : \begin{cases} \hat{E} \dot{\hat{x}}(t) &= \hat{A} \hat{x}(t) + \hat{B} u(t), \\ \hat{y}(t) &= \hat{C} \hat{x}(t) + \hat{D} u(t), \end{cases} \quad (2.2)$$

where  $\hat{A}, \hat{E} \in \mathbb{R}^{r \times r}$ ,  $\hat{B} \in \mathbb{R}^{r \times q}$  and  $\hat{C}^T \in \mathbb{R}^{r \times p}$  such that  $r \ll n$ . In the frequency domain, the original and reduced transfer functions associated to the original and the reduced systems are defined by  $H(s) = C(sE - A)^{-1}B + D$  and  $\hat{H}(s) = \hat{C}(z\hat{E} - \hat{A})^{-1}\hat{B} + \hat{D}$ , respectively. Then, in order to measure the accuracy of the resulting reduced system, one can compute the error-norm  $\|H - \hat{H}\|$  with respect to a certain norm. As it is described in [2, 9], the projection process is as follows, firstly we define two subspaces as  $\mathcal{K}_1 = \text{Range}(\mathcal{V})$  and  $\mathcal{K}_2 = \text{Range}(\mathcal{W})$ , where  $\mathcal{V}, \mathcal{W} \in \mathbb{R}^{n \times r}$  are respectively the corresponding basis matrices. Notice here that  $\mathcal{K}_1$  and  $\mathcal{K}_2$  are Krylov-based subspaces associated to the couple of matrices  $(A, B)$  and  $(A, C^T)$ . After approximating the full order state  $x(t)$  by  $\mathcal{V}\hat{x}$ , and enforcing the Petrove-Galerkin condition, we obtain

$$\begin{cases} \mathcal{W}^T(E\mathcal{V}\dot{\hat{x}}(t) - A\mathcal{V}\hat{x}(t) - B u(t)) = 0, \\ y(t) = C\mathcal{V}\hat{x}(t) + D u(t), \end{cases} \quad (2.3)$$

Finally, we get the reduced system (2.2) with the following system matrices

$$\hat{E} = \mathcal{W}^T E \mathcal{V}, \quad \hat{A} = \mathcal{W}^T A \mathcal{V}, \quad \hat{B} = \mathcal{W}^T B, \quad \hat{C} = C \mathcal{V}.$$

This is a general description of one of the "intrusive" methods based on projection techniques, which necessitates a full description of the original system. In the next subsection, we briefly describe the Loewner framework as a method for solving rationale interpolation problem, as reported in [21].

**2.2. Interpolatory reduction method.** Assume that we are dealing with the LTI system (2.1) with  $p = q > 1$ , i.e., MIMO (Multiple Input Multiple Output) system. The interpolatory reduction problem [22] goes as follows.

Given left and right interpolation points with left and right tangential directions respectively by  $\{\mu_i\}_{i=1}^l \in \mathbb{C}$ ,  $\{\lambda_j\}_{j=1}^r \in \mathbb{C}$ ,  $\{\mathbf{l}_i\}_{i=1}^l \in \mathbb{C}^p$  and  $\{\mathbf{r}_j\}_{j=1}^r \in \mathbb{C}^q$ . These interpolation points and vector directions are used to describe some left and right response measurements, given by  $\{\mathbf{1}_i^T H(\mu_i) = v_i\}_{i=1}^l$

and  $\{H(\lambda_j)\mathbf{r}_j = w_j\}_{j=1}^r$ . As in [21], we seek out the reduced system (2.2) where its transfer function  $\hat{H}(s)$  could be considered as an approximate interpolant to  $H(s)$ , which means that

$$\mathbf{1}_i^T H(\mu_i) = \mathbf{1}_i^T \hat{H}(\mu_i) = v_i, \quad i = 1, \dots, l, \quad \text{and} \quad H(\lambda_j)\mathbf{r}_j = \hat{H}(\lambda_j)\mathbf{r}_j = w_j, \quad j = 1, \dots, r. \quad (2.4)$$

We consider the function  $\Phi(s) = (sE - A)^{-1}$  and define the generalized controllability and observability matrices  $\mathcal{R}$  and  $\mathcal{O}$  in terms of the original system matrices of (2.1)  $\Phi$ ,  $\mu_i$ ,  $\lambda_j$  and directions  $\mathbf{l}_i$ ,  $\mathbf{r}_j$ . These two matrices are given by

$$\mathcal{R} = [\Phi(\lambda_1)B\mathbf{r}_1, \dots, \Phi(\lambda_l)B\mathbf{r}_l] \in \mathbb{C}^{n \times l}, \quad \mathcal{O} = \begin{bmatrix} \mathbf{1}_1 C \Phi(\mu_1) \\ \vdots \\ \mathbf{1}_r C \Phi(\mu_r) \end{bmatrix} \in \mathbb{C}^{r \times n}, \quad (2.5)$$

LEMMA 1. *Assume that  $r = l$  and choose the projection matrices, defined in the previous section, as the controllability and observability matrices; i.e.  $\mathcal{V} = \mathcal{R}$  and  $\mathcal{W} = \mathcal{O}$ . Then we can construct a projected (reduced) system of order  $r$  satisfying the following interpolation conditions*

$$H(\lambda_j)\mathbf{r}_j = \hat{H}(\lambda_j)\mathbf{r}_j, \quad j = 1, \dots, r, \quad \text{and} \quad (2.6)$$

$$\mathbf{l}_i^T H(\mu_i) = \mathbf{l}_i^T \hat{H}(\mu_i), \quad i = 1, \dots, r. \quad (2.7)$$

*Proof.* We will describe the process to get the first interpolation condition (2.6), and in a similar way we can get the one in (2.7). Let us prove that

$$\hat{\Phi}(\lambda_j)\hat{B}\mathbf{r}_j = e_j, \quad \text{for } j = 1, \dots, r,$$

where  $\hat{\Phi}(s) = (s\hat{E} - \hat{A})^{-1}$ . Notice first that

$$\hat{\Phi}(s)^{-1} = s\hat{E} - \hat{A} = \mathcal{O}(sE - A)\mathcal{R} = \mathcal{O}\Phi(s)^{-1}\mathcal{R}.$$

On the other hand, the  $j^{\text{th}}$  column of the matrix  $\mathcal{R}$  is given by  $\mathcal{R}e_j = \Phi(\lambda_j)B\mathbf{r}_j$ , and then  $\Phi(\lambda_j)^{-1}\mathcal{R}e_j = B\mathbf{r}_j$ . Multiplying the last expression, from the left by  $\mathcal{O}$ , it follows that

$$\mathcal{O}\Phi(\lambda_j)^{-1}\mathcal{R}e_j = \mathcal{O}B\mathbf{r}_j \Rightarrow \hat{\Phi}(\lambda_j)^{-1}e_j = \hat{B}\mathbf{r}_j,$$

Then, multiplying this last equation from the left by  $\hat{C}$ , it follows that  $\hat{C}\hat{\Phi}(\lambda_j)\hat{B}\mathbf{r}_j = \hat{C}e_j$  then  $\hat{H}(\lambda_j)\mathbf{r}_j = C\mathcal{R}e_j$  and finally  $\hat{H}(\lambda_j)\mathbf{r}_j = C\Phi(\lambda_j)B\mathbf{r}_j = H(\lambda_j)\mathbf{r}_j$ .  $\square$

REMARK 1. *It is well known that the transfer function of an LTI system is defined using the system matrices of the associated dynamical system and vice versa. This is actually the reason behind choosing the Loewner framework as a process, since this framework seeks to find a rational function that approximates/interpolates the data. Our primary goal is to find a transfer rational function by means of Loewner framework, which guarantees that this rational function is indeed the one associated to the desired reduced model. We always suppose that we do not have access to the structure matrices of the original system. Our contribution consists in using Krylov-subspace projection techniques via Loewner framework to exploit models that can represent accurately the data. In our case the data is described by some frequency domain measurements.*

Originally, this framework has been addressed to the rational approximation problem as reported in [21, 22]. Assuming that we have a pairs of points represented as

$$(\phi_k, \mathbf{H}_k), \quad k = 1, \dots, N,$$

where  $\mathbf{H}_k \in \mathbb{C}^{p \times q}$  are the transfer function measurements corresponding to the frequency  $\phi_k$ , for a system of  $q$  inputs and  $p$  outputs with  $q, p \ll N$ . The main goal in a rational approximation problem is to find a rational transfer function  $\mathcal{H}(s)$  that approximates the data

$$\mathbf{H}_k \approx \mathcal{H}(s = i\omega_k) \quad \text{where } \omega_k = 2\pi\phi_k \text{ and } i^2 = -1, \quad k = 1, \dots, N,$$

in a way that, in some norm, the transfer function  $\mathcal{H}(s = i\omega_k)$  should be closed to the corresponding measurements  $\mathbb{H}_k$ . We assume here that our transfer function satisfies the complex conjugate condition  $\mathbb{H}(\bar{s}) = \overline{\mathbb{H}(s)}$  and we consider the following form of our data

$$(i\omega_k, -i\omega_k, \mathbb{H}_k, \overline{\mathbb{H}_k}), \quad k = 1, \dots, N. \quad (2.8)$$

A crucial step in the Loewner framework is the partition of data into two disjoint sets [21], as it has a direct impact on the quality of the constructed reduced model. In what follows, we summarize the most natural classical known partitions

- **Half & Half (H&H)** : the first set contains the first half of  $i\omega_k$  together with their conjugates and the other set contains the second half of  $i\omega_k$  together with their conjugates

$$\underbrace{\{i\omega_1, -i\omega_1, \dots, i\omega_{N/2}, -i\omega_{N/2}\}}_{\mu_i, \text{ for } i=1, \dots, N \text{ (left points)}} \cup \underbrace{\{i\omega_{N/2+1}, -i\omega_{N/2+1}, \dots, i\omega_N, -i\omega_N\}}_{\lambda_j, \text{ for } j=1, \dots, N \text{ (right points)}},$$

with the corresponding frequency response measurements

$$\{\mathbb{H}_1, \overline{\mathbb{H}_1}, \dots, \mathbb{H}_{N/2}, \overline{\mathbb{H}_{N/2}}\} \cup \{\mathbb{H}_{N/2+1}, \overline{\mathbb{H}_{N/2+1}}, \dots, \mathbb{H}_N, \overline{\mathbb{H}_N}\}.$$

- **Odd & Even (O&E)** : the first set contains data with odd indices and the other set contains the even indices

$$\underbrace{\{i\omega_1, -i\omega_1, \dots, i\omega_{N-1}, -i\omega_{N-1}\}}_{\mu_i, \text{ for } i=1, \dots, N \text{ (left points)}} \cup \underbrace{\{i\omega_2, -i\omega_2, \dots, i\omega_N, -i\omega_N\}}_{\lambda_j, \text{ for } j=1, \dots, N \text{ (right points)}},$$

with the corresponding frequency response measurements

$$\{\mathbb{H}_1, \overline{\mathbb{H}_1}, \dots, \mathbb{H}_{N-1}, \overline{\mathbb{H}_{N-1}}\} \cup \{\mathbb{H}_{N/2+1}, \overline{\mathbb{H}_{N/2+1}}, \dots, \mathbb{H}_N, \overline{\mathbb{H}_N}\}.$$

**REMARK 2.** In [13], the authors explained that the partition **Half & Half (H&H)** has more effect on the decay of the singular values of the Loewner matrix than the partition **Odd & Even (O&E)**, even for noisy data. Therefore, a truncation index "r" from the SVD of  $\mathbb{L}$ , used to construct a r-dimension reduced model, is found easily by means of the **Half & Half (H&H)** partition. Both partitions will be used in the numerical section.

We have mentioned before, that the transfer function measurements  $\mathbb{H}_k$  are complex  $p \times q$  matrices and then, in step two of the loewner framework, chosen tangential directions are used to rewrite the data matrices (measurements) [21]. We consider  $\mathbf{l}_i \in \mathbb{C}^{1 \times p}$  and  $\mathbf{r}_j \in \mathbb{C}^q$  as a left and right tangential directions respectively, which can be chosen in a heuristic way as columns/rows of the identity matrix[21]. The left and right vectors of data could be defined as

$$\mathbf{l}_i \mathbb{H}_i = \mathbf{v}_i \in \mathbb{C}^q, \quad \mathbb{H}_j \mathbf{r}_j = \mathbf{w}_j \in \mathbb{C}^p, \quad i, j = 1, \dots, N.$$

Now, after partitioning the data and defining the left and right vectors, we compute Loewner and shifted Loewner matrices ( $\mathbb{L}$  and  $\mathbb{L}_s \in \mathbb{C}^{N \times N}$ ) as follows ([21, 22, 25])

$$[\mathbb{L}]_{i,j} = \frac{\mathbf{v}_i \mathbf{r}_j - \mathbf{l}_i \mathbf{w}_j}{\mu_i - \lambda_j} \in \mathbb{C}, \quad i, j = 1, \dots, N,$$

and

$$[\mathbb{L}_s]_{i,j} = \frac{\mu_i \mathbf{v}_i \mathbf{r}_j - \lambda_j \mathbf{l}_i \mathbf{w}_j}{\mu_i - \lambda_j} \in \mathbb{C}, \quad i, j = 1, \dots, N.$$

Using the definition of  $v_i$  and  $w_j$  given in Subsection 2.2, we get

$$[\mathbb{L}]_{i,j} = -\mathbf{l}_i C(\mu_i E - A)^{-1} E(\lambda_j E - A)^{-1} B \mathbf{r}_j, \quad i, j = 1, \dots, N,$$

and then

$$\mathbb{L} = - \begin{bmatrix} \mathbf{l}_1^T C \Phi(\mu_1), \\ \vdots \\ \mathbf{l}_N^T C \Phi(\mu_N) \end{bmatrix} E [\Phi(\lambda_1) B \mathbf{r}_1, \dots, \Phi(\lambda_N) B \mathbf{r}_N].$$

It is mentioned in [22], that besides the fact that the pairs  $(E, A, B)$  and  $(C, E, A)$  are controllable and observable, respectively, if the directions  $\mathbf{l}_i$  and  $\mathbf{r}_j$  are chosen properly, the rank of the Loewner matrix  $\mathbb{L}$  is equal to the rank of the matrix  $E$ . The data  $(\mu_i, \lambda_j, v_i, w_j)$  can be presented in the following matrix format

$$\underbrace{\Lambda = \text{diag}([\lambda_1, \dots, \lambda_N]) \in \mathbb{C}^{N \times N}, \quad M = \text{diag}([\mu_1, \dots, \mu_N]) \in \mathbb{C}^{N \times N}}_{N \times N \quad \text{diagonal matrices contains the left and right points}}, \quad (2.9)$$

$$\underbrace{L = \begin{bmatrix} \mathbf{l}_1 \\ \vdots \\ \mathbf{l}_N \end{bmatrix} \in \mathbb{C}^{N \times p}, \quad R = [\mathbf{r}_1, \dots, \mathbf{r}_N] \in \mathbb{C}^{q \times N}}_{\text{left and right tangential directions}}. \quad (2.10)$$

The left and right vectors are as follows

$$W = [w_1, \dots, w_N] \in \mathbb{C}^{p \times N}, \quad V = \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix} \in \mathbb{C}^{N \times q}. \quad (2.11)$$

The Loewner and shifted Loewner matrices have the following matrix format

$$\mathbb{L} = \begin{bmatrix} \frac{v_1 \mathbf{r}_1 - \mathbf{l}_1 w_1}{\mu_1 - \lambda_1} & \cdots & \frac{v_1 \mathbf{r}_N - \mathbf{l}_1 w_N}{\mu_1 - \lambda_N} \\ \vdots & \ddots & \vdots \\ \frac{v_N \mathbf{r}_1 - \mathbf{l}_N w_1}{\mu_N - \lambda_1} & \cdots & \frac{v_N \mathbf{r}_N - \mathbf{l}_N w_N}{\mu_N - \lambda_N} \end{bmatrix}, \quad \mathbb{L}_s = \begin{bmatrix} \frac{\mu_1 v_1 \mathbf{r}_1 - \lambda_1 \mathbf{l}_1 w_1}{\mu_1 - \lambda_1} & \cdots & \frac{\mu_1 v_1 \mathbf{r}_N - \lambda_1 \mathbf{l}_1 w_N}{\mu_1 - \lambda_N} \\ \vdots & \ddots & \vdots \\ \frac{\mu_N v_N \mathbf{r}_1 - \lambda_N \mathbf{l}_N w_1}{\mu_N - \lambda_1} & \cdots & \frac{\mu_N v_N \mathbf{r}_N - \lambda_N \mathbf{l}_N w_N}{\mu_N - \lambda_N} \end{bmatrix}. \quad (2.12)$$

It is straightforward to check that the two Loewner matrices verify the following Sylvester matrix equations

$$\underbrace{M\mathbb{L} - \mathbb{L}\Lambda = VR - LW}_{\text{equation 1}}, \quad \underbrace{M\mathbb{L}_s - \mathbb{L}_s\Lambda = MVR - LW\Lambda}_{\text{equation 2}}, \quad (2.13)$$

as well as

$$\mathbb{L}_s - \mathbb{L}\Lambda = VR, \quad \mathbb{L}_s - M\mathbb{L} = LW. \quad (2.14)$$

Notice here, that if we don't consider the left and right tangential directions in the above process, then the components of Loewner and shifted Loewner matrices would be presented as follows [18]

$$[\mathbb{L}]_{i,j} = \frac{H_i^\mu - H_j^\lambda}{\mu_i - \lambda_j} \in \mathbb{C}^{p \times q}, \quad [\mathbb{L}_s]_{i,j} = \frac{\mu_i H_i^\mu - \lambda_j H_j^\lambda}{\mu_i - \lambda_j} \in \mathbb{C}^{p \times q}, \quad i, j = 1, \dots, N,$$

where  $H_i^\mu, H_j^\lambda$  are the transfer functions measurements corresponding to the left and right points  $\mu_i$  and  $\lambda_i$ , respectively and satisfying the following Sylvester equations

$$M\mathbb{L} - \mathbb{L}\Lambda = EF^*, \quad M\mathbb{L}_s - \mathbb{L}_s\Lambda = GK^*,$$

where  $M \in \mathbb{C}^{(Np) \times (Np)}$  and  $\Lambda \in \mathbb{C}^{(Nq) \times (Nq)}$  are diagonal matrices and their diagonals are defined as

$$\text{diag}(M) = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_N \end{bmatrix} \otimes \mathbf{1}_p \in \mathbb{C}^{(Np) \times 1} \text{ and } \text{diag}(\Lambda) = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{bmatrix} \otimes \mathbf{1}_q \in \mathbb{C}^{(Nq) \times 1}, \text{ } \mathbf{1}_p \text{ or } \mathbf{1}_q \text{ refers to the } p \text{ or } q$$

dimension columns of ones. The right hand sides are presented as follows

$$E = \begin{bmatrix} H_1^\mu & -I_p \\ H_2^\mu & -I_p \\ \vdots & -I_p \\ H_N^\mu & -I_p \end{bmatrix} \in \mathbb{C}^{N \times (p+q)}, \quad F^* = \begin{bmatrix} I_q & I_q & \dots & I_q \\ H_1^\lambda & H_2^\lambda & \dots & H_N^\lambda \end{bmatrix} \in \mathbb{C}^{(p+q) \times N}$$

$$G = \begin{bmatrix} \mu_1 H_1^\mu & -I_p \\ \mu_2 H_2^\mu & -I_p \\ \vdots & -I_p \\ \mu_N H_N^\mu & -I_p \end{bmatrix} \in \mathbb{C}^{N \times (p+q)}, \text{ and } K^* = \begin{bmatrix} I_q & I_q & \dots & I_q \\ \lambda_1 H_1^\lambda & \lambda_2 H_2^\lambda & \dots & \lambda_N H_N^\lambda \end{bmatrix} \in \mathbb{C}^{(p+q) \times N}.$$

**2.3. The Loewner processing.** After the step of partitioning the data and introducing the Loewner matrices, we can now obtain a minimal realization by assuming that the data is not redundant

$$\hat{E} = -\mathbb{L}, \quad \hat{A} = -\mathbb{L}_s, \quad \hat{B} = V \text{ and } \hat{C} = W.$$

Here, we have to mention that no reduction was performed to build a model that approximates the data, only the original amount of data was chosen. A reduction process could be established by performing a singular value decomposition (SVD) of Loewner matrices. As it is described in [4, 22], we employed one of the singular value decompositions

$$[\mathbb{L} \quad \mathbb{L}_s] = Y_1 \Sigma_1 X_1^*, \text{ or } \begin{bmatrix} \mathbb{L} \\ \mathbb{L}_s \end{bmatrix} = Y_2 \Sigma_2 X_2^*.$$

Then we chose a suitable reduction order or truncation index  $r$  [21, 22] to get the projection matrices  $Y_r$  and  $X_r$  in order to build a reduced model of order  $r$ . In this work, we will not focus on the polynomial  $D$ -term defined in the description (2.1). The projection matrices  $Y_r$  and  $X_r$  are chosen by selecting the first  $r$  columns of  $Y$  and  $X$ , respectively from the following SVD of  $\mathbb{L}$

$$\mathbb{L} = Y \Sigma X^* \approx Y_r \Sigma_r X_r^*. \quad (2.15)$$

Finally, an  $r$ -order realization of an interpolant of the data can be constructed, and its system matrices is given as follows

$$\hat{E} = -Y_r^* \mathbb{L} X_r, \quad \hat{A} = -Y_r^* \mathbb{L}_s X_r, \quad \hat{B} = Y_r^* V \text{ and } \hat{C} = W X_r. \quad (2.16)$$

We recall that data-driven methods aim to build accurate reduced models from data presented by frequencies and transfer function measurements (2.8) as mentioned earlier, no exact access to the system matrices as in (2.1) is provided. The main goal is to build a reduced system whose rational transfer function interpolates the data. A solution to this problem is provided by the Loewner framework [22],

<sup>1</sup>The superscript  $*$  stands for the complex transpose conjugate, i.e,  $F^* = \overline{F^T}$ .



and the following matrices  $E = -\mathbb{L}$ ,  $A = -\mathbb{L}_s$ ,  $B = V$  and  $C = W$  form a minimal realization of an interpolant of the data, whose rational transfer function is given by

$$\mathcal{H}(s) = W(\mathbb{L}_s - s\mathbb{L})^{-1}V,$$

and interpolates the data as follows

$$\mathcal{H}(\lambda_i)\mathbf{r}_i = W(\mathbb{L}_s - \lambda_i\mathbb{L})^{-1}V\mathbf{r}_i = \mathbf{H}_i\mathbf{r}_i = w_i, \text{ for } i = 1, \dots, N, \quad (2.17)$$

$$\mathbf{1}_j\mathcal{H}(\mu_j) = \mathbf{1}_jW(\mathbb{L}_s - \mu_j\mathbb{L})^{-1}V = \mathbf{1}_j\mathbf{H}_j = v_j, \text{ for } j = 1, \dots, N. \quad (2.18)$$

LEMMA 2. Assume that the matrix pencil  $(\mathbb{L}_s, \mathbb{L})$  is regular ( $\det(s\mathbb{L} - \mathbb{L}_s) \neq 0 \forall s \in \{\mu_i\} \cup \{\lambda_j\}$ ), then (2.17) and (2.18) are satisfied.

*Proof.* Let us check the interpolation condition (2.17). We can easily verify that

$$\begin{aligned} (\mathbb{L}_s - \lambda_i\mathbb{L})e_i &= \mathbb{L}_s e_i - \lambda_i\mathbb{L}e_i, \\ &= \begin{bmatrix} \frac{\mu_1 v_1 \mathbf{r}_i - \lambda_i \mathbf{1}_1 w_i}{\mu_1 - \lambda_i} \\ \vdots \\ \frac{\mu_N v_N \mathbf{r}_i - \lambda_i \mathbf{1}_N w_i}{\mu_N - \lambda_i} \end{bmatrix} - \lambda_i \begin{bmatrix} \frac{v_1 \mathbf{r}_i - \mathbf{1}_1 w_i}{\mu_1 - \lambda_i} \\ \vdots \\ \frac{v_N \mathbf{r}_i - \mathbf{1}_N w_i}{\mu_N - \lambda_i} \end{bmatrix} = \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix} \mathbf{r}_i = V\mathbf{r}_i. \end{aligned}$$

Multiplying the last expression, from the left by  $(\mathbb{L}_s - \lambda_i\mathbb{L})^{-1}$ , we get

$$e_i = (\mathbb{L}_s - \lambda_i\mathbb{L})^{-1}V\mathbf{r}_i.$$

Then we obtain

$$w_i = We_i = W(\mathbb{L}_s - \lambda_i\mathbb{L})^{-1}V\mathbf{r}_i = H(\lambda_i)\mathbf{r}_i,$$

which corresponds to the result (2.17). To get (2.18), we first compute  $e_j^T(\mathbb{L}_s - \mu_j\mathbb{L})$  and then the same proof as above could be used.  $\square$

We notice that the Loewner matrices  $(\mathbb{L}$  and  $\mathbb{L}_s \in \mathbb{C}^{N \times N})$ , as well as the singular value decomposition of  $\mathbb{L}$ , play a fundamental role in the Loewner framework, in the realization of reduced model without requiring the availability of any system matrices of the original one, except the availability of measurement data (*frequencies and transfer function measurements*), and this actually the motivation behind *data-driven* model reduction. However, for large amount of data  $N$ , the SVD of  $\mathbb{L} \in \mathbb{C}^{N \times N}$  requires a prohibitive computational cost, and to solve this problem, we exploit the fact that  $\mathbb{L}$  fulfils the Sylvester equation (2.13) with diagonal coefficient matrices. Our attention is focused on finding an approximate solution by means of numerical linear algebra tools that will ensure proper computation of the SVD and not the solution of the equation (3.2) itself. In this paper, we focused on a low-rank approximate solution based on Extended bloc Krylov-subspace.

**3. Approximate solution to large Sylvester matrix equations.** An extended bloc Arnoldi-based method to find an approximate solution, for large scale Sylvester matrix equations is described in this section. This Krylov-subspace method has proven its efficiency for many problems, such as solving Lyapunov and Riccati equations [17, 27, 28], and also for model order reduction as reported in [5, 7]. We recall that our focus is not only on using this tool to solve the two large scale Sylvester equations (2.13), since those two equations are with diagonal coefficient matrices, and one can use a direct solver as Hessenberg-Schur method [10] that provides a good results, but the actual goal is to find an approximate solution  $\tilde{\mathbb{L}}$  to  $\mathbb{L}$  and use it instead to compute the SVD which will then be used to build the reduced model. To understand more about our contribution, let us assume that an

approximate solution  $\tilde{\mathbb{L}}$  to  $\mathbb{L}$  has been computed by using a numerical Krylov-subspace method, then as we know, and according to [1, 19], the approximate solution has the following form

$$\mathbb{L} \approx \tilde{\mathbb{L}} = \mathcal{V}_m Y_m \mathcal{W}_m^*,$$

where  $\mathcal{V}_m, \mathcal{W}_m \in \mathbb{C}^{N \times m}$  are some basis-matrices of specific Krylov-subspaces, and  $Y_m \in \mathbb{C}^{m \times m}$  is a small matrix ( $m \ll N$ ). To get the SVD of  $\mathbb{L}$ , we first compute the SVD of the small matrix  $[P, \Sigma, Q] = \text{svd}(Y_m)$ , and then the SVD of  $\mathbb{L}$  goes as

$$\mathbb{L} \approx \mathcal{V}_m P \Sigma (\mathcal{W}_m Q)^*. \quad (3.1)$$

In what follows, we present briefly, a projection method based on an extended Krylov-subspace for constructing approximate solutions  $\tilde{\mathbb{L}}$ . We recall our large scale Sylvester matrix equation (2.13)

$$MX - X\Lambda = VR - LW = \begin{bmatrix} V & -L \end{bmatrix} \begin{bmatrix} R \\ W \end{bmatrix} = FG^*, \quad (3.2)$$

where the matrices  $M, \Lambda \in \mathbb{C}^{N \times N}$ ,  $L \in \mathbb{C}^{N \times p}$ ,  $R \in \mathbb{C}^{q \times N}$  and  $W \in \mathbb{C}^{p \times N}$ ,  $V \in \mathbb{C}^{N \times q}$  are presented in (2.9), (2.10) and (2.11), respectively. The solution  $X$  of Eq. (3.2) exists and is unique [10] since  $\lambda_i(M) - \lambda_j(\Lambda) \neq 0$ , for  $i, j = 1, \dots, N$ , where  $\lambda_i(M)$  denotes the  $i$ th eigenvalue of the matrix  $M$ . Note that the Sylvester equation (3.2) can be written as an  $N^2 \times N^2$  system of linear equations  $(I_N \otimes M - \Lambda^* \otimes I_N) \text{vec}(X) = \text{vec}(FG^*)$ , where  $\otimes$  denotes the Kronecker product and  $\text{vec}$  is a linear operator defined as  $\text{vec}(X) = [X_{11}, \dots, X_{1N}, X_{21}, \dots, X_{2N}, \dots, X_{N1}, \dots, X_{NN}]^T \in \mathbb{C}^{N \times N}$ , obtained by stacking the rows of the  $N \times N$  matrix  $X$ .

**3.1. Extended block Krylov-Subspace.** Let  $Q \in \mathbb{R}^{N \times N}$  and  $J \in \mathbb{R}^{N \times s}$ , the extended block Krylov-subspace [27] can be defined as the union of two classical Krylov subspaces as follows

$$\mathcal{K}_m^{\text{ext}}(Q, J) = \mathcal{K}_m(Q, J) \cup \mathcal{K}_m(Q^{-1}, Q^{-1}J),$$

where  $\mathcal{K}_m(Q, J)$  is the subspace spanned by the columns of  $J, QJ, \dots, Q^{m-1}J$ , i.e.,

$$\mathcal{K}_m(Q, J) = \text{colspan}\{J, QJ, \dots, Q^{m-1}J\}.$$

The extended block Krylov-subspace  $\mathcal{K}_m^{\text{ext}}(Q, J)$  can be seen as

$$\mathcal{K}_m^{\text{ext}}(Q, J) = \text{colspan}\{Q^{-m}J, \dots, Q^{-1}J, J, QJ, \dots, Q^{m-1}J\}. \quad (3.3)$$

In the following algorithm, we summarize the extended block Arnoldi process to built an orthonormal basis of the extended block Krylov-subspace  $\mathcal{K}_m^{\text{ext}}(Q, J)$ .

---

**Algorithm 1** The extended block Arnoldi algorithm

---

- Input :  $Q \in \mathbb{R}^{N \times N}$ ,  $J \in \mathbb{R}^{N \times s}$  and a fixed integer  $m$ .
  - Compute  $[V_1, \Gamma^Q] = \text{qr}([J, Q^{-1}J], 0)$  (skinny qr),  $\mathbb{V}_1 = [V_1]$ .
  - for  $j = 1 \dots m - 1$ 
    1. Set  $V_j^{(1)}$ : first  $s$  columns of  $V_j$ ;  $V_j^{(2)}$ : second  $s$  columns of  $V_j$ .
    2.  $\tilde{V}_{j+1} = [Q V_j^{(1)}, Q^{-1} V_j^{(2)}]$ .
    3. Orthogonalization step:
      - for  $i = 1, 2 \dots j$ 

$$H_{ij} = V_i^T \tilde{V}_{j+1}.$$

$$\tilde{V}_{j+1} = \tilde{V}_{j+1} - V_i H_{ij}.$$
  - $\text{qr}(\tilde{V}_{j+1}) = V_{j+1} H_{j+1,j}$  ( $qr$  decomposition).
  - $\mathbb{V}_{j+1} = [V_j, V_{j+1}]$ .
  - End.
-

After  $m$  steps, Algorithm 1 builds a matrix  $\mathbb{V}_m = [V_1, \dots, V_m] \in \mathbb{R}^{N \times 2ms}$  corresponding to the orthonormal basis of the extended block Krylov subspace and a block upper Hessenberg matrix  $\mathbb{H}_m \in \mathbb{R}^{2ms \times 2ms}$  whose non-zeros blocks are the  $H_{i,j}$ . Recall that the columns of  $\mathbb{V}_m$  are mutually orthogonal since we use a Gram-Schmidt process at *step3* in Algorithm 1. Note that each sub-matrix  $H_{i,j}$  ( $1 \leq i \leq j \leq m$ ) is of order  $2s \times 2s$ . Let  $\mathbb{T}_m = \mathbb{V}_m^T Q \mathbb{V}_m \in \mathbb{R}^{2ms \times 2ms}$  denotes the restriction of the matrix  $Q$  to the extended block Krylov subspace. It was shown in [27] that  $\mathbb{T}_m$  is also a block upper Hessenberg matrix. Assume that  $m$  steps of Algorithm 1 have been run, then we get the following classical algebraic relations

$$Q \mathbb{V}_m = \mathbb{V}_{m+1} \bar{\mathbb{T}}_m \quad (3.4)$$

$$= \mathbb{V}_m \mathbb{T}_m + V_{m+1} T_{m+1,m} E_m^T, \quad (3.5)$$

where  $\bar{\mathbb{T}}_m = \mathbb{V}_{m+1}^T Q \mathbb{V}_m \in \mathbb{R}^{2(m+1)s \times 2ms}$  and  $E_m$  is the last  $2s$  columns of the identity  $I_{2ms}$ . The upper block Hessenberg matrices  $\bar{H}_m$  and  $\bar{T}_m \in \mathbb{R}^{2(m+1)s \times 2ms}$  are partitioned as follows

$$\bar{H}_m = [h_{:,1}, \dots, h_{:,2m}] \text{ and } \bar{T}_m = [t_{:,1}, \dots, t_{:,2m}],$$

where  $h_{:,j}$  and  $t_{:,j}$  are the  $j^{em}$  block column of order  $s$  of  $\bar{H}_m \in \mathbb{R}^{2(m+1)s \times 2ms}$  and  $\bar{T}_m \in \mathbb{R}^{2(m+1)s \times 2ms}$ , respectively. Also, we suppose that  $\Gamma^Q$  and  $H_{j+1,j}$  are decomposed as

$$\Gamma^Q = \begin{bmatrix} \Gamma_{11}^Q & \Gamma_{12}^Q \\ 0 & \Gamma_{22}^Q \end{bmatrix} \quad \text{and} \quad H_{j+1,j} = \begin{bmatrix} H_{j+1,j}^{(11)} & H_{j+1,j}^{(12)} \\ 0 & H_{j+1,j}^{(22)} \end{bmatrix}.$$

The following proposition allows a simple computation of the columns  $t_{:,i}$  of  $\bar{T}_m$  using only columns  $h_{:,j}$  of  $\bar{H}_m$  without involving any matrix-vector products with  $Q$  or extra inner products of long vectors.

PROPOSITION 3.1. ([27]) For  $j = 1, \dots, m$ , the odd block column are given as follows

$$t_{:,2j-1} = h_{:,2j-1},$$

while the even block column are such that

$$t_{:,2} = [\tilde{e}_1 \Gamma_{11}^Q - t_{:,1} \Gamma_{12}^Q] (\Gamma_{22}^Q)^{-1}, \quad (3.6)$$

$$t_{:,2j+2} = [\tilde{e}_{2j} - t_{:,1:2j+1} h_{1:2j+1,2j}] (H_{j+1,j}^{(22)})^{-1}, \quad (3.7)$$

where  $\tilde{e}_i = e_i \otimes I_s$ .

**3.2. A low-rank approximate solution.** In this section, we first describe the extended block Arnoldi process in order to get a low-rank approximate solution to large scale Sylvester matrix equation (3.2), for more details see [1, 19]. In the sequel, and for simplicity, we set  $k = p + q$ , i.e., the numbers of columns of the right-hand sides  $F$  and  $G$ . We apply simultaneously Algorithm 1 to the pairs  $(M, F)$  and  $(\Lambda^*, G^*)$ , and by considering that  $m$  iterations have been achieved, we get  $\mathbb{V}_m^M, \mathbb{V}_m^\Lambda \in \mathbb{C}^{N \times 2mk}$  two orthonormal basis matrices as well as two block Hessenberg matrices  $T_m^M, T_m^\Lambda \in \mathbb{C}^{2mk \times 2mk}$ . The approximate solution is given as

$$X_m = \mathbb{V}_m^M Y_m (\mathbb{V}_m^\Lambda)^*, \quad (3.8)$$

where  $Y_m \in \mathbb{C}^{2mk \times 2mk}$  is a small matrix obtained by enforcing the Galerkin condition

$$(\mathbb{V}_m^M)^* R_m \mathbb{V}_m^\Lambda = 0, \quad (3.9)$$

and  $R_m$  is the residual given by  $R_m = MX_m - X_m\Lambda - FG^*$ . Using (3.8) and (3.9), we get the following low dimensional Sylvester equation associated to  $Y_m$

$$T_m^M Y_m - Y_m T_m^\Lambda = F_m G_m^*, \quad (3.10)$$

with  $F_m = (\mathbb{V}_m^M)^* F$  and  $G_m = (\mathbb{V}_m^\Lambda)^* G$  also given as

$$F_m = E_1^1 \Gamma_{11}^M \quad \text{and} \quad G_m = E_1^1 \Gamma_{11}^\Lambda, \quad (3.11)$$

where  $E_1^1$  is the first  $k$  columns of the  $2mk \times 2mk$  identity matrix, and  $\Gamma_{11}^M, \Gamma_{11}^\Lambda \in \mathbb{C}^{k \times k}$  two matrices obtained from the  $QR$  decomposition of  $[F, M^{-1}F]$  and  $[G, \Lambda^{-1}G]$ , respectively, and given as follows

$$[F, M^{-1}F] = V_1^M \Gamma^M \quad \text{and} \quad [G, \Lambda^{-1}G] = V_1^\Lambda \Gamma^\Lambda,$$

with

$$\Gamma^M = \begin{bmatrix} \Gamma_{11}^M & \Gamma_{12}^M \\ 0 & \Gamma_{22}^M \end{bmatrix} \quad \text{and} \quad \Gamma^\Lambda = \begin{bmatrix} \Gamma_{11}^\Lambda & \Gamma_{12}^\Lambda \\ 0 & \Gamma_{22}^\Lambda \end{bmatrix}.$$

The purpose of the simplified expression (3.11), is to make the calculations much easier for solving the low dimensional equation (3.10), by means of a direct method as described in [10]. In the numerical section, we treat this equation using the built-in MATLAB equation `layp`.

The iterations are stopped when the residual in a certain norm  $\|R_m\|$  is less than a chosen tolerance  $\epsilon$  *i.e.*,  $\|R_m\| < \epsilon$ . However, as  $m$  increases and for large scale matrices  $M, \Lambda, F$  and  $G$ , a direct computation of the residual  $R_m$  is very expensive. The next result, shows how to compute the residual norm  $\|R_m\|$  in an efficient way avoiding extra products with the approximation  $X_m$ . We then show that a factorization of  $X_m$  can be established when the convergence is achieved.

**THEOREM 1.** *Assume that  $m$  iterations of Algorithm 1 have been run and let  $X_m = \mathbb{V}_m^M Y_m (\mathbb{V}_m^\Lambda)^*$  be the obtained approximate solution where  $Y_m$  solves the low dimensional Sylvester equation (3.10). Then, the residual norm  $\|R_m\|_F$  can be expressed as*

$$\|R_m\|_F = \sqrt{r_m^1 + r_m^2}, \quad (3.12)$$

where  $r_m^1 = \|T_{m+1,m}^M (E_m^k)^T Y_m\|_F$ ,  $r_m^2 = \|T_{m+1,m}^\Lambda (E_m^k)^T Y_m^\Lambda\|_F$  and  $E_m^k$  is the last  $k$  columns of  $2mk \times 2mk$  identity matrix. The norm  $\|\cdot\|_F$  is the Frobenius norm.

It is necessary here to underline the fact that we may encounter difficulties to compute the approximation  $X_m = \mathbb{V}_m^M Y_m (\mathbb{V}_m^\Lambda)^*$  as  $m$  increases. To solve this problem, we show in the following how to represent the approximate solution  $X_m$  in factored form. In fact, let  $Y_m = P\Sigma Q^*$  be the SVD of  $Y_m$  where  $\Sigma$  is the matrix of the singular values of  $Y_m$  sorted in decreasing order. Then consider some tolerance `dtol` and define  $P_r, Q_r$  as the first  $r$  columns of  $P$  and  $Q$ , respectively corresponding to the  $r$  singular values of magnitude greater than `dtol`. Setting  $\Sigma_r = [\sigma_1, \dots, \sigma_r]$ , we get  $Y_m \approx P_r \Sigma_r Q_r^*$ , and we obtain the desired factorization as

$$X_m \approx Z_m^1 (Z_m^2)^*, \quad (3.13)$$

with  $Z_m^1 = \mathbb{V}_m^M P_r (\Sigma_r)^{1/2}$  and  $Z_m^2 = \mathbb{V}_m^\Lambda Q_r (\Sigma_r)^{1/2}$ . We summarize all the results depicted above in the following algorithm

---

**Algorithm 2** Extended block Arnoldi Sylvester algorithm (EBASA)

---

- Inputs:  $M, \Lambda \in \mathbb{C}^{N \times N}$ ,  $F, G \in \mathbb{C}^{N \times k}$  ( $k = p + q$ ), a tolerance  $\epsilon$  dtol and a number  $m_{max}$  of maximum iterations.
  - Outputs: the approximate solution  $X_m \approx Z_m^1 (Z_m^2)^*$ .
  - For  $m = 1, \dots, m_{max}$
  - Use Algorithm 1 to built  $\mathbb{V}_m^M, \mathbb{V}_m^\Lambda$  an orthonormal matrices and  $\mathbb{T}_m^M, \mathbb{T}_m^\Lambda$  two block Hessenberg matrices.
  - Solve the low-dimensional Sylvester equation (3.10) using the MATLAB function `lyap`.
  - Compute the residual norm  $\|R_m\|$  using (3.12), and if it is less than  $\epsilon$ , then
    1. compute the SVD of  $Y_m = U\Sigma V$  where  $\Sigma = \text{diag}[\sigma_1, \dots, \sigma_{2m}]$ ,
    2. determine  $r$  such that  $\sigma_{r+1} < \text{dtol} \leq \sigma_r$ , set  $\Sigma_r = \text{diag}[\sigma_1, \dots, \sigma_r]$  and compute  $Z_m^1 = \mathbb{V}_m^M P_r (\Sigma_r)^{1/2}$  and  $Z_m^2 = \mathbb{V}_m^\Lambda Q_r (\Sigma_r)^{1/2}$ ,
end if.
  - End For
- 

Notice that  $m$  is the number of iterations performed by Algorithm 2 to built the approximate solutions, particularly  $\mathbb{L} \approx \mathbb{V}_m^M Y_m (\mathbb{V}_m^\Lambda)^*$ . We denote by  $r \leq m \ll N$  the truncated number from the SVD of  $Y_m$  in order to build an  $r$ -dimensional reduced model.

As mentioned earlier, the computation of the Loewner and shifted Loewner matrices is of primary importance in our approach, also the singular value decomposition of  $\mathbb{L}$  is crucial for building a reduced model from original data without requiring any differential equations. When we deal with small or medium number of data, a direct computation of  $\mathbb{L}, \mathbb{L}_s$  and the SVD of  $\mathbb{L}$  can be performed efficiently up to medium sizes. For large problems, the process of data-driven model reduction might require a prohibitive computational cost. However, and thanks to the extended block Arnoldi Sylvester algorithm (EBASA), we can obtain good approximations to the Loewner and shifted Loewner matrices  $\mathbb{L}, \mathbb{L}_s$  in appropriate manner and we can rewrite these approximations in a factored form to reduce storage and computation required during the process. In the following, we resume the process of our method based on data-driven Loewner framework via the extended bloc Krylov subspace technique.

---

**Algorithm 3** Loewner framework via Krylov subspace technique

---

- Input :  $\{\omega_j = 2\pi\phi_j \in \mathbb{C}, H_j \in \mathbb{C}^{p \times q}\}_{j=1}^N$  (measurements).
  1. Splitting data  $\{\omega_j, H(i\omega_j)\}_{j=1}^N$  into left and right data, and then setting up the matrices  $\{M, \Lambda, R, L, V, W\}$  as described in Section 2.
  2. Solve the two large scale Sylvester equations using Algorithm 2, and get approximate solutions to  $\mathbb{L}$  and  $\mathbb{L}_s$  are constructed.
  3. Compute the SVD of  $Y_m$ , i.e.,  $Y_m = P\Sigma Q$ . In the following,  $P_r$  and  $Q_r$  refer to the first  $r$  columns of  $P$  and  $Q$ .
- Output : The system matrices of the reduced order model

$$E_r = -\mathcal{Y}_r^* \mathbb{L}_1 (\mathbb{L}_2^* \mathcal{X}_r), \quad A_r = -\mathcal{Y}_r^* \mathbb{L}_s^1 ((\mathbb{L}_s^2)^* \mathcal{X}_r), \quad B_r = \mathcal{Y}_r^* V \text{ and } C_r = W \mathcal{X}_r,$$

where  $\mathcal{Y}_r = \mathbb{V}_r^M P_r$ ,  $\mathcal{X}_r = \mathbb{V}_r^\Lambda Q_r$ , and we used the technique described in (3.13) to rewrite  $\mathbb{L}$  and  $\mathbb{L}_s$  in a factored form as  $\mathbb{L} \approx \mathbb{L}_1 \mathbb{L}_2^*$  and  $\mathbb{L}_s \approx \mathbb{L}_s^1 (\mathbb{L}_s^2)^*$ .

---

**4. Numerical experiments.** In this section, we evaluate the effectiveness of our Algorithm 3 and report the obtained results compared to other known methods on some numerical tests. We show the accuracy of the obtained reduced models from original data with  $p = q = 1$  (i.e., Single Input Single Output models, (SISO)), and also with  $p, q > 1$  (i.e., Multiple Input Multiple Output models, (MIMO)). All the experiments were carried out using MATLAB R2018a on a computer with Intel<sup>®</sup> core i7 at 2.6GHz and 16Gb of RAM. As described above,  $N$  refers to the cardinality of our data  $\{\omega_j, H_j\}_{j=1}^N$ , with  $\omega_j = 2\pi i\phi_j$ ,  $i^2 = -1$  and  $\phi_j \in \mathbb{R}$  are the frequencies. The matrices  $H_j \in \mathbb{C}^{p \times q}$

represent samples of transfer function matrices. The data was built according to the following:

- The built-in function in MATLAB `logspace` to get the logarithmically spaced points  $\phi_j$ .
- In some examples,  $H_j \in \mathbb{C}^{p \times q}$  are exactly samples of transfer functions from a benchmark examples, and in other examples, we obtained the matrices  $H_j$  from a dynamic system generated from  $n$  random complex poles ( $\sigma_i \in \mathbb{C}$ ) and residues ( $R_i \in \mathbb{C}^{p \times q}$ ).

$$H(s) = \sum_{i=1}^n \frac{R_i}{s - \sigma_i} \in \mathbb{C}^{p \times q}. \quad (4.1)$$

From Algorithm 3, we get the system matrices  $(E_r, A_r, \dots)$  of the desired reduced model, where  $\mathcal{H}_r(s) = C_r(sE_r - A_r)^{-1}B_r$  is the associated transfer function that approximates the data, as it is described in Lemma 2. To measure how well our reduced model can reproduce the behaviour of the original one, we need to plot the following error vector denoted by  $\mathbf{err}_\infty$  where its components are given by

$$[\mathbf{err}_\infty]_k = \sigma_{max}(H_k - \mathcal{H}_r(\omega_k)), \quad \text{for } k = 1, \dots, N,$$

where  $\sigma_{max}(\cdot)$  is the largest singular values of the corresponding  $p \times q$  matrix.

**REMARK 3.** *As mentioned in Algorithm 2,  $k = p + q$  is the column number of the low-rank hand sides of the Sylvester equations (2.13). Therefore, the dimension of the computed extended bloc Krylov-subspace is "2mk", particularly in the SISO case, the dimension is 4m.*

**4.1. SISO-models with small amount of data.** In this numerical part, we give a comparison of our proposed method with Balanced Truncation (BT) [11] that was reformulated to be fitted with the principals of a data-driven framework using some quadrature techniques. We use three benchmark models `CDplayer`, `Heat-cont` and `iss`. As the first two models are originally with Multiple Inputs and Multiple Outputs, we take only the first input and first output to get SISO models. We choose  $N = 400$  logarithmically spaced points in the interval  $[10^{-1}, 10^2]$  for `CDplayer` and `iss` models, and  $N = 120$  points for `heat-cont` model in the interval  $[10^{-1}, 10^3]$ . We solved the two Sylvester equations (2.13) by means of Algorithm 2. The iterations were stopped when  $\|R_m\| < \epsilon$ , with  $\epsilon=1e-5$ . We recall that  $\|\cdot\|$  is the *Frobenius norm*  $\|\cdot\|_F$ . We follow [Algorithm 2 (QuadBT), in [11] p. 10], and report the obtained results for the three models, showing the error norm  $\mathbf{err}_\infty$  by varying the reduction order  $r$  as depicted in Figure 4.1.

**4.2. Comparison with the original Loewner framework.** We follow the main steps described in [22] for the standard data driven approach based on a Loewner framework. The key tool of this approach is based on the SVD of the Loewner matrix  $\mathbb{L}$ . We compared these results with those obtained by our Algorithm 3. We used two benchmark models to generate the frequency response measurements for `beam` and `FOM` models<sup>0</sup>. For `beam`-model, we choose  $N = 1500$  logarithmically spaced points  $\omega \in [10^{-2}, 1]$  scaled by  $2\pi$ , and for `FOM`-model we select  $N = 2500$  logarithmically spaced points  $\omega \in [1, 10^4]$  scaled by  $2\pi$ . In this experiment, we used the partition `H&H` to get the results depicted in Figure 4.2 for `beam`-model and in Figure 4.3 for `FOM`-model. As shown in Figures 4.3 and 4.2, our Algorithm 3 and the process of the original Loewner framework have a similar results, but regarding the total computational time (from generating the data to the constructed model) for `beam`-model, the original Loewner framework needs 18.38sec while our algorithm needs 12.75sec, and concerning the `FOM`-model original Loewner framework needs 28.07sec and our algorithm needs only 1.61sec. Figure 4.2 reports results obtained for the `Beam` model. In the top plot, we compared the original transfer function matrices magnitude (*i.e.*,  $\sigma_{max}(H_k)$ ) with the ones produced by the standard data driven approach (*i.e.*,  $\sigma_{max}(\mathcal{H}_r(\omega_k))$ ), while the second plot gives the error approximation. In the third and fourth plots of Figure 4.2, we compared the results obtained by our proposed method with those of the original data. Figure 4.3 reports results obtained for the `FOM`-model.

<sup>0</sup><http://slicot.org/20-site/126-benchmark-examples-for-model-reduction>

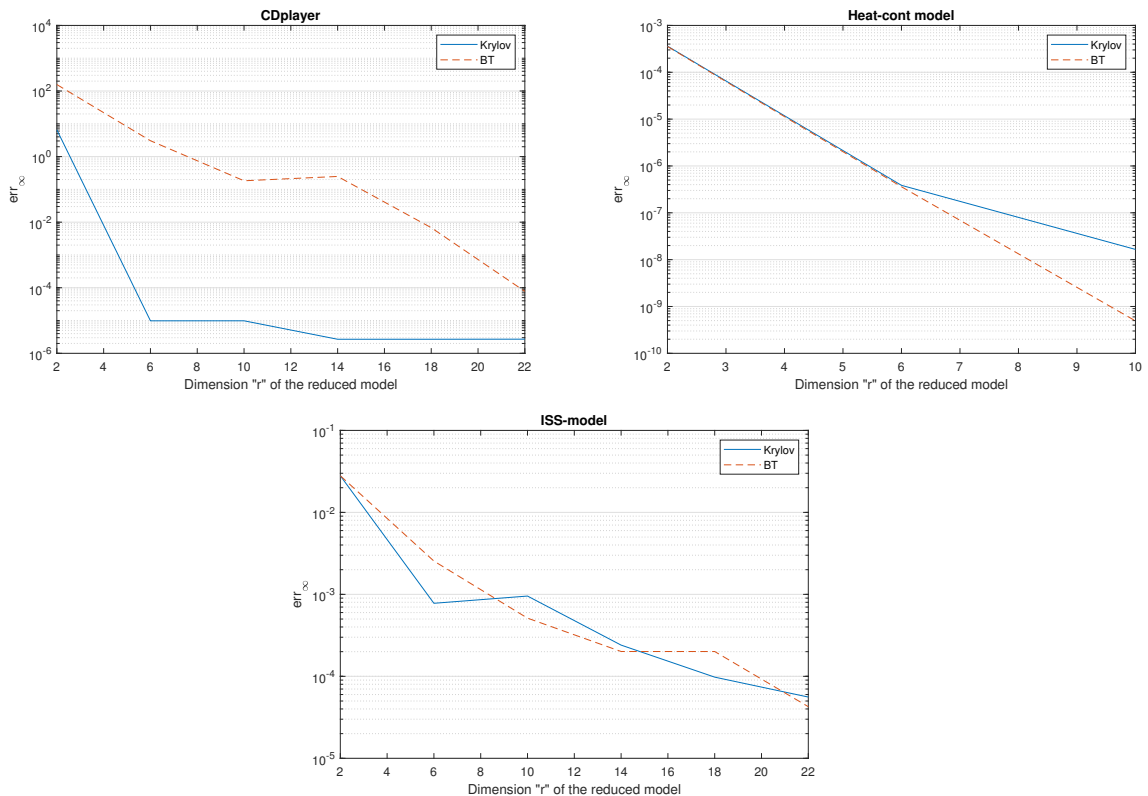


FIG. 4.1. The error between the original data and the constructed model for different values of  $r$  using our method described in Algorithm 3 (solid line) and Algorithm QuadBT (dashed line) for three benchmark models.

**4.3. SISO-model with large amount of data.** This subsection reports on the obtained numerical results using a high order SISO systems (*i.e.*,  $N > 5000$  and  $p = q = 1$ ). We get the samples  $\{H_j \in \mathbb{C}\}_{j=1}^N$  from a dynamic system generated with  $n = 50$  random complex poles and residues as it is mentioned in the beginning of Section 4. A comparison is also established between our method and the one described in [18], where the two Sylvester equations (2.13) are solved using ADI-Galerkin method from [6].

For both methods, we select  $N$  logarithmically spaced points  $\omega \in [1, 10^5]$  where  $N = \{5000, 10000\}$ . Unfortunately, we cannot know in advance which partition (*i.e.*, **H&H** or **O&E**) leads to a fast convergence and good approximations to the solutions of the two Sylvester equations (2.13), which are considered the key to build the reduced model. We have observed that using the **O&E** partition for our method and using the **H&H** partition for the one based on ADI-Galerkin process, leads to a good approximations with minimal computational coast and also to efficient reduced models which is the main purpose of this paper. Before reviewing the details of our results, we should mention that we have already tried to switch between the partitions to solve the two Sylvester equations (2.13), using both methods: Algorithm 2 and the ADI-Galerkin method, and as a consequence, we have obtained better results than the ones depicted in Table 4.1. However, the computed reduced models are quite far from the desired ones compared to the original data. The cpu-time mentioned in Table 4.1 refers to the time needed to compute the approximate solutions for both methods. In Table 4.2, we mentioned the total time that includes the construction of data, the approximation in a factored form of  $\mathbb{L}$  and  $\mathbb{L}_s$  as well as the projection and plotting steps. When solving the Sylvester equation `equa1` (2.13), we stopped the iterations when the residual norm  $\|R_m\|$  is less than a fixed tolerance  $\epsilon = 1e^{-5}$ . As mentioned before, we computed  $\|R_m\|$  in an appropriate way as described in (3.12), and when it comes to the

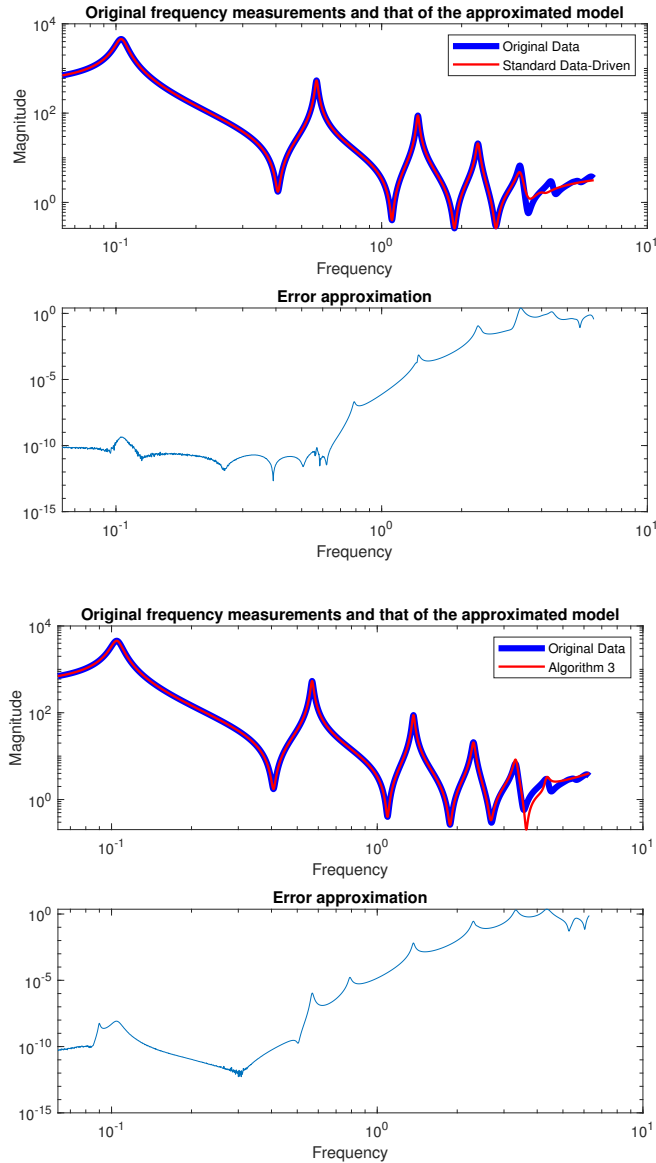


FIG. 4.2. Results for Beam-model

ADI-Galerkin method we follow what it has been suggested in [6, 18] where the  $\|R_m\|$  could be calculated by using method of *power iteration*. For `equa2` from (2.13), we stopped the iterations when the residual norm is less than  $\epsilon' = 1e^{-2}$ . The reason behind this choice of  $\epsilon'$  is that for  $\epsilon' = 1e^{-5}$  or less, both methods (i.e., ADI-Galerkin and Algorithm 2) need more iterations and will cost a lot of time to satisfy the convergence criterion, moreover, this choice was sufficient to obtain a satisfied reduced models as shown in figures below. Notice that the Loewner framework via ADI-Galerkin method in Figure 4.1 where  $N = 5000$  starts out very well, but apparently, after  $\phi = 10^5$ , the method lost its effectiveness and cannot match the picks. However, from Figure 4.4, our proposed method (Algorithm 3), gives better results. The same remark goes also for  $N = 10000$  as shown in Figures 4.6-4.7.



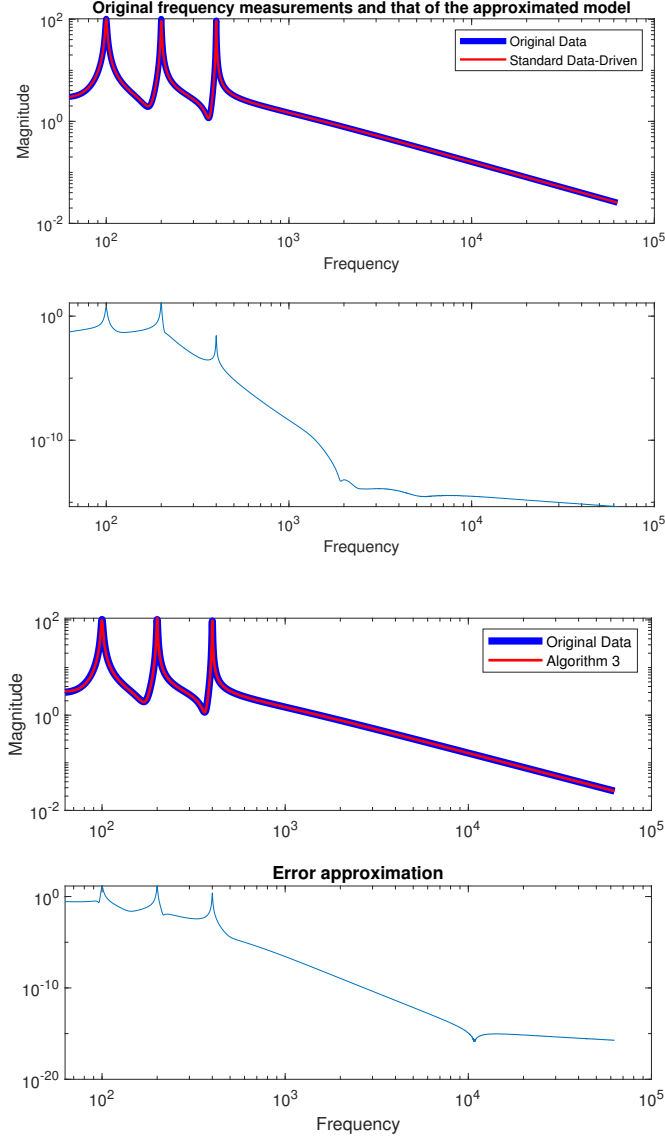


FIG. 4.3. Results for FOM-model.

**4.4. Large scale MIMO systems.** The examples treated here are devoted to high order MIMO systems (*i.e.*,  $N > 5000$  and  $p = q = 10$ ). We used (4.1) to get the samples of transfer matrices  $H_j \in \mathbb{C}^{p \times q}$  and chose  $N \in \{5000, \dots, 30000\}$  logarithmically spaced points  $\{\phi_j\}_{j=1}^N$  scaled by  $2\pi$  in the interval  $[10^4, 10^7]$ . We compare again our method with the one described in [18] where the two Sylvester equations (2.10) that characterize the right-hand sides of the Sylvester equations (2.13) must be chosen [21], and in order to conduct a fair comparison, we must select them in a similar way for both methods. Heuristically, these directions could be chosen as columns-rows of the identity matrix perturbed with some random columns-rows. In fact, there is no theory behind this specific choice, but normally this is what we have observed from the numerical experiments. We follow the same instructions already mentioned in the previous subsection to solve the Sylvester equations (2.13)

TABLE 4.1  
The obtained results of both methods in the case of SISO systems

Algorithm 2				
DATA	Sylv. equations (2.13)	cpu-time	resi. norm (3.12)	iter ("m")
$N = 5000$	equa1	1.21sec	$4.73 \times 10^{-06}$	40
	equa2	6.80sec	0.0050	80
$N = 10000$	equa1	1.89sec	$9.13 \times 10^{-06}$	39
	equa2	1.93sec	0.0895	40
ADI-Galerkin method				
DATA	Sylv. equations (2.13)	cpu-time	resi. norm	iter ("m")
$N = 5000$	equa1	3.87sec	$6.07 \times 10^{-06}$	25
	equa2	4.40sec	0.0234	50
$N = 10000$	equa1	25.26 sec	$8.39 \times 10^{-06}$	35
	equa2	27.22sec	0.0229	60

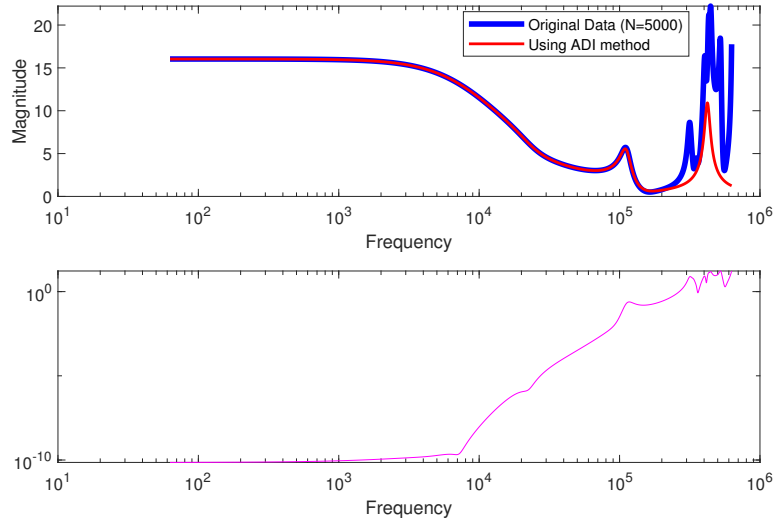


FIG. 4.4. Top : Original frequency measurements ( $\sigma_{max}(H_k)$ ) and the approximated model ( $\sigma_{max}(\mathcal{H}_r(\omega_k))$ ) by ADI-Galerkin method. Bottom : The error-norm versus frequencies.

TABLE 4.2  
Total computational time to obtain a reduced model of dimension  $r$ .

$N = 5000$	Algorithm 3	Data-driven via ADI-Galerkin
Total time (sec)	3.37	13.91
Dim. $r$ of the reduced model	$r = 45$	$r = 60$
$N = 10000$		
Total time (sec)	19.30	69.39
Dim. $r$ of the reduced model	$r = 57$	$r = 70$

using Algorithm 2 and the ADI-Galerkin method as done in [18]. In Table 4.3 we reported the cpu-time, the obtained error-norm and the total number of iterations.

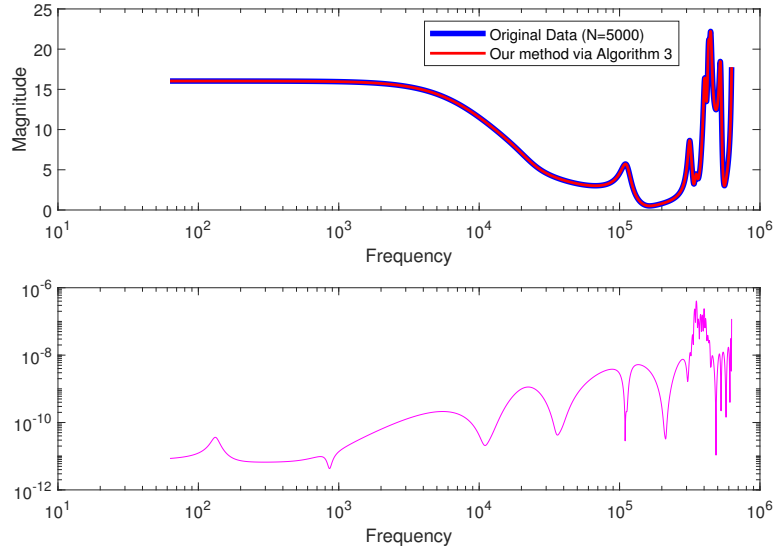


FIG. 4.5. *Top* : Original frequency measurements ( $\sigma_{max}(H_k)$ ) and the approximated model ( $\sigma_{max}(\mathcal{H}_r(\omega_k))$ ) using Algorithm 3. *Bottom* : The error-norm versus frequencies.

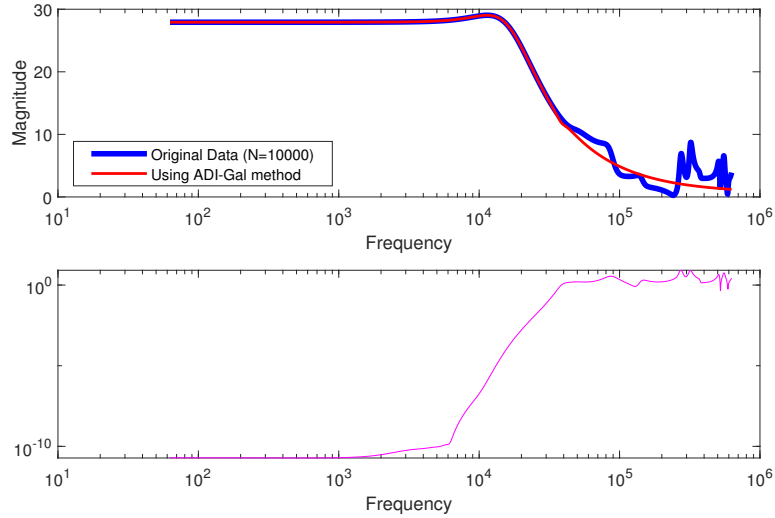


FIG. 4.6. *Top* : Original frequency measurements ( $\sigma_{max}(H_k)$ ) and the approximated model ( $\sigma_{max}(\mathcal{H}_r(\omega_k))$ ) by ADI-Galerkin method. *Bottom* : The error-norm versus frequencies.

To visualize the behaviour of the reduced system and to see how it represents the data, we plot in Figure 4.8 the magnitude of the original transfer function matrices (*i.e.*,  $\sigma_{max}(H_k)$ ) with  $\sigma_{max}(\mathcal{H}_r(\omega_k))$  given by the method described in [18] and based on the ADI-Galerkin approach. Figure 4.9 is devoted to the same experiment using our method from Algorithm 3.

The key tool to construct models in the standard data-driven via the Loewner framework is to compute an SVD of the Loewner matrix  $\mathbb{L} \in \mathbb{C}^{N \times N}$  [22], where we can notice from its definition in (2.12) that it is a dense matrix. As consequence and if we deal with a large amount of data  $N$ , the computation of the SVD should require a prohibitive computational cost. In Figure 4.10, we reported the results

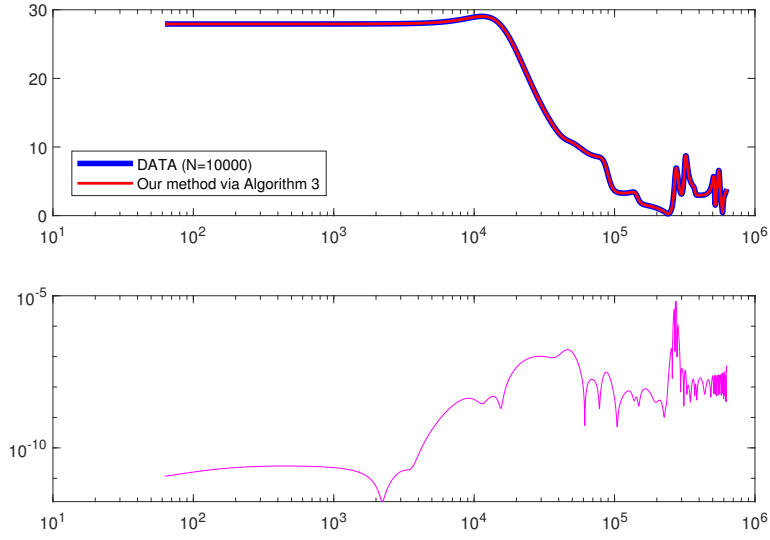


FIG. 4.7. Top : Original frequency measurements ( $\sigma_{max}(H_k)$ ) and the approximated model ( $\sigma_{max}(\mathcal{H}_r(\omega_k))$ ) using Algorithm 3. Bottom : The error-norm versus frequencies.

TABLE 4.3  
The obtained results of both methods in the case of MIMO systems

Extended Krylov-based method				
DATA	Sylv. equations (2.13)	cpu-time	resi. norm (3.12)	iter ( $m$ )
$N = 10\,000$	equa1	1.97sec	$5.90 \times 10^{-07}$	7
	equa2	6.49sec	0.0090	13
$N = 20\,000$	equa1	3.96sec	$1.85 \times 10^{-07}$	7
	equa2	4.08sec	0.0086	13
$N = 30\,000$	equa1	13.64sec	$8.87 \times 10^{-08}$	7
	equa2	19.4sec	0.0058	10
ADI-Galerkin method				
DATA	Sylv. equations (2.13)	cpu-time	resi. norm	iter (" $m$ ")
$N = 10\,000$	equa1	32.47sec	$2.40 \times 10^{-09}$	6
	equa2	39.16sec	0.0017	6
$N = 20\,000$	equa1	312.14sec	$6.72 \times 10^{-09}$	6
	equa2	281.23sec	0.0053	6
$N = 30\,000$	equa1, equa2	>400sec	-	-

regarding the computational cpu time to get this decomposition for different values of  $N$ . We showed the time needed to achieve this decomposition using the idea of the proposed method in Algorithm 3 compared to a direct computation using the built-in MATLAB function `svd` as described in [22]. We can notice from the graph the exponential growth of the cpu-time required to get the SVD as  $N$  increases, which makes this option quite impractical. Here, we need to mention that  $m$  is fixed between 20 and 160, it depends actually on the amount of the data  $N$ , which means that when  $N$  increases,  $m$  also increases. Recall that  $2m(p+q)$  is the dimension of the extended Krylov subspace used to solve the Sylvester equations (2.13).

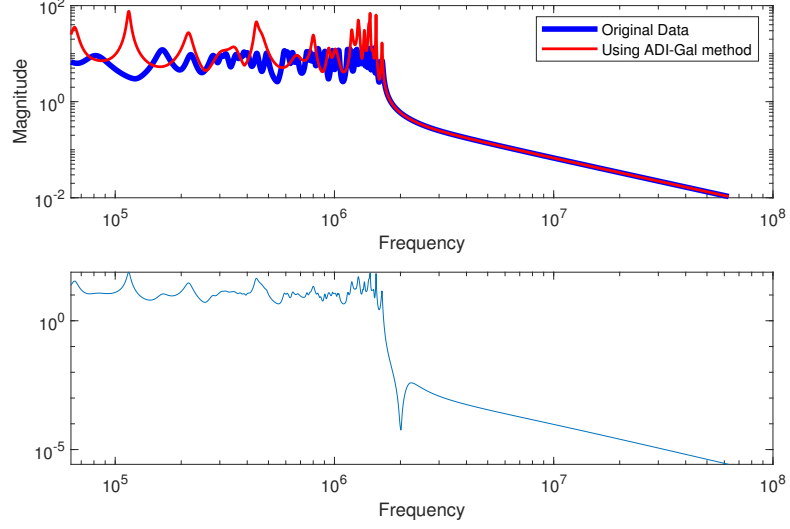


FIG. 4.8. *Top* : Original responses ( $\sigma_{max}(H_k)$ ) vs the responses of the reduced system ( $\sigma_{max}(\mathcal{H}_r(\omega_k))$ ) constructed by ADI-Galerkin method. *Bottom* : The error-norm versus frequencies.

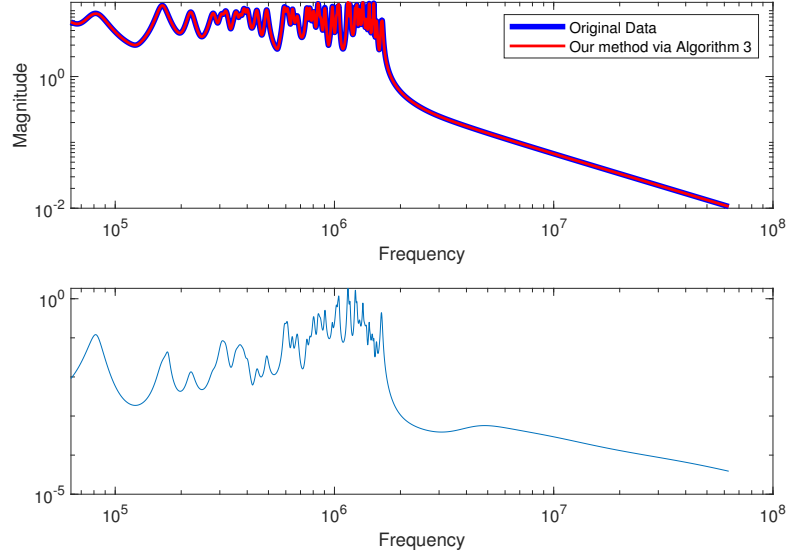


FIG. 4.9. *Top* : Original responses ( $\sigma_{max}(H_k)$ ) vs the responses of the reduced system ( $\sigma_{max}(\mathcal{H}_r(\omega_k))$ ) constructed by Algorithm 3. *Bottom* : The error-norm versus frequencies.

**Conclusion.** In some physics or engineering problems, we deal with complex systems with high non-linear terms or in some cases we have no access to the system matrices as it is the case for the problematic brain modelling. All these issues lead to the introduction of data based methods that are not taking into account governing physics or system's structure to obtain inferred dynamical systems simple to manipulate or control. To this end, Loewner approach is among the many widely used approaches to derive dynamical systems from data. The process of this approach starts by setting up the data and constructing the Loewner and shifted Loewner matrices ( $\mathbb{L}$  and  $\mathbb{L}_s$ ), which

TABLE 4.4  
 Total computational time to obtain a reduced model of dimension  $r$

$N = 10\,000$	Algorithm 3	Data-driven via ADI-Galerkin
Total time (sec)	39.75	102.74
Dim. $r$ of the reduced model	$r = 160$	$r = 120$
<hr/>		
$N = 20\,000$		
Total time (sec)	199.40	622.15
Dim. $r$ of the reduced model	$r = 160$	$r = 120$

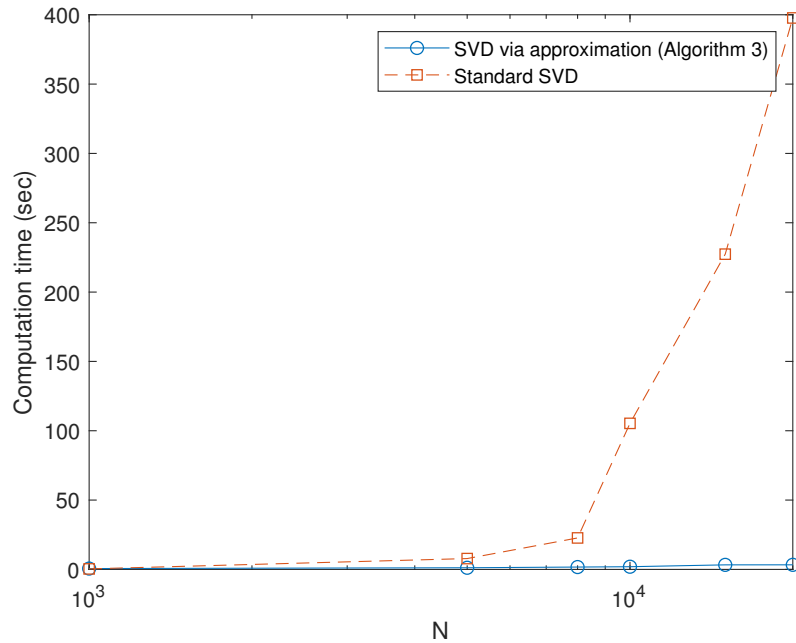


FIG. 4.10. The computation time needed to compute the SVD of  $\mathbb{L} \in \mathbb{C}^{N \times N}$  for different choice of  $N$  and for  $p = q = 10$ .

are considered the key tool of this approach. The main problem we have addressed in this paper is to construct reduced systems from a large amount of data. We have exploited the fact that  $\mathbb{L}$  and  $\mathbb{L}_s$  both satisfy certain Sylvester matrix equations. Then, an extended block Krylov-subspace method is proposed to compute efficient approximate solutions in a factored form. Two benefits have been obtained by following this technique such as storage and computations that have been reduced by means of the use of approximate factored solutions. This also guaranteed a proper SVD computation of  $\mathbb{L}$  as described above. Finally, we evaluated our proposed method by providing some examples of SISO and MIMO systems with comparisons to other known methods.

**Acknowledgements.** The authors would like to thank the two referees for carefully checking the paper and providing a number of helpful remarks and suggestions.

#### REFERENCES

- [1] I. ABDAOUI, L. ELBOUYAHYAOU, AND M.HEYOUNI, An alternative extended block arnoldi method for solving low-rank sylvester equations, *Comput. Math. with Appl.*, 78(8) (2019), pp. 2817–2830.
- [2] A. ANTOULAS, Approximation of large-scale dynamical systems, SIAM Philadelphia, (2005).

- [3] A. ANTOULAS, C. BEATTIE, AND S. GUGERCINE, Interpolatory methods for model reduction, *Comput. Science and Engineering* 21. SIAM Philadelphia, (2020).
- [4] A. ANTOULAS, S. LEFTERIU, AND A. IONITA, A tutorial introduction to the loewner framework for model reduction, In : *Model Reduction and Approximation*, chap. 8 (2017), pp. 0898–1221.
- [5] H. BARKOUKI, A. H. BENTBIB, AND K. JBILOU, An extended nonsymmetric block lanczos method for model reduction in large scale dynamical systems, *Calcolo*, 55(1) (2018), pp. 1–23.
- [6] P. BENNER AND P. KÜRSCHNER, Computing real low-rank solutions of sylvester equations by the factored adi method, *Comput. Math. with Appl.*, 67(9) (2009), pp. 0898–1221.
- [7] P. BENNER, M. OHLBERGER, A. COHEN, AND K. WILLCOX, Model reduction and approximation, SIAM Philadelphia, (2017).
- [8] K. CHERIFI, P. GOYAL, AND P. BENNER, A greedy data collection scheme for linear dynamical systems, <https://arxiv.org/abs/2107.12950v1>, (2021).
- [9] M. FRANGOS AND I. M. JAIMOUKHA, Adaptive rational krylov algorithms for model reduction, In *Proceedings of the European Control Conference*, (2007), pp. 4179–4186.
- [10] G. GOLUB, S. NASH, AND C. LOAN, A hessenberg-schur method for the problem  $ax + xb = c$ , *IEEE Trans. Automat. Control*, 24(6) (1979), p. 909–913.
- [11] I. GOSEA, S. GUGERCIN, AND C. BEATTIE, Data-driven balancing of linear dynamical systems, <https://arxiv.org/abs/2104.01006v1>, (2021).
- [12] I. GOSEA, C. VASSAL, AND A. ANTOULAS, Data-driven modeling and control of large-scale dynamical systems in the loewner framework, <https://arxiv.org/abs/2108.11870>, (2021).
- [13] I. V. GOSEA, Q. ZHANG, AND A. ANTOULAS, Data-driven modeling from noisy measurements, *Proceedings in Appl. Math. and Mechanics*, 67(9) (2021), pp. 0898–1221.
- [14] E. GRIMME, Krylov projection methods for model reduction, Ph.D. thesis, Coordinated Science Laboratory, University of Illinois at Urbana Champaign, (1997).
- [15] S. GUGERCIN AND A. C. ANTOULAS, A survey of model reduction by balanced truncation and some new results, *Internat. J. Control*, 77(8) (2003), pp. 748–766.
- [16] M. A. HAMADI, K. JBILOU, AND A. RATNANI, Model reduction method in large scale dynamical systems using an extended-rational block arnoldi method, *J. Appl. Math. Comput.*, (2021).
- [17] M. HEYOUNI AND K. JBILOU, An extended block arnoldi method for large matrix riccati equations, *Elect. Trans. Numer. Anal.*, 33 (2009), pp. 53–62.
- [18] A. HOCHMAN, Fast singular-value decomposition of loewner matrices for state-space macromodeling, *IEEE 24th Electrical Performance of Electronic Packaging and Systems (EPEPS)*, (2015), pp. 177–180.
- [19] K. JBILOU, Low rank approximate solutions to large sylvester matrix equations, *Appl. Math. and Comput.*, 177(1) (2006), pp. 365–376.
- [20] L. KNIZHNERMAN, D. DRUSKIN, AND M. ZASLAVSKY, On optimal convergence rate of the rational krylov subspace reduction for electromagnetic problems in unbounded domains, *SIAM J. Numer. Anal.*, 47(2) (2009), pp. 953–971.
- [21] S. LEFTERIU AND A. ANTOULAS, A new approach to modeling multiport systems from frequency-domain data, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29 (2010), pp. 14–27.
- [22] A. MAYO AND A. ANTOULAS, A framework for the solution of the generalized realization problem, *Linear Algebra and Its Applications*, 425(2-3) (2007), p. 634–662.
- [23] V. MEHRMANN AND T. STYKEL, Balanced truncation model reduction for large-scale systems in descriptor form, *Lecture Notes in Computational Science and Engineering*, 45 (2005), pp. 83–115.
- [24] B. C. MOORE, Principal component analysis in linear systems: controllability, observability and model reduction, *IEEE Trans. Auto. Cont.*, AC-26 (1981), pp. 17–32.
- [25] D. PALITTA AND S. LEFTERIU, An efficient, memory-saving approach for the loewner framework, <https://arxiv.org/abs/2103.07146>, (2021).
- [26] B. PEHERSTORFER, S. GUGERCIN, AND K. WILLCOX, Data-driven reduced model construction with time-domain loewner models, *SIAM J. Sc. Comput.*, 39(5) (2017), pp. A2152–78.
- [27] V. SIMONCINI, A new iterative method for solving large-scale lyapunov matrix equations, *SIAM J. Sci. Comput.*, 29(3) (2007), pp. 1268–1288.
- [28] V. SIMONCINI, D. B. SZYLD, AND M. MARLLINY, On two numerical methods for the solution of large-scale algebraic riccati equations, *IMA Journal of Numerical Analysis*, 34 (2014), pp. 904–920.