SIAM J. SCI. COMPUT.
Vol. 46, No. 1, pp. A483–A507



# DATA-DRIVEN AND LOW-RANK IMPLEMENTATIONS OF BALANCED SINGULAR PERTURBATION APPROXIMATION*

BJÖRN LILJEGREN-SAILER† AND ION VICTOR GOSEA‡

**Abstract.** Balanced singular perturbation approximation (SPA) is a model order reduction method for linear time-invariant systems that guarantees asymptotic stability and for which there exists an a priori error bound. In that respect, it is similar to balanced truncation (BT). However, the reduced models obtained by SPA generally introduce better approximation in the lower frequency range and near steady-states, whereas BT is better suited for the higher frequency range. Even so, independently of the frequency range of interest, BT and its variants are more often applied in practice, since there exist more efficient algorithmic realizations thereof. In this paper, we aim at closing this practically relevant gap for SPA. We propose two novel and efficient algorithms that are adapted for different settings. First, we derive a low-rank implementation of SPA that is applicable in the large-scale setting. Second, a data-driven reinterpretation of the method is proposed that only requires input-output data and thus is realization-free. A main tool for our derivations is the reciprocal transformation, which induces a distinct view on implementing the method. While the reciprocal transformation and the characterization of SPA are not new, their significance for the practical algorithmic realization has been overlooked in the literature. Our proposed algorithms have well-established counterparts for BT and, as such, a comparable computational complexity. The numerical performance of the two novel implementations is tested for several numerical benchmarks, and comparisons to their counterparts for BT as well as to existing implementations of SPA are made.

**Key words.** balanced singular perturbation approximation, data-driven modeling, balanced truncation, numerical quadrature, non-intrusive methods, low-rank implementation, linear systems, realization-free

**MSC codes.** 93A15, 93B15, 93B40, 37M99, 65D30

**DOI.** 10.1137/23M155791X


See reproducibility of computational results at end of the article.

## 1. Introduction.

For many modern applications in the applied sciences, the simulation, control, and optimization of very large-scale dynamical systems is required. The increased development of modern computing environments and high-performance computing tools has pushed the boundaries of what is computationally feasible in this regard. Another way to navigate this issue and to save computational time is by means of approximating such large-scale dynamical systems with reduced models. This is the essence of model reduction, which is a subfield at the intersection of many established fields, such as automatic control, systems theory, approximation theory, and numerical linear algebra; see, e.g., the books [1, 36, 11, 6, 2].

For the reduction of linear time-invariant systems, system-theoretic methods are widely used. Moment matching methods and projection-based balancing-related methods, such as balanced truncation (BT), are among the most popular methods falling in this category; see [17, 4] and references therein. Moment matching methods are particularly efficient in terms of computational complexity. The resulting reduced

*Submitted to the journal's Numerical Algorithms for Scientific Computing section March 9, 2023; accepted for publication (in revised form) October 9, 2023; published electronically February 7, 2024.
https://doi.org/10.1137/23M155791X

†Corresponding author. Johann Radon Institute Linz, 4040 Linz, Austria (bsailer@ricam.oeaw.ac.at).

‡Max Planck Institute for Dynamics of Complex Technical Systems, D-39106 Magdeburg, Germany (gosea@mpi-magdeburg.mpg.de).

models are also typically of high fidelity, but in general it is difficult to enforce theoretical guarantees.

The balancing-related methods preserve important qualitative features of a system, such as stability, and there exists a priori error bounds for those methods [21]. The concept behind BT, originally proposed in [32, 31], is to identify and truncate components of the original system that are weakly controllable and observable. This turns out to be a natural approach, assuming the truncated states have a fast dynamic. Another closely related method is (balanced) singular perturbation approximation (SPA) [20, 29]. It is a nonprojection-based balancing-related method, for which the same a priori guarantees as for BT are valid. Other than that, SPA is better suited for the lower frequency range and near steady-states.

Notably, the numerical algorithms available for BT are more developed than those for SPA. For example, low-rank implementations are exclusive to BT, and until now SPA could not be applied in the truly large-scale setting; cf. section 4 for references and more details. The recent contribution [22] connects classical projection-based BT approach to the data-driven interpolation-based approach and provides a nonintrusive implementation of the method. With the ever-increasing availability of data, i.e., measurements related to the original model, the interest in nonintrusive methods has grown significantly. Among other methods, we would like to mention the Loewner framework [30], dynamic mode decomposition [40], and operator inference [34, 5].

The main objective of this paper is twofold: first, we aim at deriving an efficient algorithm for SPA in a large-scale setting and, second, at providing a realization-free implementation of SPA. The latter will be performed in a nonintrusive manner, using solely transfer function evaluations and corresponding quadrature weights. Both new implementations have strong resemblances to existing implementations of BT. The essential link between BT and SPA is given by the so-called reciprocal transformation, and this plays a crucial role in our derivations.

The remainder of the paper is organized as follows. Section 2 provides an overview on balancing and the balancing-related model reduction methods BT and SPA and points toward the open issues for the implementation of SPA.

Then, in section 3 the reciprocal transformation is defined and analyzed in a slightly generalized setting as compared to the literature. In section 4 we recall the square-root and low-rank implementations of BT and then present a novel counterpart for SPA. Numerical results for the novel low-rank algorithm are provided. Next, section 5 introduces in detail the data-driven interpretation of BTand, by that, sets the stage for the newly proposed data-driven implementation of SPA. Moreover, connections to the Loewner framework are discussed. An extensive numerical study for the data-driven methods is provided in section 6 for two benchmark examples. Finally, in section 7 the conclusions of the paper are outlined, together with a brief outlook to future research endeavors.

**2. Balancing and balancing-related model reduction.** Consider the linear time-invariant (LTI) dynamical system

$$\begin{aligned} \mathbf{E}\,\dot{\mathbf{x}}(t) &= \mathbf{A}\,\mathbf{x}(t) + \mathbf{B}\,\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}\,\mathbf{x}(t) + \mathbf{D}\,\mathbf{u}(t), \end{aligned} \tag{2.1}$$

where the input mapping is given by $\mathbf{u} : [0, \infty] \to \mathbb{R}^m$, the (generalized) state variable is $\mathbf{x} : [0, \infty] \to \mathbb{R}^n$, and the output mapping is $\mathbf{y} : [0, \infty] \to \mathbb{R}^p$. The system matrices are given by $\mathbf{E}, \mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{p\times n}$, and $\mathbf{D} \in \mathbb{R}^{p\times m}$. The system is closed by choosing initial conditions $\mathbf{x}(0) = \mathbf{x}^0 \in \mathbb{R}^n$. We assume the system to be asymptotically stable, i.e., $\mathbf{E}$ and $\mathbf{A}$ are nonsingular, and the eigenvalues of $\mathbf{E}^{-1}\mathbf{A}$ are

located in the open left half-plane. In general, the state dimension $n$ may be large, and (2.1) will also be referred to as the full order model (FOM).

The input-output behavior of the system, for $\mathbf{x}(0) = \mathbf{0}$ and in the frequency domain, is fully characterized by the transfer function, given by

$$\mathbf{H}(s) = \mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}, \qquad \mathbf{H}(s) \in \mathbb{C}^{p\times q}, \tag{2.2}$$

for any scalar (frequency) $s \in \mathbb{C}$.

Model reduction aims at computing a reduced order model (ROM), i.e., an LTI system of a reduced state dimension $r \ll n$, that is supposed to reproduce a similar output $\mathbf{y}_r \approx \mathbf{y}$ for the same inputs. This ROM is given by

$$\begin{aligned} \dot{\mathbf{x}}_r(t) &= \mathbf{A}_r\,\mathbf{x}_r(t) + \mathbf{B}_r\,\mathbf{u}(t), \\ \mathbf{y}_r(t) &= \mathbf{C}_r\,\mathbf{x}_r(t) + \mathbf{D}_r\,\mathbf{u}(t), \end{aligned} \tag{2.3}$$

with initial conditions $\mathbf{x}_r(0) = \mathbf{x}_r^0 \in \mathbb{R}^r$ and reduced dimension matrices $\mathbf{A}_r \in \mathbb{R}^{r\times r}$, $\mathbf{B}_r \in \mathbb{R}^{r\times m}$, $\mathbf{C}_r \in \mathbb{R}^{p\times r}$, and $\mathbf{D}_r \in \mathbb{R}^{p\times m}$. Arguably, the projection-based approach is most commonly used [1, 8], in which reduction bases $\mathbf{W}_r, \mathbf{V}_r \in \mathbb{R}^{n\times r}$ with $\mathbf{W}_r^T\mathbf{E}\mathbf{V}_r = \mathbf{I}_r \in \mathbb{R}^{r\times r}$ (unit matrix) are used to construct the reduced model. In that case, $\mathbf{A}_r = \mathbf{W}_r^T\mathbf{A}\mathbf{V}_r$, $\mathbf{B}_r = \mathbf{W}_r^T\mathbf{A}$, $\mathbf{C}_r = \mathbf{C}\mathbf{V}_r$, and $\mathbf{D} = \mathbf{D}_r$.

**2.1. Balanced realization.** The two fundamental quantities in BT and SPA are the controllability and observability Gramians $\mathbf{P}$ and $\mathbf{Q}$, given as the solutions of the two Lyapunov equations

$$\mathbf{A}\mathbf{P}\mathbf{E}^T + \mathbf{E}\mathbf{P}\mathbf{A}^T + \mathbf{B}\mathbf{B}^T = \mathbf{0}, \quad \mathbf{A}^T\mathbf{Q}\mathbf{E} + \mathbf{E}^T\mathbf{Q}\mathbf{A} + \mathbf{C}^T\mathbf{C} = \mathbf{0}. \tag{2.4}$$

The Gramian $\mathbf{P}$ induces a measure for controllability of a state, and likewise, $\mathbf{Q}$ induces a measure for observability. However, these measures are not invariant under state transformations. Let matrix $\mathbf{F} \in \mathbb{R}^{n\times n}$ such that $\mathbf{F}\mathbf{E}\mathbf{F}^{-1} = \mathbf{I}$, and define the transformed matrices $\hat{\mathbf{A}} = \mathbf{F}\mathbf{A}\mathbf{F}^{-1}$, $\hat{\mathbf{B}} = \mathbf{F}\mathbf{B}$, and $\hat{\mathbf{C}} = \mathbf{C}\mathbf{F}^{-1}$. Then the input-output behavior of (2.1) is equivalently described by the alternative LTI realization

$$\dot{\hat{\mathbf{x}}}(t) = \hat{\mathbf{A}}\,\hat{\mathbf{x}}(t) + \hat{\mathbf{B}}\,\mathbf{u}(t), \quad \mathbf{y}(t) = \hat{\mathbf{C}}\,\hat{\mathbf{x}}(t) + \mathbf{D}\,\mathbf{u}(t),$$

and $\hat{\mathbf{x}}(0) = \mathbf{F}\mathbf{x}_0$. A realization is called *balanced* if the Gramians are diagonal and equal to each other. More specifically, the transformed Gramians $\hat{\mathbf{P}}$ and $\hat{\mathbf{Q}}$ need to satisfy the relation

$$\hat{\mathbf{P}} = \mathbf{F}\mathbf{P}\mathbf{F}^T \stackrel{!}{=} \hat{\mathbf{Q}} = \mathbf{F}^{-T}\mathbf{Q}\mathbf{F}^{-1} \stackrel{!}{=} \operatorname{diag}(\sigma_1, \ldots, \sigma_n) \in \mathbb{R}^{n\times n},$$

with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$ sorted in descending order. The $\sigma_i$'s are invariants of the LTI system (they are the same for any realization of the state-space) and referred to as to the *Hankel singular values.*

**2.2. Balancing-related model reduction.** Balancing-related model reduction methods construct a reduced model based on the most observable and most controllable states, which relate to the dominant Hankel singular values. Let $1 < r < n$, $\sigma_r > \sigma_{r+1}$, and the balanced state $\hat{\mathbf{x}} = \mathbf{F}\mathbf{x}$ be partitioned as $\hat{\mathbf{x}} = [\hat{\mathbf{x}}_1^T, \hat{\mathbf{x}}_2^T]^T$, with $\hat{\mathbf{x}}_1 : [0,\infty] \to \mathbb{R}^r$ and the remainder $\hat{\mathbf{x}}_2 : [0,\infty] \to \mathbb{R}^{n-r}$. The transformed system matrices are partitioned accordingly as

$$\hat{\mathbf{A}} = \begin{bmatrix} \hat{\mathbf{A}}_{11} & \hat{\mathbf{A}}_{12} \\ \hat{\mathbf{A}}_{21} & \hat{\mathbf{A}}_{22} \end{bmatrix}, \qquad \hat{\mathbf{B}} = \begin{bmatrix} \hat{\mathbf{B}}_1 \\ \hat{\mathbf{B}}_2 \end{bmatrix}, \qquad \hat{\mathbf{C}} = \begin{bmatrix} \hat{\mathbf{C}}_1 & \hat{\mathbf{C}}_2 \end{bmatrix},$$

and thus the following relations hold:

$$\dot{\hat{\mathbf{x}}}_1(t) = \hat{\mathbf{A}}_{11}\hat{\mathbf{x}}_1(t) + \hat{\mathbf{A}}_{12}\hat{\mathbf{x}}_2(t) + \hat{\mathbf{B}}_1\mathbf{u}(t), \tag{2.5a}$$

$$\dot{\hat{\mathbf{x}}}_2(t) = \hat{\mathbf{A}}_{21}\hat{\mathbf{x}}_1(t) + \hat{\mathbf{A}}_{22}\hat{\mathbf{x}}_2(t) + \hat{\mathbf{B}}_2\mathbf{u}(t), \tag{2.5b}$$

$$\mathbf{y}(t) = \hat{\mathbf{C}}_1\hat{\mathbf{x}}_1(t) + \hat{\mathbf{C}}_2\hat{\mathbf{x}}_2(t) + \mathbf{D}\mathbf{u}(t). \tag{2.5c}$$

For both BT and SPA, reduced models are constructed by a simplification of (2.5a) which can be expressed in terms of the variable $\mathbf{x}_r \approx \hat{\mathbf{x}}_1$.

*Balanced truncation.* In BT, the reduced model is obtained by completely eliminating the second (remainder) component of the state, i.e., by enforcing the simplified dynamics $\hat{\mathbf{x}}_2 \approx \mathbf{0}$. The ensuing ROM is given as in (2.3), with the following choice of reduced matrices:

$$\mathbf{A}_r = \hat{\mathbf{A}}_{11}, \quad \mathbf{B}_r = \hat{\mathbf{B}}_1, \quad \mathbf{C}_r = \hat{\mathbf{C}}_1, \quad \mathbf{D}_r = \mathbf{D}.$$

This approximation assumes that the remainder component $\hat{\mathbf{x}}_2$ consists of fast dynamics that rapidly decay to zero.

*Singular perturbation approximation.* SPA pursues, in some sense, a converse path from that of BT. The remainder component of the state is approximated using the steady-state equation $\frac{d}{dt}\hat{\mathbf{x}}_2(t) \approx \mathbf{0}$ for $t \geq 0$. Thus, (2.5b) is modified to $\mathbf{0} \approx \hat{\mathbf{A}}_{21}\hat{\mathbf{x}}_1(t) + \hat{\mathbf{A}}_{22}\hat{\mathbf{x}}_2(t) + \hat{\mathbf{B}}_2\mathbf{u}(t)$, which implies the approximation $\hat{\mathbf{x}}_2(t) \approx -\hat{\mathbf{A}}_{22}^{-1}(\hat{\mathbf{A}}_{21}\hat{\mathbf{x}}_1(t) + \hat{\mathbf{B}}_2\mathbf{u}(t))$. A reduced formulation with $\mathbf{x}_r(t) \approx \hat{\mathbf{x}}_1(t)$ enforcing this approximated dynamics can be obtained by basic manipulations. This yields the matrices of the SPA reduced system, which read

$$\begin{aligned} \mathbf{A}_r &= \hat{\mathbf{A}}_{11} - \hat{\mathbf{A}}_{12}\hat{\mathbf{A}}_{22}^{-1}\hat{\mathbf{A}}_{21}, & \mathbf{B}_r &= \hat{\mathbf{B}}_1 - \hat{\mathbf{A}}_{12}\hat{\mathbf{A}}_{22}^{-1}\hat{\mathbf{B}}_2, \\ \mathbf{C}_r &= \hat{\mathbf{C}}_1 - \hat{\mathbf{C}}_2\hat{\mathbf{A}}_{22}^{-1}, & \mathbf{D}_r &= \mathbf{D} - \hat{\mathbf{C}}_2\hat{\mathbf{A}}_{22}^{-1}\hat{\mathbf{B}}_2. \end{aligned} \tag{2.6}$$

Note that this method leads, in general, to a feedthrough term $\mathbf{D}_r \neq \mathbf{0}$, even if $\mathbf{D}$ is equal to zero. Thus, the SPA method is *not a projection-based approach.*

**2.3. Open issues for the implementation of singular perturbation approximation.** The numerical implementation of balanced realizations turns out to be computationally demanding and not well-conditioned for large-scale systems. Therefore, balancing-free implementations have been proposed for various balancing-related model reduction methods. Efficient implementations for dense systems of medium size have been derived based on the sign function method and similar techniques for both BT (see [12, 10, 1]) and SPA [43, 13, 3]. Other significant results and application areas have exclusively been developed for BT and other projection-based balancing-related methods. It should be noted that the existing implementations of SPA explicitly use formula (2.6), which inherently prohibits the use of low-rank approximations. Furthermore, a data-driven (realization-free) implementation of SPA has not yet been proposed in the literature. In this paper, we aim at filling this gap for the case of SPA by deriving low-rank and realization-free algorithms.

**3. Reciprocal transformation.** It is well known that the SPA can be interpreted in terms of the so-called reciprocal transformation [29, 23]. While this interpretation has been used for theoretical investigations of SPA, its relevance for the algorithmic implementation was not broadly investigated until now. In this section, we propose a definition of the reciprocal transformation in a slightly modified setting, considering LTI systems with general regular matrices $\mathbf{E}$ as opposed to the special case $\mathbf{E} = \mathbf{I}$ used in the literature. We show that our extension inherits the fundamental properties that have been derived in [29, 23, 26] for this special case.

DEFINITION 3.1. *The reciprocal system, i.e., the reciprocal transformed counterpart of the* LTI *system in* (2.1)*, is defined as*

$$\begin{aligned}\mathscr{E}\,\dot{\mathsf{x}}(t) &= \mathscr{A}\,\mathsf{x}(t) + \mathscr{B}\,\mathsf{u}(t),\\ \mathsf{y}(t) &= \mathscr{C}\,\mathsf{x}(t) + \mathscr{D}\,\mathsf{u}(t),\end{aligned}$$

$$\begin{aligned}\textit{with}\quad \mathscr{A} &= \boldsymbol{E}\boldsymbol{A}^{-1}\boldsymbol{E}, & \mathscr{B} &= \boldsymbol{E}\boldsymbol{A}^{-1}\boldsymbol{B}, \quad \mathscr{E} = \boldsymbol{E},\\ \mathscr{C} &= -\boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{E}, & \mathscr{D} &= \boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B}.\end{aligned}$$

*The reciprocal transfer function is defined by* $\mathscr{H}(s) = \mathscr{C}(s\mathscr{E} - \mathscr{A})^{-1}\mathscr{B} + \mathscr{D}$ *for* $s \in \mathbb{C}$.

From a theoretical point of view, one could directly invert the matrix $\mathbf{E}$ to end up in the standard setting. However, this step needs to be avoided for an efficient implementation of SPA in the general large-scale setting; cf. subsection 4.2.

PROPOSITION 3.2. *The reciprocal transformed system of an asymptotically stable* LTI *system as in* (2.1) *is well-defined and asymptotically stable. Moreover, by applying twice the reciprocal transformation to an* LTI *system, the same* LTI *system is obtained.*

The latter result follows by straightforward calculus. Next, the relation in the frequency domain between the two transfer functions is provided; this will clarify the origin of the term "reciprocal."

THEOREM 3.3. *The transfer function* $\mathbf{H}$ *of the* LTI *system in* (2.1) *and the reciprocal transfer function* $\mathscr{H}$ *are reciprocal to each other, in the sense that for any* $s \in \mathbb{C}^+$*, it holds that*

$$\mathscr{H}(s) = \mathbf{H}\left(\frac{1}{s}\right) \quad \textit{and, respectively,} \quad \mathscr{H}\left(\frac{1}{s}\right) = \mathbf{H}(s).$$

*Proof.* We first prove that $(1/s\mathbf{E} - \mathbf{A})^{-1} = (-\mathscr{E}^{-1}\mathscr{A})[\mathscr{E}^{-1} + (s\mathscr{E} - \mathscr{A})^{-1}\mathscr{A}\mathscr{E}^{-1}]$ as a preliminary step. The latter holds due to

$$\begin{aligned}&(1/s\mathbf{E} - \mathbf{A})(-\mathscr{E}^{-1}\mathscr{A})\left[\mathscr{E}^{-1} + (s\mathscr{E} - \mathscr{A})^{-1}\mathscr{A}\mathscr{E}^{-1}\right]\\ &\quad = 1/s(s\mathscr{E} - \mathscr{A})\left[\mathscr{E}^{-1} + (s\mathscr{E} - \mathscr{A})^{-1}\mathscr{A}\mathscr{E}^{-1}\right] = 1/s[(s\mathscr{E} - \mathscr{A})\mathscr{E}^{-1} + \mathscr{A}\mathscr{E}^{-1}] = \mathbf{I}.\end{aligned}$$

With this equality, it follows that

$$\begin{aligned}\mathbf{H}(1/s) &= \mathbf{D} + \mathbf{C}(1/s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}\\ &= \mathbf{D} + \mathbf{C}(-\mathscr{E}^{-1}\mathscr{A})\left[\mathscr{E}^{-1} + (s\mathscr{E} - \mathscr{A})^{-1}\mathscr{A}\mathscr{E}^{-1}\right]\mathbf{B}\\ &= [\mathbf{D} - \mathbf{C}\mathscr{E}^{-1}\mathscr{A}\mathscr{E}^{-1}\mathbf{B}] - \mathbf{C}[\mathscr{E}^{-1}\mathscr{A}(s\mathscr{E} - \mathscr{A})^{-1}\mathscr{A}\mathscr{E}^{-1}]\mathbf{B}\\ &= \mathscr{D} + [-\mathbf{C}\mathscr{E}^{-1}\mathscr{A}](s\mathscr{E} - \mathscr{A})^{-1}[\mathscr{A}\mathscr{E}^{-1}\mathbf{B}] = \mathscr{D} + \mathscr{C}\left(s\mathscr{E} - \mathscr{A}\right)^{-1}\mathscr{B} = \mathscr{H}(s).\end{aligned}$$

The other relation follows by interchanging the roles of the LTI system and its reciprocal system; cf. Proposition 3.2. ∎

The main results of this paper are based on the following alternative characterization of SPA that is illustrated by means of Figure 1.

PROPOSITION 3.4. *For an asymptotically stable* LTI *system with Hankel singular values fulfilling* $\sigma_r > \sigma_{r+1}$*, the* ROM *of dimension* $r$ *obtained by* SPA*, defined in* (2.6)*, can be equivalently obtained by a procedure that incorporates the following steps:*

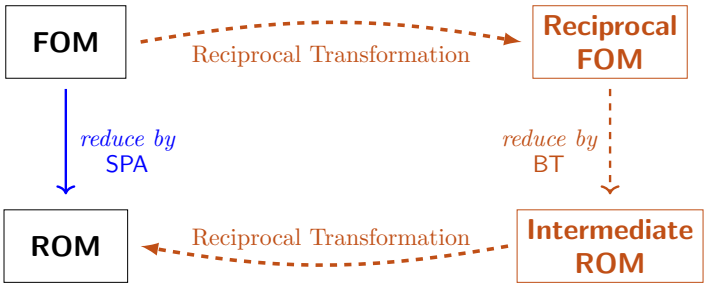


FIG. 1. *Commutative diagram and alternative characterization of SPA (following the brown dashed route) that is stated in Proposition* 3.4.

(i) *Apply the reciprocal transformation to the FOM.*
(ii) *Construct an intermediate reduced model of dimension $r$ for the reciprocal system, using BT.*
(iii) *Apply the reciprocal transformation to the intermediate reduced model.*

The proof of this proposition can be found in [29, 26] for the special case $\mathbf{E} = \mathbf{I}$. Since the ROM obtained by a balancing-related method does not depend on the state realization of the LTI system, it is clear that the proposition also holds for our consistently extended definition of a reciprocal system.

The last result presented here provides the relation between the Gramians of an LTI and those of the reciprocal system.

PROPOSITION 3.5. *The controllability and observability Gramians of the LTI system coincide with those of its reciprocal system in Definition* 3.1.

*Proof.* It can be proven that the Lyapunov equations of the LTI system and those of the reciprocal system are equivalent. Since these equations have unique solutions, this implies the equality of the Gramians. For example, the Lyapunov equation characterizing the controllability Gramian $\mathscr{P}$ of the reciprocal system reads, by definition, $\mathscr{A}\mathscr{P}\mathscr{E}^T + \mathscr{E}\mathscr{P}\mathscr{A}^T + \mathscr{B}\mathscr{B}^T = \mathbf{0}$. This is equivalent to

$$(\mathbf{E}\mathbf{A}^{-1}\mathbf{E})\mathscr{P}\mathbf{E}^T + \mathbf{E}\mathscr{P}(\mathbf{E}\mathbf{A}^{-1}\mathbf{E})^T + (\mathbf{E}\mathbf{A}^{-1})\mathbf{B}\mathbf{B}^T(\mathbf{E}\mathbf{A}^{-1})^T = \mathbf{0}.$$

By multiplying this equation by $(\mathbf{E}\mathbf{A}^{-1})^{-1}$ to the left and by $(\mathbf{E}\mathbf{A}^{-1})^{-T}$ to the right, the first Lyapunov equation in (2.4) is obtained, which implies $\mathscr{P} = \mathbf{P}$. The equality between the observability Gramians of the LTI system and of its reciprocal system can be shown to hold true in a similar manner. ∎

**4. Low-rank implementations of balancing-related methods.** In this section, we describe some archetypal strategies for implementing a low-rank variant of BT, based on the balancing-free square-root method. Afterward, the first main result is provided, consisting of the new balancing-free implementation of SPA. Its main advantage is that it allows for a low-rank variant similar to its BT counterpart. The gained computational efficiency is illustrated by a large-scale numerical example.

**4.1. Low-rank implementation of balanced truncation.** Since the original LTI system (i.e., the FOM) is assumed to be asymptotic stable, its Gramians are symmetric, positive semidefinite matrices. Thus, one may compute a factorization

$$\mathbf{P} = \mathbf{U}\mathbf{U}^T \quad \text{and} \quad \mathbf{Q} = \mathbf{L}\mathbf{L}^T \tag{4.1}$$

with $\mathbf{L}, \mathbf{U} \in \mathbb{R}^{n\times n}$, e.g., via a Cholesky factorization. This provides the essential elements for an implementation of BT by the square-root method outlined in Algorithm 4.1. It should be noted that, instead of first computing the Gramians explicitly and then the factors $\mathbf{L}$ and $\mathbf{U}$, one can directly compute the latter quantities by means of the Hammarling method implemented in the MATLAB function lyapchol or by a sign-function-based solver; cf. [12]. Such approaches hence circumvent potential numerical issues that may arise when computing Cholesky factors of the already computed Gramians. Another option for optimizing the computational steps is to replace the true Cholesky factors in (4.1) by approximated low-rank factors thereof. Let $\mathbf{P} \approx \breve{\mathbf{U}}\breve{\mathbf{U}}^T$ and $\mathbf{Q} \approx \breve{\mathbf{L}}\breve{\mathbf{L}}^T$, with $\breve{\mathbf{U}} \in \mathbb{R}^{n\times r_U}, \breve{\mathbf{L}} \in \mathbb{R}^{n\times r_L}$ having significantly fewer columns than the square-root factors, i.e., $r_L, r_U \ll n$. Then, the computation of a full singular value decomposition (SVD) for a large-scale matrix is averted, since $\mathbf{L}^T\mathbf{E}\mathbf{U} \in \mathbb{R}^{n\times n}$ is replaced by $\breve{\mathbf{L}}^T\mathbf{E}\breve{\mathbf{U}} \in \mathbb{R}^{r_L\times r_U}$. Such low-rank factors of the Gramians can be computed by means of, among others, iterative solvers for the Lyapunov equation based on the matrix sign function, low-rank alternating direction implicit (ADI) methods, Smith-type methods, etc. We refer the reader to the more detailed analysis on such approaches in [35, 27, 9, 41].

The low-rank adaptation of the square-root implementation of BT is remarkably straightforward: the exact factors of the Gramians have to be replaced by the approximated low-rank factors in Algorithm 4.1; see, e.g., [12, 25, 3]. This very simple adaptation follows essentially from the projection-based nature of BT. The underlying projection spaces consisting of the most observable and controllable states are efficiently approximated by the low-rank factors. The practical relevance of the low-rank implementations is not to be underestimated, as large-scale sparse systems appear in various applications in computational fluid dynamics and mechanical, electronic, chemical, and civil engineering. Specific applications are numerous and include microelectronics, aerodynamics, acoustics, electromagnetics, neuroscience, and chemical process optimization, to enumerate only a few. We refer the reader to [10, 11, 7] for more details.

---

**Algorithm 4.1** Balanced truncation (BT) (square-root/low-rank implementation).

**Require:** LTI system described by matrices $\mathbf{E}, \mathbf{A} \in \mathbb{R}^{n\times n}, \mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{p\times n}$, $\mathbf{D} \in \mathbb{R}^{p\times m}$.

**Ensure:** ROM: $\mathbf{A}_r \in \mathbb{R}^{r\times r}, \mathbf{B}_r \in \mathbb{R}^{r\times m}, \mathbf{C}_r \in \mathbb{R}^{p\times r}$, and $\mathbf{D}_r \in \mathbb{R}^{p\times m}$.

1: Compute the Lyapunov factors $\mathbf{U}, \mathbf{L} \in \mathbb{R}^{n\times n}$ from (4.1) and pick a reduced dimension $1 \le r \le \min(\mathsf{rank}(\mathbf{U}), \mathsf{rank}(\mathbf{L}))$.

2: Compute the SVD of the matrix $\mathbf{L}^T\mathbf{E}\mathbf{U}$, partitioned as follows:

$$\mathbf{L}^T\mathbf{E}\mathbf{U} = \begin{bmatrix}\mathbf{Z}_1 & \mathbf{Z}_2\end{bmatrix} \begin{bmatrix}\mathbf{S}_1 & \\ & \mathbf{S}_2\end{bmatrix} \begin{bmatrix}\mathbf{Y}_1^T \\ \mathbf{Y}_2^T\end{bmatrix},$$

where $\mathbf{S}_1 \in \mathbb{R}^{r\times r}$ and $\mathbf{S}_2 \in \mathbb{R}^{(n-r)\times(n-r)}$.

3: Construct the model reduction bases $\mathbf{W}_r = \mathbf{L}\mathbf{Z}_1\mathbf{S}_1^{-1/2}$ and $\mathbf{V}_r = \mathbf{U}\mathbf{Y}_1\mathbf{S}_1^{-1/2}$.

4: The reduced order system matrices are given by

$$\mathbf{A}_r = \mathbf{W}_r^T\mathbf{A}\mathbf{V}_r, \quad \mathbf{B}_r = \mathbf{W}_r^T\mathbf{B}, \quad \mathbf{C}_r = \mathbf{C}\mathbf{V}_r, \quad \text{and} \quad \mathbf{D}_r = \mathbf{D}.$$

---

**4.2. Low-rank implementation of singular perturbation approximation.** In contrast to BT, the reduction by SPA has, the the best of the authors' knowledge, no direct interpretation as a projection-based framework. Moreover, from the characterization (2.6), which is explicitly used in the algorithms proposed in [13, 3, 43], it was not clear how to explicitly derive a low-rank implementation of SPA.

We therefore consider the alternative procedure suggested by Proposition 3.4 for the derivation of our new algorithm. In what follows, a basic variant is sketched (this proves to be rather inefficient). Then, the novel efficient implementation, which allows for a low-rank adaptation, is derived from the latter by a few, yet crucial, modifications.

The basic procedure suggested by Proposition 3.4 will now be summarized. First, the reciprocal transformation is applied to the FOM to obtain the reciprocal system $(\mathscr{A}, \mathscr{B}, \mathscr{C}, \mathscr{D}, \mathscr{E})$. Then, the reduction bases $\mathscr{W}_r, \mathscr{V}_r \in \mathbb{R}^{n\times r}$ for the reciprocal system are determined (using BT on the reciprocal system), and the intermediate ROM is obtained by a projection step, i.e.,

$$\mathscr{A}_r = \mathscr{W}_r^T \mathscr{A} \mathscr{V}_r, \qquad \mathscr{B}_r = \mathscr{W}_r^T \mathscr{B}, \qquad \mathscr{C}_r = \mathscr{C}\mathscr{V}_r, \quad \text{and} \quad \mathscr{D}_r = \mathscr{D}. \tag{4.2}$$

As a final step, the reciprocal transformation is applied to the intermediate reduced model $(\mathscr{A}_r, \mathscr{B}_r, \mathscr{C}_r, \mathscr{D}_r)$; this is marked by the dashed brown arrow in the diagram displayed in Figure 1.

One possible issue with this procedure is that the reciprocal transformation of the FOM is explicitly needed, which requires the inverse of the matrix $\mathbf{A}$ corresponding to the FOM. Furthermore, note that the reciprocal system has dense state matrices even if those of the FOM are sparse. Thus, the procedure needs to be modified in such a way that the explicit determination of the full-order reciprocal system is omitted.

A first crucial observation in this direction is that the construction of $\mathscr{W}_r$ and $\mathscr{V}_r$ does not depend on the reciprocal system itself but only on its Gramians. By Proposition 3.5, the reciprocal system has the same Gramians as the FOM, and thus the reduction bases for applying BT to the reciprocal system are the same as for applying BT to the FOM, i.e., $\mathscr{W}_r = \mathbf{W}_r$ and $\mathscr{V}_r = \mathbf{V}_r$ (with $\mathbf{W}_r$ and $\mathbf{V}_r$ as in Algorithm 4.1). The second observation is that the projection step (4.2) for the intermediate ROM does not require the full reciprocal system either but can be obtained by solving a linear system of equations having moderate size. For example, the construction of the state matrix $\mathscr{A}_r$ of the intermediate ROM can be done according to the following:

$$\begin{aligned} \mathscr{A}_r = \mathscr{W}_r^T \mathscr{A} \mathscr{V}_r = \mathbf{W}_r^T \mathscr{A} \mathbf{V}_r = \mathbf{W}_r^T \mathbf{E}\mathbf{A}^{-1}\mathbf{E}\mathbf{V}_r \\ \iff \mathscr{A}_r = \mathbf{W}_r^T \mathbf{E} \mathscr{A}_V, \qquad \text{with } \mathscr{A}_V \text{ solving } \mathbf{A}\mathscr{A}_V = \mathbf{E}\mathbf{V}_r. \end{aligned} \tag{4.3}$$

The efficient construction of the other matrices $\mathscr{B}_r$, $\mathscr{C}_r$, and $\mathscr{D}_r$ follows similarly.

The newly proposed, efficient implementation of SPA, which follows by taking into account all the previously mentioned steps, is summarized in Algorithm 4.2. Similarly to BT, one can replace the exact factors of the Gramians ($\mathbf{L}$ and $\mathbf{U}$) in step 1 of this algorithm with low-rank approximate factors to obtain an efficient implementation for the large-scale setting. Two final practical hints for the implementation of Algorithm 4.2 are provided in Remark 4.1.

*Remark* 4.1. The inverse of the potentially large matrix $\mathbf{A}$ does not need to be explicitly formed in step 4 of Algorithm 4.2, but instead, a few linear equations can be solved. This is done by following the steps indicated in (4.3). Moreover, the last

**Algorithm 4.2** Singular perturbation approximation (SPA) (square-root/low-rank).

**Require:** FOM: $\mathbf{E}, \mathbf{A} \in \mathbb{R}^{n\times n}, \mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{p\times n}$, $\mathbf{D} \in \mathbb{R}^{p\times m}$.
**Ensure:** ROM: $\mathbf{A}_r \in \mathbb{R}^{r\times r}, \mathbf{B}_r \in \mathbb{R}^{r\times m}, \mathbf{C}_r \in \mathbb{R}^{p\times r}$, and $\mathbf{D}_r \in \mathbb{R}^{p\times m}$.

1: Compute the matrix factors $\mathbf{U}, \mathbf{L} \in \mathbb{R}^{n\times n}$ for approximating the solutions of Lyapunov equations (2.4) and choose reduced dimension $1 \le r \le \min(\mathsf{rank}(\mathbf{U}), \mathsf{rank}(\mathbf{L}))$.
2: Compute the SVD of the matrix $\mathbf{L}^T\mathbf{E}\mathbf{U}$ as follows:

$$\mathbf{L}^T\mathbf{E}\mathbf{U} = \begin{bmatrix}\mathbf{Z}_1 & \mathbf{Z}_2\end{bmatrix} \begin{bmatrix}\mathbf{S}_1 & \\ & \mathbf{S}_2\end{bmatrix} \begin{bmatrix}\mathbf{Y}_1^T \\ \mathbf{Y}_2^T\end{bmatrix},$$

where $\mathbf{S}_1 \in \mathbb{R}^{r\times r}$ and $\mathbf{S}_2 \in \mathbb{R}^{(n-r)\times(n-r)}$.
3: Construct the model reduction bases $\mathbf{W}_r = \mathbf{L}\mathbf{Z}_1\mathbf{S}_1^{-1/2}$ and $\mathbf{V}_r = \mathbf{U}\mathbf{Y}_1\mathbf{S}_1^{-1/2}$.
4: Calculate $\mathscr{A}_V = \mathbf{A}^{-1}(\mathbf{E}\mathbf{V}_r)$ and $\mathscr{B}_A = \mathbf{A}^{-1}\mathbf{B}$. Construct an intermediate ROM approximating the reciprocal system:

$$\mathscr{A}_r = \mathbf{W}_r^T\mathbf{E}\mathscr{A}_V, \quad \mathscr{B}_r = \mathbf{W}_r^T\mathbf{E}\mathscr{B}_A, \quad \mathscr{C}_r = -\mathbf{C}\mathscr{A}_V, \quad \text{and} \quad \mathscr{D}_r = \mathbf{D} - \mathbf{C}\mathscr{B}_A.$$

5: Get the ROM by means of the reciprocal transformation of the intermediate ROM, i.e.,

$$\mathbf{A}_r = \mathscr{A}_r^{-1}, \quad \mathbf{B}_r = \mathscr{A}_r^{-1}\mathscr{B}_r, \quad \mathbf{C}_r = -\mathscr{C}_r\mathscr{A}_r^{-1}, \quad \text{and} \quad \mathbf{D}_r = \mathscr{D}_r - \mathscr{C}_r\mathscr{A}_r^{-1}\mathscr{B}_r.$$

step in the algorithm is most efficiently implemented using a successive evaluation of the matrices using the following ordering. First, $\mathbf{A}_r = \mathscr{A}_r^{-1}$ is evaluated, then $\mathbf{B}_r = \mathbf{A}_r\mathscr{B}_r$, followed by $\mathbf{C}_r = -\mathscr{C}_r\mathbf{A}_r$. Finally, $\mathbf{D}_r = \mathscr{D}_r + \mathbf{C}_r\mathscr{B}_r$ is determined.

**4.3. Numerical study for the low-rank implementation.** It has been shown in several previous works that the use of low-rank Lyapunov equation solvers is beneficial in a large-scale setting. This allows one to solve for dimensions that are computationally infeasible for direct (dense) solvers, based mainly on classical algorithms proposed by Bartels–Stewart and Hammarling, and on more recent adaptations (see [42] for an overview). These results directly transfer to BT and, as we showcase in this section, also to our newly proposed Algorithm 4.2, which is the first low-rank implementation of SPA (to the best of our knowledge). In what follows, we refer to the method as "`Alg. 2, low-rank`" and compare it to its dense counterpart "`Alg. 2, dense`" as well as to the MATLAB built-in method `balred`, realizing SPA.

In this section, the numerical studies are performed for a benchmark problem described in [10, section 19] and referred to as Rail, which is given by a semidiscretized heat transfer model of a steel rail. Depending on the underlying mesh resolution of the discretization, the FOM has a dimension of $n \in \{109, 371, 1\,357, 5\,177, 20\,209, 79\,841\}$. Both the benchmark data and the low-rank (ADI-based) Lyapunov equation solvers originate from the `MESS` toolbox [39]. The numerical results have been generated using MATLAB version 9.12 (R2022a) on a desktop computer running with an Intel Core i7-8700 CPU with 32.0 GB. All reported results were generated with the code and benchmark data provided in [28].

Notably, regardless of which of the three implementations was chosen for the realization of SPA, almost no effect on the quality of the resulting ROM was observed for the test cases with $n \le 5\,177$. More precisely, the deviations with respect to the

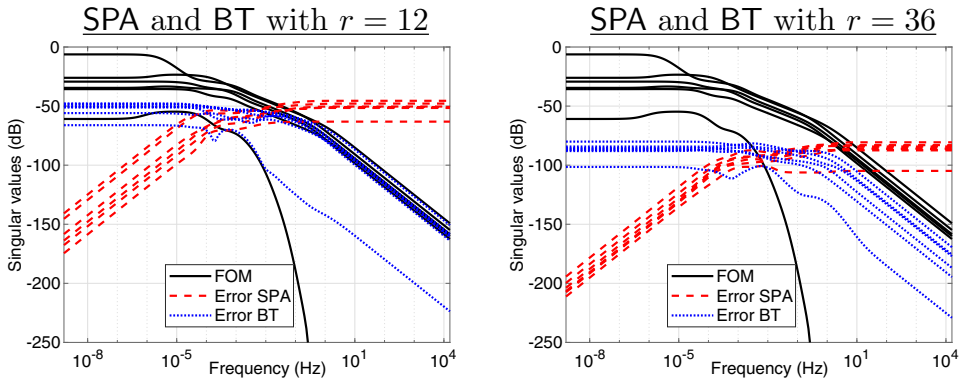


FIG. 2. *Rail (with $n = 20\,209$). Frequency response of FOM and the errors related to SPA and BT, using low-rank operations.*

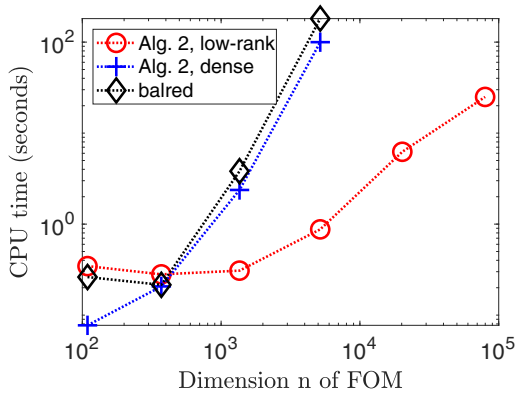


FIG. 3. *Rail with varying dimension $n$. CPU times for SPA realized by Algorithm 2 (using either low-rank or dense operations) and MATLAB built-in function balred ($r = 12$ for all ROMs).*

$\mathcal{H}_\infty$ norm were below $10^{-10}$. However, larger dimensions could not be considered for the dense solvers, since they did not run to completion on the running machine.

The qualitative behavior of the FOM and the approximation errors for SPA and BT are illustrated in the frequency domain in Figure 2 for two different choices of the ROM order, i.e., $r \in \{12, 36\}$. As expected, SPA performs better in the lower frequency range, while BT is more accurate in the higher frequency range. The maximum frequency-domain errors of the methods are fairly similar, ranging from about $-50db$ for $r = 12$ to about $-75db$ for $r = 36$.

However, the computational times of the different SPA implementations vary significantly as illustrated in Figure 3. While the two variants using dense solvers are almost the same in this respect, the low-rank implementation `Alg. 2, low-rank` is already seven times faster for a FOM dimension of $n = 1\,357$, i.e., 0.3 seconds compared to 2.8 seconds. The difference becomes even more pronounced for larger dimensions. The sparse solver is even faster for $n = 79\,841$ than `balred` for $n = 5\,177$. (`Alg. 2, low-rank` for $n = 79\,841$ required 25.0 seconds compared to `balred` for $n = 5\,177$, which needed 181.2 seconds.) Larger dimensions are not feasible for `balred` due to the limited internal memory of the computing machine. It should be noted that the very high memory demand of the dense solvers is a strong limitation in the large-scale setting.

**5. Data-driven implementation of balancing-related methods.** In [22], it was shown that BT can be (approximately) realized in a nonintrusive, data-driven manner. There, in the frequency domain, *"data" are transfer function evaluations,* or samples of the impulse response (and its derivative), for the time-domain case. This is fundamentally different from the other data-driven BT approaches in [31, 44, 38, 33], which require snapshots of the full state. The main idea underlying the approach in [22] is to make use only implicitly of quadrature approximations of the two system Gramians for the construction of a low-order approximately balanced model.

The novel contribution of this section is represented by a similar data-driven adaptation of SPA, which will be referred to as QuadSPA. Since this new approach follows principles similar to those of QuadBT in [22], we set the stage in subsection 5.1 by reviewing the basic construction underlying the latter. In subsection 5.2, QuadSPA is derived, and its numerical performance is examined in section 6.

Since we are interested in realization-free approaches, we are assuming $\mathbf{E} = \mathbf{I}$ from here on (the case of noninvertible matrix $\mathbf{E}$ is not treated in this contribution). This modification carries over to the FOM, i.e., instead of (2.1), we have

$$\dot{\mathbf{x}}(t) = \mathbf{A}\,\mathbf{x}(t) + \mathbf{B}\,\mathbf{u}(t), \quad \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t).$$

**5.1. Data-driven balanced truncation.** Following the BT implementation in Algorithm 4.1 (with $\mathbf{E} = \mathbf{I}$), square-root factors $\mathbf{U}$, and $\mathbf{L}$ of the Gramians are defined, and their product is approximated by a singular value decomposition, according to

$$\mathbf{L}^T\mathbf{U} \approx \mathbf{Z}_1\mathbf{S}_1\mathbf{Y}_1^T, \quad \mathbf{S}_1 \in \mathbb{R}^{r,r}.$$

By the definition of the reduction bases $\mathbf{W}_r = \mathbf{L}\mathbf{Z}_1\mathbf{S}_1^{-1/2}$ and $\mathbf{V}_r = \mathbf{U}\mathbf{Y}_1\mathbf{S}_1^{-1/2}$ (step 3 in the algorithm), the matrices of the ROM realization are expressed as

$$\mathbf{A}_r = \mathbf{W}_r^T\mathbf{A}\mathbf{V}_r = \mathbf{S}_1^{-1/2}\mathbf{Z}_1^T\,(\mathbf{L}^T\mathbf{A}\mathbf{U})\,\mathbf{Y}_1\mathbf{S}_1^{-1/2}, \quad \mathbf{B}_r = \mathbf{W}_r^T\mathbf{B} = \mathbf{S}_1^{-1/2}\mathbf{Z}_1^T\,(\mathbf{L}^T\mathbf{B}),$$
$$\mathbf{C}_r = \mathbf{C}\mathbf{V}_r = (\mathbf{C}\mathbf{U})\,\mathbf{Y}_1\mathbf{S}_1^{-1/2}.$$

By the latter relations, we observe that the ROM is fully characterized in terms of the following key quantities:

$$\mathbf{L}^T\mathbf{U}, \quad \mathbf{L}^T\mathbf{A}\mathbf{U}, \quad \mathbf{L}^T\mathbf{B}, \quad \text{and} \quad \mathbf{C}\mathbf{U}. \tag{5.1}$$

These matrices can be approximated from input-output data as shown next.

The starting point is a quadrature approximation of the Gramians in the frequency domain. Let $\imath$ be the complex unit with $\imath^2 = -1$. Then, the controllability Gramian $\mathbf{P}$ and the observability Gramian $\mathbf{Q}$. (i.e., the solutions to (2.4)) can be expressed as

$$\mathbf{P} = \frac{1}{2\pi}\int_{-\infty}^{\infty} (\mathit{i}\zeta\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{B}^T(-\mathit{i}\zeta\mathbf{I}^T - \mathbf{A}^T)^{-1}d\zeta,$$
$$\mathbf{Q} = \frac{1}{2\pi}\int_{-\infty}^{\infty} (-\mathit{i}\omega\mathbf{I} - \mathbf{A}^T)^{-1}\mathbf{C}\mathbf{C}^T(\mathit{i}\omega\mathbf{I} - \mathbf{A})^{-1}d\omega.$$

First, we consider a numerical quadrature rule that approximates the frequency integral defining the controllability Gramian, yielding an approximate Gramian $\widetilde{\mathbf{P}}$:

$$\mathbf{P} \approx \widetilde{\mathbf{P}} = \sum_{k=1}^{J} \rho_k^2 (\mathit{i}\omega_k\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{B}^T(-\mathit{i}\omega_k\mathbf{I}^T - \mathbf{A}^T)^{-1}. \tag{5.2}$$

In this formula, $\omega_k$ and $\rho_k^2$ denote the quadrature nodes and quadrature weights, respectively. Thus, $\rho_k$ are, by construction, the square-roots of the quadrature weights. Next, the quadrature-based Gramian approximation is decomposed as $\widetilde{\mathbf{P}} = \widetilde{\mathbf{U}}\widetilde{\mathbf{U}}^*$ with a square-root factor $\widetilde{\mathbf{U}} \in \mathbb{C}^{n\times Jm}$ defined as

$$\widetilde{\mathbf{U}} = \begin{bmatrix} \rho_1(\mathit{i}\omega_1\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} & \cdots & \rho_J(\mathit{i}\omega_J\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} \end{bmatrix}. \tag{5.3}$$

Note that both $\mathbf{P}$ and its quadrature approximation, $\widetilde{\mathbf{P}}$, are real-valued matrices, yet the explicit square-root factor, $\widetilde{\mathbf{U}}$, is overtly complex, and subsequent computation engages complex floating point arithmetic. However, in QuadBT, it is not required to explicitly compute the matrix $\widetilde{\mathbf{U}}$. The quadrature approximation of $\mathbf{Q}$ is defined by

$$\mathbf{Q} \approx \widetilde{\mathbf{Q}} = \sum_{j=1}^{J} \varphi_j^2 (-\mathit{i}\zeta_j\mathbf{I}^T - \mathbf{A}^T)^{-1}\mathbf{C}^T\mathbf{C}(\mathit{i}\zeta_j\mathbf{I} - \mathbf{A})^{-1},$$

with quadrature points $\zeta_j$ and quadrature weights $\varphi_j^2$ (for convenience we assume the number of quadrature points to be the same for $\mathbf{Q}$ and $\mathbf{P}$). The corresponding square-root factor decomposition reads $\widetilde{\mathbf{Q}} = \widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^*$, where

$$\widetilde{\mathbf{L}}^* = \begin{bmatrix} \varphi_1\mathbf{C}(\mathit{i}\zeta_1\mathbf{I} - \mathbf{A})^{-1} \\ \vdots \\ \varphi_J\mathbf{C}(\mathit{i}\zeta_J\mathbf{I} - \mathbf{A})^{-1} \end{bmatrix} \in \mathbb{C}^{Jp\times n}. \tag{5.4}$$

For a concise notation in the multi-input multi-output setting, the following two technical definitions are to be used.

DEFINITION 5.1. *Let $\boldsymbol{X} \in \mathbb{C}^{Jp\times Jm}$. Then, for $1 \le k, j \le J$, we say that the $(k,j)$th block $(p,m)$ entry of matrix $\boldsymbol{X}$ is a matrix in $\mathbb{C}^{p\times m}$, denoted with $\boldsymbol{X}_{k,j}$, for which its $p$ rows are a subset of the rows of matrix $\boldsymbol{X}$, indexed from $(k-1)p+1$ to $kp$. Additionally, the $m$ columns of the matrix $\boldsymbol{X}_{k,j}$ are a subset of the columns of the matrix $\boldsymbol{X}$, indexed from $(k-1)m+1$ to $km$.*

In the case of $m = p = 1$ (corresponding to the single-input single-output case), $\mathbf{X}_{k,j}$ in Definition 5.1 is nothing else but the $(k,j)$th scalar entry of matrix $\mathbf{X} \in \mathbb{C}^{J\times J}$.

DEFINITION 5.2. *Let $\boldsymbol{X} \in \mathbb{C}^{Jp\times m}$. Then, for $1 \le k \le J$, we say that the $k$th block $(p,m)$ entry of matrix $\boldsymbol{X}$ is a matrix in $\mathbb{C}^{p\times m}$, denoted with $\boldsymbol{X}_k$, for which its $p$ rows are a subset of the rows of matrix $\boldsymbol{X}$, indexed from $(k-1)p+1$ to $kp$. Additionally, the $m$ columns of matrix $\boldsymbol{x}_k$ are exactly the same as those of matrix $\boldsymbol{X}$ (an equivalent definition can be formulated for $\boldsymbol{X} \in \mathbb{C}^{p\times Jm}$; however, these details will be skipped).*

PROPOSITION 5.3. *Let* $\widetilde{\boldsymbol{U}}$ *and* $\widetilde{\boldsymbol{L}}$ *be as defined in* (5.3) *and* (5.4)*, whereby the quadrature points* $\omega_k \neq \zeta_j$ *are distinct for* $1 \leq k, j \leq J$*. Also, let*

$$\mathbf{H}_\infty(s) = \boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} = \mathbf{H}(s) - \lim_{\tilde{s}\to\infty} \mathbf{H}(\tilde{s}). \tag{5.5}$$

*Define the matrices* $\widetilde{\mathbf{N}} = \widetilde{\boldsymbol{L}}^* \widetilde{\boldsymbol{U}} \in \mathbb{C}^{Jp\times Jm}$ *and* $\widetilde{\mathbf{M}} = \widetilde{\boldsymbol{L}}^* \boldsymbol{A}\, \widetilde{\boldsymbol{U}} \in \mathbb{C}^{Jp\times Jm}$.

*Then, for* $1 \leq k, j \leq J$*, the* $(k, j)$*th block* $(p, m)$ *entries of matrices* $\widetilde{\mathbf{N}}$ *and* $\widetilde{\mathbf{M}}$*, respectively, read as (following Definition* 5.1*)*

$$\widetilde{\mathbf{N}}_{k,j} = -\rho_k \varphi_j \frac{\mathbf{H}_\infty(\imath\omega_k) - \mathbf{H}_\infty(\imath\zeta_j)}{\imath\omega_k - \imath\zeta_j},$$
$$\widetilde{\mathbf{M}}_{k,j} = -\rho_k \varphi_j \frac{\imath\omega_k \mathbf{H}_\infty(\imath\omega_k) - \imath\zeta_j \mathbf{H}_\infty(\imath\zeta_j)}{\imath\omega_k - \imath\zeta_j}.$$

*Likewise, defining* $\widetilde{\mathbf{T}} = \widetilde{\boldsymbol{L}}^* \boldsymbol{B} \in \mathbb{C}^{Jp\times m}$ *and* $\widetilde{\mathbf{G}}^T = \boldsymbol{C}^T \widetilde{\boldsymbol{U}} \in \mathbb{C}^{p\times Jm}$*, and by following Definition* 5.2*, we find*

$$\widetilde{\mathbf{T}}_k = \rho_k \mathbf{H}_\infty(\imath\omega_k), \quad \textit{and} \quad \widetilde{\mathbf{G}}_j = \varphi_j \mathbf{H}_\infty(\imath\zeta_j), \quad \textit{for } 1 \leq k, j \leq J.$$

By the previous considerations, one can conclude that the main advantage of QuadBT is that all necessary quantities are approximated by means of transfer function evaluations. More details and various extensions (e.g., using time-domain data or for discrete-time systems) can be found in [22]. Based on Proposition 5.3, the data-driven adaptation of BT can be implemented as shown in Algorithm 5.1. A proof of

**Algorithm 5.1** Quadrature-based balanced truncation (QuadBT).

**Require:** a). Two sets of quadrature weights and nodes for approximating an integral of the form $1/(2\pi)\int_{-\infty}^{\infty} f(s)ds$, given by $\omega_k$, $\rho_k$, respectively $\zeta_j$, $\varphi_j$ for $k, j = 1, 2, \ldots, J$ (assuming $\omega_k \neq \zeta_j$).

b). Evaluations of the FOM's transfer function $\mathbf{H}(s)$, at the quadrature nodes above.

c). Reduced dimension $r$, $1 \leq r \leq J\min(m, p)$.

**Ensure:** ROM: $\tilde{\mathbf{A}}_r \in \mathbb{R}^{r\times r}$, $\tilde{\mathbf{B}}_r \in \mathbb{R}^{r\times m}$, $\tilde{\mathbf{C}}_r \in \mathbb{R}^{p\times r}$, and $\tilde{\mathbf{D}}_r \in \mathbb{R}^{p\times m}$.

1: Determine feedthrough term $\mathbf{D} = \lim_{s\to\infty} \mathbf{H}(s)$.

2: Subtract the term $\mathbf{D}$ from the original data, and put together measurements $\{\mathbf{H}_\infty(\imath\omega_j)\}_{j=1}^J$ and $\{\mathbf{H}_\infty(\imath\zeta_k)\}_{k=1}^J$, where $\mathbf{H}_\infty(s) = \mathbf{H}(s) - \mathbf{D}$ is as in (5.5). Using the samples and the quadrature weights, construct $\widetilde{\mathbf{N}}, \widetilde{\mathbf{M}}$, $\widetilde{\mathbf{T}}$, and $\widetilde{\mathbf{G}}$ as in Proposition 5.3.

3: Compute the SVD of matrix $\widetilde{\mathbf{N}} \in \mathbb{C}^{Jp\times Jm}$ as

$$\widetilde{\mathbf{N}} = \begin{bmatrix}\widetilde{\mathbf{Z}}_1 & \widetilde{\mathbf{Z}}_2\end{bmatrix} \begin{bmatrix}\widetilde{\mathbf{S}}_1 & \\ & \widetilde{\mathbf{S}}_2\end{bmatrix} \begin{bmatrix}\widetilde{\mathbf{Y}}_1^* \\ \widetilde{\mathbf{Y}}_2^*\end{bmatrix},$$

where $\widetilde{\mathbf{S}}_1 \in \mathbb{R}^{r\times r}$ and $\widetilde{\mathbf{S}}_2 \in \mathbb{R}^{(Jp-r)\times(Jm-r)}$.

4: Construct the reduced order matrices:

$$\tilde{\mathbf{A}}_r = \widetilde{\mathbf{S}}_1^{-1/2} \widetilde{\mathbf{Z}}_1^* \widetilde{\mathbf{M}}\, \widetilde{\mathbf{Y}}_1 \widetilde{\mathbf{S}}_1^{-1/2}, \quad \tilde{\mathbf{B}}_r = \widetilde{\mathbf{S}}_1^{-1/2} \widetilde{\mathbf{Z}}_1^* \widetilde{\mathbf{T}}, \tilde{\mathbf{C}}_r = \widetilde{\mathbf{G}}^T \widetilde{\mathbf{Y}}_1 \widetilde{\mathbf{S}}_1^{-1/2}, \text{ and } \tilde{\mathbf{D}}_r = \mathbf{D}$$

the proposition can be found in Appendix A; this could provide additional insight and also contribute to setting the stage for the novel QuadSPA approach presented in the next subsection.

**5.2. Data-driven singular perturbation approximation.** The formulation of the novel data-driven QuadSPA implementation is presented in this subsection. Similarly to the low-rank implementation of SPA (subsection 4.2), we rely on a construction based on the alternative characterization of Proposition 3.4 and illustrated in Figure 1. First, an intermediate ROM $(\widetilde{\mathscr{A}_r}, \widetilde{\mathscr{B}}_r, \widetilde{\mathscr{C}_r}, \widetilde{\mathscr{D}}_r)$ approximating the reciprocal counterpart of the FOM is constructed. Then, by applying the reciprocal transformation of this low-dimensional intermediate ROM, we obtain the QuadSPA reduced system approximating the FOM.

However, the intermediate ROM is constructed from data here, following a similar quadrature approximation as in the data-driven BT; cf. subsection 5.1. This approximation requires a careful choice of quadrature rule. To derive an appropriate one, and to highlight the similarities to the QuadBT approach, we start from the reciprocal frequency representations of the Gramians $\mathscr{P}$ and $\mathscr{Q}$ and then rewrite them in terms of evaluations of the original transfer function in (2.2). Thus, we do not explicitly use the equality of the original and reciprocal Gramians (Proposition 3.5), which could also be used instead. Let

$$\mathscr{K} : \mathbb{C} \to \mathbb{C}^{n,n}, \qquad \mathscr{K}(s) = (s\mathbf{I} - \mathscr{A})^{-1},$$

so that the strictly proper part of the reciprocal transfer function $s \mapsto \mathscr{H}(s) - \mathscr{H}(\infty)$ can be equivalently expressed as $s \mapsto \mathscr{C}\mathscr{K}(s)\mathscr{B}$. Similarly to the derivation of QuadBT, we consider the frequency representations of the Gramians. The reciprocal controllability Gramian reads

$$\mathscr{P} = \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathscr{K}(\imath\zeta)\mathscr{B}\mathscr{B}^T\mathscr{K}(-\imath\zeta)^T d\zeta = \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathscr{K}\left(\frac{1}{\imath\zeta}\right)\mathscr{B}\mathscr{B}^T\mathscr{K}\left(\frac{1}{-\imath\zeta}\right)^T \frac{1}{\zeta^2} d\zeta,$$

whereby the last equality follows from integration using the substitution $s \to -1/s$ on $\mathbf{F}(\zeta) := \frac{1}{2\pi}\mathscr{K}(\imath\zeta)\mathscr{B}\mathscr{B}^T\mathscr{K}(-\imath\zeta)^T$, according to the following derivations:

$$\begin{aligned}\int_{-\infty}^{\infty} \mathbf{F}(\zeta)d\zeta &= \int_{-\infty}^{0} \mathbf{F}(\zeta)d\zeta + \int_{0}^{\infty} \mathbf{F}(\zeta)d\zeta \\ &= \int_{0}^{\infty} \mathbf{F}\left(-\frac{1}{\zeta}\right)\frac{1}{\zeta^2}d\zeta + \int_{-\infty}^{0} \mathbf{F}\left(-\frac{1}{\zeta}\right)\frac{1}{\zeta^2}d\zeta = \int_{-\infty}^{\infty} \mathbf{F}\left(-\frac{1}{\zeta}\right)\frac{1}{\zeta^2}d\zeta.\end{aligned}$$

Based on the reformulation, we consider the following approximation by quadrature,

$$\mathscr{P} \approx \tilde{\mathscr{P}} = \sum_{k=1}^{J} \left(\frac{\rho_k}{\omega_k}\right)^2 \mathscr{K}\left(\frac{1}{\imath\omega_k}\right)\mathscr{B}\mathscr{B}^T\mathscr{K}\left(\frac{1}{-\imath\omega_k}\right)^T,$$

making use of the same quadrature nodes $\omega_k$ and the same quadrature weights $\rho_k^2$ as for $\widetilde{\mathbf{P}}$ (in the context of QuadBT). This is analogous to the approximation provided by (5.2). We define the square-root factor $\widetilde{\mathscr{U}}$, which fulfills $\widetilde{\mathscr{P}} = \widetilde{\mathscr{U}}\widetilde{\mathscr{U}}^*$, as follows:

$$\widetilde{\mathscr{U}} = \left[ \frac{\rho_1}{\omega_1} \mathscr{K}\left(\frac{1}{\imath\omega_1}\right)\mathscr{B} \quad \cdots \quad \frac{\rho_J}{\omega_J} \mathscr{K}\left(\frac{1}{\imath\omega_J}\right)\mathscr{B} \right] \in \mathbb{C}^{n\times Jm}. \tag{5.6}$$

Similarly, the reciprocal observability Gramian is approximated by quadrature

$$\mathscr{Q} \approx \widetilde{\mathscr{Q}} = \sum_{k=1}^{J} \left(\frac{\varphi_j}{\zeta_j}\right)^2 \mathscr{K}\left(\frac{1}{-\imath\zeta_j}\right)^T \mathscr{C}^T\mathscr{C}\,\mathscr{K}\left(\frac{1}{\imath\zeta_j}\right),$$

using the same quadrature points $\zeta_j$ and the same quadrature weights $\varphi_j^2$ as for $\widetilde{\mathbf{Q}}$ (in the context of QuadBT). The corresponding square-root factor decomposition reads $\widetilde{\mathscr{Q}} = \widetilde{\mathscr{L}}\widetilde{\mathscr{L}}^*$, where

$$\widetilde{\mathscr{L}}^* = \begin{bmatrix} \frac{\varphi_1}{\zeta_1}\mathscr{C}\,\mathscr{K}\left(\frac{1}{\imath\zeta_1}\right) \\ \vdots \\ \frac{\varphi_J}{\zeta_J}\mathscr{C}\,\mathscr{K}\left(\frac{1}{\imath\zeta_J}\right) \end{bmatrix} \in \mathbb{C}^{Jp\times n}. \tag{5.7}$$

Certain key quantities of the reciprocal system can be directly approximated from data, as the following result shows. The construction is similar to the one used in Proposition 5.3.

PROPOSITION 5.4. *Let $\widetilde{\mathscr{U}}$ and $\widetilde{\mathscr{L}}$ be as defined in* (5.6) *and* (5.7)*, whereby the quadrature points $\omega_k \neq \zeta_j$ are distinct and nonzero for $1 \le k, j \le J$. Also, let*

$$\mathbf{H}_0 : \mathbb{C} \to \mathbb{C}^{p,m}, \qquad \mathbf{H}_0(s) = \mathbf{H}(s) - \mathbf{H}(0). \tag{5.8}$$

*Define the matrices $\widetilde{\mathscr{N}} = \widetilde{\mathscr{L}}^*\widetilde{\mathscr{U}} \in \mathbb{C}^{Jp\times Jm}$ and $\widetilde{\mathscr{M}} = \widetilde{\mathscr{L}}^*\mathscr{A}\widetilde{\mathscr{U}} \in \mathbb{C}^{Jp\times Jm}$. Then, for $1 \le k, j \le J$, the $(k,j)$th block $(p,m)$ entries of matrices $\widetilde{\mathscr{N}}$ and $\widetilde{\mathscr{M}}$, respectively, read (following Definition* 5.1*)*

$$\widetilde{\mathscr{N}}_{k,j} = -\rho_k\varphi_j \frac{\mathbf{H}_0(\imath\omega_k) - \mathbf{H}_0(\imath\zeta_j)}{\imath\omega_k - \imath\zeta_j},$$
$$\widetilde{\mathscr{M}}_{k,j} = -\rho_k\varphi_j \frac{(\imath\omega_k)^{-1}\mathbf{H}_0(\imath\omega_k) - (\imath\zeta_j)^{-1}\mathbf{H}_0(\imath\zeta_j)}{\imath\omega_k - \imath\zeta_j}.$$

*Likewise, defining $\widetilde{\mathscr{T}} = \widetilde{\mathscr{L}}^*\mathscr{B} \in \mathbb{C}^{Jp\times m}$ and $\widetilde{\mathscr{G}}^T = \mathscr{C}\widetilde{\mathscr{U}} \in \mathbb{C}^{p\times Jm}$, we find*

$$\widetilde{\mathscr{T}}_k = \frac{\rho_k}{\omega_k}\mathbf{H}_0(\imath\omega_k), \quad \text{and} \quad \widetilde{\mathscr{G}}_j = \frac{\varphi_j}{\zeta_j}\mathbf{H}_0(\imath\zeta_j), \quad \text{for } 1 \le k, j \le J.$$

*Proof.* For the following derivations, we introduce abbreviations $\nu = (\imath\omega_k)^{-1}$ and $\mu = (\imath\zeta_j)^{-1}$. Simple calculations show that

$$\mathscr{K}(\nu)\mathscr{K}(\mu) = \frac{1}{\nu-\mu}\mathscr{K}(\nu)\left[(\nu\mathbf{I} - \mathscr{A}) - (\mu\mathbf{I} - \mathscr{A})\right]\mathscr{K}(\mu) = -\frac{1}{\nu-\mu}\left(\mathscr{K}(\nu) - \mathscr{K}(\mu)\right).$$

This identity is also known as the "first resolvent formula"; e.g., see Theorem VI.5 in [37]. By using this formula, together with identities $\mathscr{C}\mathscr{K}(\nu)\mathscr{B} = \mathbf{H}_0(1/\nu) = \mathbf{H}_0(\imath\omega_k)$ and $\mathscr{C}\mathscr{K}(\mu)\mathscr{B} = \mathbf{H}_0(\imath\zeta_j)$, it follows that

$$\widetilde{\mathscr{N}}_{k,j} = (\rho_k/\omega_k)(\varphi_j/\zeta_j)\mathscr{C}\mathscr{K}(\nu)\mathscr{K}(\mu)\mathscr{B} = -\rho_k\varphi_j\frac{1}{\omega_k\zeta_j(\nu-\mu)}\mathscr{C}\left[\mathscr{K}(\nu) - \mathscr{K}(\mu)\right]\mathscr{B}$$
$$= -\rho_k\varphi_j\frac{\mathbf{H}_0(\imath\omega_k) - \mathbf{H}_0(\imath\zeta_j)}{\imath\omega_k - \imath\zeta_j},$$

whereby $\omega_k\zeta_j(\nu-\mu) = \imath\omega_k - \imath\zeta_j$ was used in the last step. Moreover, using the relation

$$\mathscr{K}(\nu)\mathscr{A}\mathscr{K}(\mu) = -\frac{1}{\nu-\mu}\left[\nu\mathscr{K}(\nu) - \mu\mathscr{K}(\mu)\right],$$

which is also known as the "second resolvent formula" (e.g., according to [37]), we can show in a similar manner that

$$\begin{aligned}\widetilde{\mathscr{M}}_{k,j} &= (\rho_k/\omega_k)(\varphi_j/\zeta_j)\,\mathscr{C}\left(\mathscr{K}(\nu)\mathscr{A}\mathscr{K}(\mu)\right)\mathscr{B}\\ &= -\rho_k\varphi_j\frac{1}{\omega_k\zeta_j(\nu-\mu)}\mathscr{C}\left(\nu\mathscr{K}(\nu) - \mu\mathscr{K}(\mu)\right)\mathscr{B}\\ &= -\rho_k\varphi_j\frac{(\imath\omega_k)^{-1}\mathbf{H}_0(\imath\omega_k) - (\imath\zeta_j)^{-1}\mathbf{H}_0(\imath\zeta_j)}{\imath\omega_k - \imath\zeta_j}.\end{aligned}$$

The explicit representations of $\widetilde{\mathscr{T}}$ and $\widetilde{\mathscr{G}}$ (as data matrices) are given as

$$\begin{aligned}\widetilde{\mathscr{T}}_k &= (\rho_k/\omega_k)\,\mathscr{C}\mathscr{K}(\imath\omega_k)\mathscr{B} = \frac{\rho_k}{\omega_k}\mathbf{H}_0(\imath\omega_k),\\ \widetilde{\mathscr{G}}_j &= (\varphi_j/\zeta_j)\,\mathscr{C}\mathscr{K}(\imath\zeta_j)\mathscr{B} = \frac{\varphi_j}{\zeta_j}\mathbf{H}_0(\imath\zeta_j).\end{aligned}$$

∎

The main contribution of this section, i.e., the data-driven, realization-free implementation of SPA is summarized in Algorithm 5.2.

*Remark* 5.5. A generalized version of Proposition 5.4 omitting the assumption $\omega_k \neq \zeta_j$ for all $j,k$ can be derived, but it requires evaluations of the derivative of the transfer function. For details, we refer the reader to Appendix B.

**5.3. Connections between QuadSPA and the Loewner framework.** In this subsection, we aim at revealing some connections that the newly developed method, QuadSPA, shares with the Loewner framework in [30]. Note that a connection between QuadBT and the Loewner framework has already been made in [22, section 3.4].

First, by comparing Proposition 5.4 and Proposition 5.3, it can be shown that the data matrix $\widetilde{\mathscr{N}}$ used in QuadSPA and the matrix $\widetilde{\mathbf{N}}$ used in QuadBT are exactly the same. They are actually diagonally scaled Loewner matrices. The other data matrices, i.e., $\widetilde{\mathbf{M}}$ in QuadBT and $\widetilde{\mathscr{M}}$ in QuadSPA, relate to diagonally scaled shifted Loewner matrices (see [30] for more details). In QuadBT, the shifts are given by $\imath\omega_k$, $1 \le k \le J$, while in QuadSPA, the "inverted shifts" $(\imath\omega_k)^{-1}$ are used instead.

This new interpretation of QuadSPA gives insight into some observations detailed in what follows. First, QuadSPA has a computational complexity similar to that of the Loewner framework. Choosing more data points $J$ will, in general, make the nonintrusive ROM a closer and closer approximation to the intrusively computed ROM but will also increase the dimension of the data matrices. Second, the quality of the ROM strongly depends on the quality of the data. When using data sets with perturbed (noisy) measurements, or that contain redundant data (not rich enough to capture the essential properties), the QuadSPA may face challenges similar to those of the Loewner framework. We intend to study these aspects in more detail for future works.

As was pointed out in preceding publications, the numerical conditioning of Loewner matrices can be an important issue. While the application of quadrature weights (as done in QuadBT and in QuadSPA) may improve the conditioning, regularization strategies might still be necessary. Of course, the SVD is an essential step that

**Algorithm 5.2** Quadrature-based Singular Perturbation Approximation (QuadSPA).

**Require:** a). Two sets of quadrature weights and nodes for approximating an integral of the form $1/(2\pi)\int_{-\infty}^{\infty} f(s)ds$, given by $\omega_k$, $\rho_k$, respectively $\zeta_j$, $\varphi_j$ for $k,j=1,2,\ldots,J$ (assuming $\omega_k \neq \zeta_j$);

b). Evaluations of the FOM's transfer function $\mathbf{H}(s)$;

c). Reduced dimension $r$, $1 \leq r \leq \min(Jp, Jm)$.

**Ensure:** ROM: $\tilde{\mathbf{A}}_r \in \mathbb{R}^{r\times r}$, $\tilde{\mathbf{B}}_r \in \mathbb{R}^{r\times m}$, $\tilde{\mathbf{C}}_r \in \mathbb{R}^{p\times r}$, and $\tilde{\mathbf{D}}_r \in \mathbb{R}^{p\times m}$

1: Determine the moment $\mathbf{D}_0 = \mathbf{H}(0)$.

2: Subtract the term $\mathbf{D}_0$ from the original data and put together measurements $\{\mathbf{H}_0(\mathrm{i}\omega_k)\}_{k=1}^J$ and $\{\mathbf{H}_0(\mathrm{i}\zeta_j)\}_{j=1}^J$, where $\mathbf{H}_0(s) = \mathbf{H}(s) - \mathbf{D}_0$ is as in (5.8). Using the samples and the quadrature weights, construct the data matrices $\widetilde{\mathscr{N}}$, $\widetilde{\mathscr{M}}$, $\widetilde{\mathscr{T}}$, and $\widetilde{\mathscr{G}}$ as in Proposition 5.4.

3: Compute the SVD of matrix $\widetilde{\mathscr{N}} \in \mathbb{C}^{Jp\times Jm}$:

$$\widetilde{\mathscr{N}} = \begin{bmatrix}\widetilde{\mathbf{Z}}_1 & \widetilde{\mathbf{Z}}_2\end{bmatrix}\begin{bmatrix}\widetilde{\mathbf{S}}_1 & \\ & \widetilde{\mathbf{S}}_2\end{bmatrix}\begin{bmatrix}\widetilde{\mathbf{Y}}_1^* \\ \widetilde{\mathbf{Y}}_2^*\end{bmatrix},$$

where $\widetilde{\mathbf{S}}_1 \in \mathbb{R}^{r\times r}$ and $\widetilde{\mathbf{S}}_2 \in \mathbb{R}^{(Jp-r)\times(Jm-r)}$.

4: Construct the intermediate reduced order matrices:

$$\widetilde{\mathscr{A}}_r = \widetilde{\mathbf{S}}_1^{-1/2}\widetilde{\mathbf{Z}}_1^*\widetilde{\mathscr{M}}\widetilde{\mathbf{Y}}_1\widetilde{\mathbf{S}}_1^{-1/2},\ \ \widetilde{\mathscr{B}}_r = \widetilde{\mathbf{S}}_1^{-1/2}\widetilde{\mathbf{Z}}_1^*\widetilde{\mathscr{T}},\ \ \widetilde{\mathscr{C}}_r = \widetilde{\mathscr{G}}^T\widetilde{\mathbf{Y}}_1\widetilde{\mathbf{S}}_1^{-1/2},\ \text{ and }\ \widetilde{\mathscr{D}}_r = \mathbf{D}_0.$$

5: Get the ROM by means of the reciprocal transformation of the intermediate ROM, i.e.,

$$\tilde{\mathbf{A}}_r = \widetilde{\mathscr{A}}_r^{-1},\ \ \tilde{\mathbf{B}}_r = \widetilde{\mathscr{A}}_r^{-1}\widetilde{\mathscr{B}}_r,\ \ \tilde{\mathbf{C}}_r = -\widetilde{\mathscr{C}}_r\widetilde{\mathscr{A}}_r^{-1},\ \text{ and }\ \tilde{\mathbf{D}}_r = \widetilde{\mathscr{D}}_r - \widetilde{\mathscr{C}}_r\widetilde{\mathscr{A}}_r^{-1}\widetilde{\mathscr{B}}_r.$$

acts as an implicit regularizer, removing the redundancies, provided that the truncation order is small enough. Other possible strategies are described in [16], while the robustness of the Loewner framework to noise or perturbations was investigated in [45, 19]. Also, we note that Algorithm 5.2 requires a complex SVD, which implies that the resulting ROM realization is, in general complex-valued. However, under the additional assumption that the quadrature points used in the approximation of either Gramian are chosen in complex-conjugate pairs, a real-valued ROM can be enforced. This can be done in a manner similar to that in [2, section A.1] for the Loewner framework, and in [22, section 4.1] for QuadBT.

**6. Numerical study for the data-driven implementation.** In this section, we compare the performance of the quadrature-based, data-driven implementation of SPA according to Algorithm 5.2, i.e., QuadSPA, with the following model reduction methods from the literature: the intrusive MATLAB built-in implementations of singular perturbation analysis (SPA) and balanced truncation (BT), and the quadrature-based implementation of BT, i.e., QuadBT from [22], realized by Algorithm 5.1.

We consider the following two benchmark examples, one of low dimension and one of higher dimension, both available in the MOR-Wiki[1] database; cf. [10]:

1. The LAbuild system models a motion problem in the Los Angeles University hospital building, with 8 floors, where each havs 3 degrees of freedom, namely

[1] https://morwiki.mpi-magdeburg.mpg.de/morwiki/index.php/Category:Benchmark.

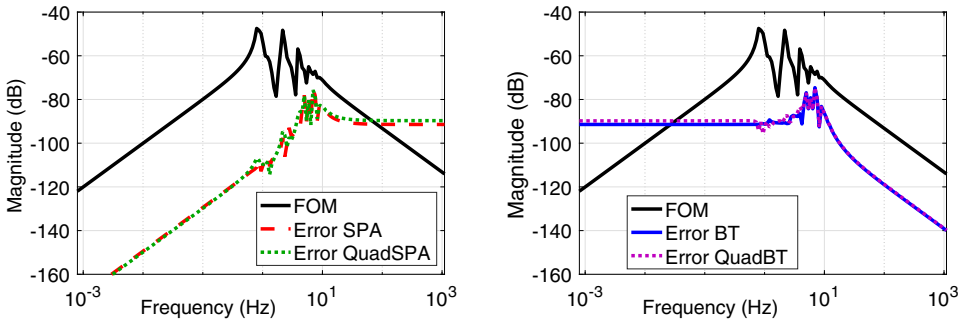


FIG. 4. *LAbuild. Frequency response of FOM and the errors using SPA and QuadSPA (left) and using BT and QuadBT (right);* $J = 100$ *and* $r = 18$ *are chosen for all ROMs.*

displacements in the $x$ and $y$ directions, and rotation. The model has $n = 48$ states, $m = 1$ input, and $p = 1$ output. We refer the reader to [18] for more details.

2. The ISS12A system models the structural response of the Russian Service 12A Module of the International Space Station (ISS). The model has $n = 1412$ states, $m = 3$ inputs, and $p = 3$ outputs. We refer the reader to [24] for more details.

In all reported experiments, the quadrature weights required for QuadSPA and QuadBT are determined according to the given transfer function measurements and to the exponentially convergent trapezoid quadrature rule used also in [22]. Other choices of quadrature rules are, of course, possible; we refer the reader to, e.g., [14, 15, 22], which consider other, more involved quadrature schemes. Extended details are provided below; we also refer to the MATLAB code provided in [28], which was used to generate the results.

**6.1. LAbuild model.** In the first experiment, we choose $J = 100$ logarithmically distributed (sampling) points in the range $\left[10^0, 10^2\right] \cdot i$. These will act as quadrature nodes. We compare the results of applying the two data-based methods against their intrusive counterparts, i.e., SPA and BT. We fix $r = 18$ as the dimension of all four ROMs for the first experiment. In the left pane of Figure 4, we depict the magnitude of the frequency response for the original system and approximation errors for the two ROMs computed with SPA and QuadSPA. In the right pane of Figure 4, we depict the magnitude of the frequency response for the original system and approximation errors for the two ROMs computed with BT and QuadBT. We notice that the SPA-based methods provide worse approximation quality in the higher frequency range, while the BT-based methods provide worse approximation quality in the lower frequency range. Additionally, it should be noticed that the error curve corresponding to QuadSPA faithfully reproduces that of SPA. The same can be said about BT.

In the next experiment, we vary the dimension $r$ of the ROM for all four methods in increments of 6. The number of data points is fixed to the same value as before, i.e., $J = 100$. We compute the $\mathcal{H}_\infty$ norm of the error systems produced by the four model reduction methods, scaled by the $\mathcal{H}_\infty$ norm of the original system. The numerical results are shown in Table 1. As expected, the approximation errors decrease as $r$ increases. Additionally, the quadrature-based methods produce errors comparable to the intrusive counterparts. Actually, for $r \in \{18, 24\}$, the QuadSPA method outperforms SPA in terms of the $\mathcal{H}_\infty$ norm approximation. However, for $r = 12$, the error

TABLE 1

*LAbuild. Relative approximation errors in the $\mathcal{H}_\infty$ norm for varying ROM order $r$ (QuadSPA and QuadBT, with $J = 100$).*

| | $r = 6$ | $r = 12$ | $r = 18$ | $r = 24$ | $r = 30$ |
|---|---|---|---|---|---|
| SPA | $2.4185 \cdot 10^{-1}$ | $9.3060 \cdot 10^{-2}$ | $3.7846 \cdot 10^{-2}$ | $1.0922 \cdot 10^{-2}$ | $9.0847 \cdot 10^{-4}$ |
| QuadSPA | $2.4971 \cdot 10^{-1}$ | $5.7480 \cdot 10^{-1}$ | $3.6713 \cdot 10^{-2}$ | $1.0836 \cdot 10^{-2}$ | $3.9822 \cdot 10^{-3}$ |
| BT | $2.3084 \cdot 10^{-1}$ | $1.0317 \cdot 10^{-1}$ | $3.8312 \cdot 10^{-2}$ | $1.0613 \cdot 10^{-2}$ | $9.4410 \cdot 10^{-4}$ |
| QuadBT | $2.7935 \cdot 10^{-1}$ | $1.0442 \cdot 10^{-1}$ | $3.8193 \cdot 10^{-2}$ | $1.0285 \cdot 10^{-2}$ | $4.5524 \cdot 10^{-3}$ |

TABLE 2

*LAbuild. Relative approximation errors in the $\mathcal{H}_\infty$ norm for varying quadrature points $J$, compared to relative errors of intrusive ROMs ($r = 18$ for all ROMs).*

| | $J = 20$ | $J = 30$ | $J = 50$ | $J = 70$ | $J = 100$ |
|---|---|---|---|---|---|
| QuadSPA | $3.4571 \cdot 10^{-1}$ | $3.2915 \cdot 10^{-1}$ | $6.0762 \cdot 10^{-1}$ | $7.0414 \cdot 10^{-2}$ | $3.6713 \cdot 10^{-2}$ |
| QuadBT | $7.9048 \cdot 10^{-1}$ | $3.8459 \cdot 10^{-1}$ | $1.9527 \cdot 10^{-1}$ | $9.9991 \cdot 10^{-2}$ | $3.8193 \cdot 10^{-2}$ |
| | SPA: $3.7846 \cdot 10^{-2}$ | | BT: $3.8312 \cdot 10^{-2}$ | | |

provided by QuadSPA is much higher than that of SPA (due to a pair of rogue poles), and a similar behavior is noticed for $r = 30$, both for QuadSPA and for QuadBT.

What we generally observe, is that a higher reduction order $r$ requires more quadrature nodes $J$ in QuadSPA and QuadBT in order to have fidelity similar to that of the intrusive methods SPA and BT, respectively. For example, in the latter experiment, we require about $J = 300$ points (three times more than before) to ensure that the quadrature-based methods reproduce the quality of the intrusive methods for $r = 30$. With this choice of parameters, the relative $\mathcal{H}_\infty$ error for QuadSPA is $9.3103 \cdot 10^{-4}$ for QuadSPA and $7.9862 \cdot 10^{-4}$ for QuadBT, which is very close to the errors of SPA and, interestingly, even slightly better than for BT, respectively; cf. Table 1.

For the next experiment, we vary instead the $J$ parameter between $[20, 100]$. We fix the dimension of the ROMs as $r = 18$. We compute again the relative approximation errors in the $\mathcal{H}_\infty$ norm for the two quadrature-based methods. The numerical results are shown in Table 2. As expected, for increasing values of $J$, the relative approximation errors are decreasing for both QuadSPA and QuadBT, and typically they reach the quality of their intrusive counterparts (SPA and BT). Notably, the quadrature-based methods can lead to better results in exceptional cases. This is observed here for $J = 100$. However, this by no means guarantees that for $J > 100$, the same trend is valid; e.g., when choosing $J = 120$, we noticed that BT outperforms QuadBT.

**6.2. ISS12A model.** For the ISS12A model, we first choose $J = 1000$ nodes, chosen to be logarithmically spaced in $[10^{-1}, 10^2] \cdot \imath$. Since the FOM has three inputs and three outputs, its frequency-response plot is composed of three curves, i.e., corresponding to the three singular values of $\mathbf{H}(\imath\omega_j) \in \mathbb{C}^{3\times 3}$, for any frequency point $\omega_j$. The reduction order of all ROMs is fixed at $r = 100$.

The frequency response plots of the FOM and the reduction errors related to four ROMs are depicted in Figure 5. We notice a good agreement between all responses and a comparable approximation quality for the intrusive and nonintrusive methods. Moreover, as to be expected, the SPA and QuadSPA methods lead to an improved fidelity in the low frequency range, whereas the errors for BT and QuadBT

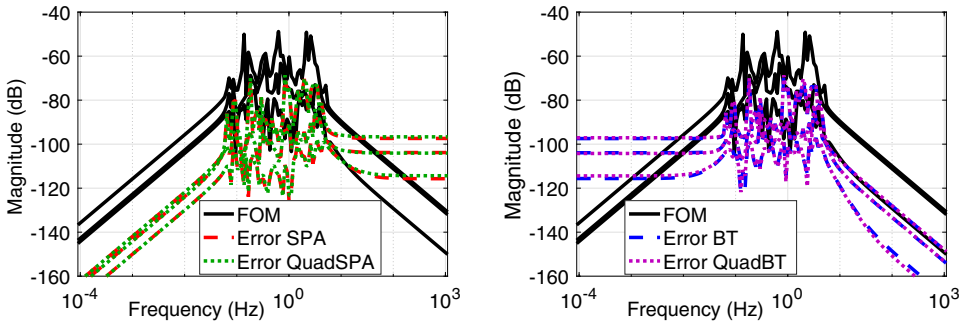


FIG. 5. *ISS12A. Frequency response of FOM and the errors using SPA and QuadSPA (left) and using BT and QuadBT (right); $J = 1000$ and $r = 100$ for all ROMs.*

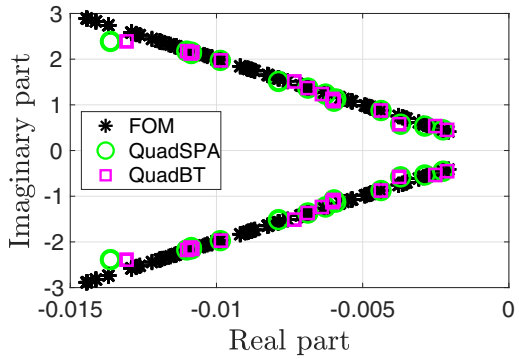


FIG. 6. *ISS12A. Distribution of system poles for QuadBT and QuadSPA; $r = 100$ and $J = 1000$ were used.*

are specifically small for high frequencies.

In Figure 6 we depict the configuration of the system poles for the FOM and for the two ROMs based on quadrature approximations. The dominant poles of the original system seem to be well matched in both cases. Moreover, we would like to mention that the poles in the ROMs occur in complex-conjugate pairs, which is in line with the real-valued property of the fitted models.

In the next experiment, we vary the number of points $J$ between 600 and 1 000, and we compute the relative approximation errors in the $\mathcal{H}_\infty$ norm between the original system and the two ROMs, i.e., computed by means of QuadSPA and QuadBT. As expected, the approximation error decreases as $J$ increases, as shown in Table 3.

Finally, we compute absolute deviations in the $\mathcal{H}_\infty$ norm between the ROMs computed with intrusive model reduction methods (SPA and BT) and their data-based counterparts. We vary $J$ between 200 and 1 000 and compute the $\mathcal{H}_\infty$ quantities in Table 4. As the number of data points is increased, the deviation between the quadrature-based implementations QuadSPA and QuadBT and their intrusive implementation generally decreases and becomes negligible compared to the reduction error of the method.

**7. Conclusion.** This paper addressed some open issues corresponding to the practical usability of the SPA approach. Specifically, two new algorithmic implementations of this model order reduction method were proposed with different use cases

TABLE 3
*ISS12A. Relative approximation errors in the $\mathcal{H}_\infty$ norm for varying number $J$ of quadrature points (ROM dimension $r = 100$).*

| | $J = 600$ | $J = 800$ | $J = 1000$ |
|---|---|---|---|
| QuadSPA | $5.6244 \cdot 10^{-1}$ | $1.6758 \cdot 10^{-1}$ | $3.2221 \cdot 10^{-2}$ |
| QuadBT | $6.0982 \cdot 10^{-1}$ | $4.4826 \cdot 10^{-2}$ | $3.2171 \cdot 10^{-2}$ |
| SPA: $3.3153 \cdot 10^{-2}$ | | BT: $3.2941 \cdot 10^{-2}$ | |

TABLE 4
*ISS12A. $\mathcal{H}_\infty$ norm deviations between intrusive and quadrature-based methods for varying $J$ (ROM dimension is $r = 100$).*

| | $J = 200$ | $J = 600$ | $J = 800$ | $J = 1000$ |
|---|---|---|---|---|
| SPA versus QuadSPA | $1.7813 \cdot 10^{-2}$ | $6.6425 \cdot 10^{-3}$ | $1.9466 \cdot 10^{-3}$ | $2.6942 \cdot 10^{-4}$ |
| BT versus QuadBT | $2.0562 \cdot 10^{-2}$ | $7.1800 \cdot 10^{-3}$ | $4.3901 \cdot 10^{-4}$ | $2.9588 \cdot 10^{-4}$ |

in mind. The first one was a low-rank implementation that can be used for the reduction of systems much larger than those manageable by existing implementations (based mostly on dense solvers). Notably, our SPA implementation has about the same computational complexity as the state-of-the-art algorithms for BT. Second, we derived a data-driven, nonintrusive reinterpretation of SPA that is based on a quadrature approximation requiring solely input-output data. Since these data could also be obtained by measurements, the algorithm can be considered realization-free. It shares connections with the data-driven reinterpretation of BT from [22] and thus behaves similarly in some sense. For example, the choice of data is shown to be crucial for the approximation quality, similarly to the Loewner framework. The findings of this paper were validated by several numerical tests, which illustrated the good correspondence between our new contributions for SPA and the well-established results for BT. Future research endeavors could include the study of generalized versions of SPA with techniques similar to the ones used in this paper, for both intrusive and data-driven settings.

**Appendix A. Proof of Proposition 5.3.** For completeness, and since the proof of our main resultProposition 5.4 follows using similar arguments, we state a proof for Proposition 5.3 in the following; cf. [22].

Let the assumptions of the proposition hold, and let $\mathbf{K} : \mathbb{C} \to \mathbb{C}^{n,n}$ be given by $\mathbf{K}(s) = (s\mathbf{I} - \mathbf{A})^{-1}$, so that $\mathbf{H}_\infty(s) = \mathbf{C}\mathbf{K}(s)\mathbf{B}$, as well as

$$\widetilde{\mathbf{L}}^* = \begin{bmatrix} \rho_1 \mathbf{C}\mathbf{K}(\imath\omega_1) \\ \vdots \\ \rho_J \mathbf{C}\mathbf{K}(\imath\omega_J) \end{bmatrix} \quad \text{and} \quad \widetilde{\mathbf{U}} = \begin{bmatrix} \rho_1 \mathbf{K}(\imath\omega_1)\mathbf{B} & \cdots & \rho_J \mathbf{K}(\imath\omega_J)\mathbf{B} \end{bmatrix}.$$

A direct calculation shows that

$$\begin{aligned} \mathbf{K}(\imath\omega_k)\mathbf{K}(\imath\zeta_j) &= \frac{1}{\imath\omega_k - \imath\zeta_j} \mathbf{K}(\imath\omega_k) \left[ (\imath\omega_k \mathbf{I} - \mathbf{A}) - (\imath\zeta_j \mathbf{I} - \mathbf{A}) \right] \mathbf{K}(\imath\zeta_j) \\ &= -\frac{1}{\imath\omega_k - \imath\zeta_j} \left( \mathbf{K}(\imath\omega_k) - \mathbf{K}(\imath\zeta_j) \right). \end{aligned}$$

By the latter, we conclude that

$$\begin{aligned}\widetilde{\mathbf{N}}_{k,j} = \widetilde{\mathbf{L}}_k \widetilde{\mathbf{U}}_j &= \rho_k \varphi_j \mathbf{C}\mathbf{K}(\imath\omega_k)\mathbf{K}(\imath\zeta_j)\mathbf{B} = -\frac{\rho_k \varphi_j}{\imath\omega_k - \imath\zeta_j}\left(\mathbf{C}\mathbf{K}(\imath\omega_k)\mathbf{B} - \mathbf{C}\mathbf{K}(\imath\zeta_j)\mathbf{B}\right) \\ &= -\rho_k \varphi_j \frac{\mathbf{H}_\infty(\imath\omega_k) - \mathbf{H}_\infty(\imath\zeta_j)}{\imath\omega_k - \imath\zeta_j}.\end{aligned}$$

Similarly, by using the fact that

$$\begin{aligned}\mathbf{K}(\imath\omega_k)\mathbf{A}\mathbf{K}(\imath\zeta_j) &= \frac{1}{\imath\omega_k - \imath\zeta_j}\mathbf{K}(\imath\omega_k)\left[\imath\zeta_j(\imath\omega_k\mathbf{I} - \mathbf{A}) - \imath\omega_k(\imath\zeta_j\mathbf{I} - \mathbf{A})\right]\mathbf{K}(\imath\zeta_j) \\ &= -\frac{1}{\imath\omega_k - \imath\zeta_j}\left(\imath\omega_k\mathbf{K}(\imath\omega_k) - \imath\zeta_j\mathbf{K}(\imath\zeta_j)\right),\end{aligned}$$

we can write

$$\begin{aligned}\widetilde{\mathbf{M}}_{k,j} = \rho_k \varphi_j \mathbf{C}\mathbf{K}(\imath\omega_k)\mathbf{A}\mathbf{K}(\imath\zeta_j)\mathbf{B} &= -\frac{\rho_k \varphi_j}{\imath\omega_k - \imath\zeta_j}\left(\imath\omega_k\mathbf{C}\mathbf{K}(\imath\omega_k)\mathbf{B} - \imath\zeta_j\mathbf{C}\mathbf{K}(\imath\zeta_j)\mathbf{B}\right) \\ &= -\rho_k \varphi_j \frac{\imath\omega_k\mathbf{H}_\infty(\imath\omega_k) - \imath\zeta_j\mathbf{H}_\infty(\imath\zeta_j)}{\imath\omega_k - \imath\zeta_j}.\end{aligned}$$

Finally, the representation of $\widetilde{\mathbf{T}}$ and $\widetilde{\mathbf{G}}$ (as data matrices) claimed in the proposition also follows in a straightforward manner.

**Appendix B. An extension of Proposition 5.4.** The assumption $\omega_k \neq \zeta_j$ for $k, j \in \{1, \ldots, J\}$ can be omitted in Proposition 5.4. However, in those cases, samples corresponding to the derivative of the transfer function are required.

The required adaptations concern the data matrices $\widetilde{\mathscr{N}}$ and $\widetilde{\mathscr{M}}$ only, given as

$$\widetilde{\mathscr{N}}_{k,j} = \begin{cases} -\rho_k \varphi_j \dfrac{\mathbf{H}_0(\imath\omega_k) - \mathbf{H}_0(\imath\zeta_j)}{\imath\omega_k - \imath\zeta_j}, & \omega_k \neq \zeta_j, \\ -\rho_k \varphi_j \frac{d}{ds}\mathbf{H}_0(s)_{s=\imath\zeta_j}, & \omega_k = \zeta_j, \end{cases}$$

$$\widetilde{\mathscr{M}}_{k,j} = \begin{cases} -\rho_k \varphi_j \dfrac{(\imath\omega_k)^{-1}\mathbf{H}_0(\imath\omega_k) - (\imath\zeta_j)^{-1}\mathbf{H}_0(\imath\zeta_j)}{\imath\omega_k - \imath\zeta_j}, & \omega_k \neq \zeta_j \\ -\rho_k \varphi_j \left((\imath\zeta_j)^{-1}\frac{d}{ds}\mathbf{H}_0(s)_{s=\imath\zeta_j} - (\imath\zeta_j)^{-2}\mathbf{H}_0(\imath\zeta_j)\right), & \omega_k = \zeta_j. \end{cases}$$

In light of the proof that was presented for Proposition 5.4, it only remains to show the validity of the representations $\widetilde{\mathscr{N}}_{k,j}$ and $\widetilde{\mathscr{M}}_{k,j}$ for the special case $\omega_k = \zeta_j$. We do this by considering the limit $\omega_k \to \zeta_j$ of the representations that have already been shown for $\omega_k \neq \zeta_j$ using the classical L'Hospital's rule.

For $\widetilde{\mathscr{M}}_{k,j}$ with $\omega_k = \zeta_j$, the crucial step is

$$\lim_{\omega_k \to \zeta_j} \frac{\mathbf{H}_0(\imath\omega_k) - \mathbf{H}_0(\imath\zeta_j)}{\imath\omega_k - \imath\zeta_j} = \lim_{\omega_k \to \zeta_j} \frac{\frac{d}{ds}\left(\mathbf{H}_0(s) - \mathbf{H}_0(\imath\zeta_j)\right)_{s=\imath\omega_k}}{\frac{d}{ds}\left(s - \imath\zeta_j\right)_{s=\imath\omega_k}} = \frac{d}{ds}\mathbf{H}_0(s)_{s=\imath\zeta_j}.$$

From the latter result, the claimed representation directly follows. Similarly, the expression for $\widetilde{\mathscr{N}}_{k,j}$ and $\omega_k = \zeta_j$ can be shown using

$$\begin{aligned}\lim_{\omega_k \to \zeta_j} \frac{(\imath\omega_k)^{-1}\mathbf{H}_0(\imath\omega_k) - (\imath\zeta_j)^{-1}\mathbf{H}_0(\imath\zeta_j)}{\imath\omega_k - \imath\zeta_j} &= \frac{d}{ds}\left(s^{-1}\mathbf{H}_0(s)\right)_{s=\imath\zeta_j} \\ &= (\imath\zeta_j)^{-1}\frac{d}{ds}\mathbf{H}_0(s)_{s=\imath\zeta_j} - (\imath\zeta_j)^{-2}\mathbf{H}_0(\imath\zeta_j).\end{aligned}$$

**Reproducibility of computational results.** This paper has been awarded the "SIAM Reproducibility Badge: code and data available" as a recognition that the authors have followed reproducibility principles valued by SISC and the scientific computing community. Code and data that allow readers to reproduce the results in this paper are available at https://doi.org/10.5281/zenodo.7671264 as well as in the accompanying supplementary materials (CodeBalancedSPA.zip [local/web 6.62MB]).

**Acknowledgments.** The authors would like to thank Peter Benner and the anonymous referees for providing insightful comments that helped to improve the manuscript.

REFERENCES

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

[2] A. C. Antoulas, C. A. Beattie, and S. Güğercin, *Interpolatory Methods for Model Reduction*, Comput. Sci. Engrg. 21, SIAM, Philadelphia, 2020, https://doi.org/10.1137/1.9781611976083.

[3] U. Baur and P. Benner, *Gramian-based model reduction for data-sparse systems*, SIAM J. Sci. Comput., 31 (2008), pp. 776–798, https://doi.org/10.1137/070711578.

[4] P. Benner and L. Feng, *Model order reduction based on moment-matching*, in Model Order Reduction, Volume 1: System-and Data-Driven Methods and Algorithms, P. Benner, et al., eds., De Gruyter, Berlin, 2021, pp. 57–96, https://doi.org/10.1515/9783110498967-003.

[5] P. Benner, P. Goyal, B. Kramer, B. Peherstorfer, and K. Willcox, *Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms*, Comput. Methods Appl. Mech. Engrg., 372 (2020), 113433, https://doi.org/10.1016/j.cma.2020.113433.

[6] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. H. A. Schilders, and L. M. Silveira, eds., *Model Order Reduction*, *Volume* 1: *System- and Data-Driven Methods and Algorithms*, De Gruyter, Berlin, 2021, https://doi.org/10.1515/9783110498967.

[7] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. H. A. Schilders, and L. M. Silveira, eds., *Model Order Reduction*, *Volume* 3: *Applications*, De Gruyter, Berlin, 2021, https://doi.org/10.1515/9783110499001.

[8] P. Benner, S. Gugercin, and K. Willcox, *A survey of projection-based model reduction methods for parametric dynamical systems*, SIAM Rev., 57 (2015), pp. 483–531, https://doi.org/10.1137/130932715.

[9] P. Benner, P. Kürschner, and J. Saak, *Efficient handling of complex shift parameters in the low-rank Cholesky factor ADI method*, Numer. Algorithms, 62 (2013), pp. 225–251, https://doi.org/10.1007/s11075-012-9569-7.

[10] P. Benner, D. C. Sorensen, and V. Mehrmann, *Dimension Reduction of Large-Scale Systems*, Lect. Notes Comput. Sci. Eng. 45, Springer-Verlag, Berlin/Heidelberg, Germany, 2005, https://doi.org/10.1007/3-540-27909-1.

[11] P. Benner, M. Ohlberger, A. Cohen, and K. Willcox, eds., *Model Reduction and Approximation: Theory and Algorithms*, Comput. Sci. Eng., SIAM, Philadelphia, 2017, https://doi.org/10.1137/1.9781611974829.

[12] P. Benner, E. S. Quintana-Ortí, and G. Quintana-Ortí, *Balanced truncation model reduction of large-scale dense systems on parallel computers*, Math. Comput. Model. Dyn. Syst., 6 (2000), pp. 383–405, https://doi.org/10.1076/mcmd.6.4.383.3658.

[13] P. Benner, E. S. Quintana-Ortí, and G. Quintana-Ortí, *Singular perturbation approximation of large, dense linear systems*, in CACSD, Conference Proceedings, IEEE International Symposium on Computer-Aided Control System Design, IEEE, 2000, pp. 255–260, https://doi.org/10.1109/CACSD.2000.900220.

[14] P. Benner and A. Schneider, *Balanced truncation model order reduction for LTI systems with many inputs or outputs*, in Proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems, A. Edelmayer, ed., Budapest, Hungary, 2010, pp. 1971–1974, isbn: 978-963-311-370-7.

[15] C. Bertram and H. Fassbender, *A quadrature framework for solving Lyapunov and Sylvester equations*, Linear Algebra Appl., 622 (2022), pp. 66–103, https://doi.org/10.1016/j.laa.2021.03.029.

[16] A. Binder, V. Mehrmann, A. Miedlar, and P. Schulze, *A MATLAB Toolbox for the Regularization of Descriptor Systems Arising from Generalized Realization Procedures*, preprint, https://arxiv.org/pdf/2212.02604.pdf, 2022.

[17] T. Breiten and T. Stykel, *Balancing-related model reduction methods*, in Model Order Reduction, Volume 1: System-and Data-Driven Methods and Algorithms, P. Benner, et al., eds., De Gruyter, Berlin, 2021, pp. 15–56, https://doi.org/10.1515/9783110498967-002.

[18] Y. Chahlaoui and P. Van Dooren, *A Collection of Benchmark Examples for Model Reduction of Linear Time Invariant Dynamical Systems*, Technical report 2002-2, SLICOT Working Note, 2002, www.slicot.org.

[19] Z. Drmač and B. Peherstorfer, *Learning low-dimensional dynamical-system models from noisy frequency-response data with Loewner rational interpolation*, in Realization and Model Reduction of Dynamical Systems, C. Beattie, et al., eds., Springer, Cham, 2022, pp. 39–57, https://doi.org/10.1007/978-3-030-95157-3˙3.

[20] K. V. Fernando and H. Nicholson, *Singular perturbational model reduction of balanced systems*, IEEE Trans. Automat. Control, 27 (1982), pp. 466–468, https://doi.org/10.1109/TAC.1982.1102932.

[21] K. Glover, *All optimal Hankel-norm approximations of linear multivariable systems and their $L^\infty$-error norms*, Internat. J. Control, 39 (1984), pp. 1115–1193, https://doi.org/10.1080/00207178408933239.

[22] I. V. Gosea, S. Gugercin, and C. Beattie, *Data-driven balancing of linear dynamical systems*, SIAM J. Sci. Comput., 44 (2022), pp. A554–A582, https://doi.org/10.1137/21M1411081.

[23] M. Green and D. J. N. Limebeer, *Linear Robust Control*, Dover Books on Electrical Engineering, reprint, Dover Publications, 2012.

[24] S. Gugercin, A. C. Antoulas, and M. Bedrossian, *Approximation of the International Space Station 1$R$ and 12$A$ models*, in Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No.01CH37228), Orlando, FL, 2001, pp. 1515–1516, https://doi.org/10.1109/CDC.2001.981109.

[25] S. Gugercin and J.-R. Li, *Smith-type methods for balanced truncation of large systems*, in Dimension Reduction of Large-Scale Systems, Lect. Notes Comput. Sci. Eng. 45, P. Benner, D. C. Sorensen, and V. Mehrmann, eds., Springer-Verlag, Berlin, 2005, pp. 49–82, https://doi.org/10.1007/3-540-27909-1_2.

[26] C. Guiver, *The generalised singular perturbation approximation for bounded real and positive real control systems*, Math. Control Relat. Fieds, 9 (2017), pp. 313–350, https://doi.org/10.3934/mcrf.2019016.

[27] J.-R. Li and J. White, *Low rank solution of Lyapunov equations*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 260–280, https://doi.org/10.1137/S0895479801384937.

[28] B. Liljegren-Sailer and I. V. Gosea, *Code for paper 'Data-driven and low-rank implementations of balanced singular perturbation approximation'*, 2023, https://doi.org/10.5281/zenodo.7671264.

[29] Y. Liu and B. D. O. Anderson, *Singular perturbation approximation of balanced systems*, Internat. J. Control, 50 (1989), pp. 1379–1405, https://doi.org/10.1080/00207178908953437.

[30] A. J. Mayo and A. C. Antoulas, *A framework for the solution of the generalized realization problem*, Linear Algebra Appl., 425 (2007), pp. 634–662, https://doi.org/10.1016/j.laa.2007.03.008.

[31] B. Moore, *Principal component analysis in linear systems: Controllability, observability, and model reduction*, IEEE Trans. Automat. Control, 26 (1981), pp. 17–32, https://doi.org/10.1109/TAC.1981.1102568.

[32] C. Mullis and R. Roberts, *Synthesis of minimum roundoff noise fixed point digital filters*, IEEE Trans. Circuits Syst., 23 (1976), pp. 551–562, https://doi.org/10.1109/TCS.1976.1084254.

[33] M. R. Opmeer, *Model order reduction by balanced proper orthogonal decomposition and by rational interpolation*, IEEE Trans. Automat. Control, 57 (2012), pp. 472–477, https://doi.org/10.1109/TAC.2011.2164018.

[34] B. Peherstorfer and K. Willcox, *Data-driven operator inference for nonintrusive projection-based model reduction*, Comput. Methods Appl. Mech. Engrg., 306 (2016), pp. 196–215, https://doi.org/10.1016/j.cma.2016.03.025.

[35] T. Penzl, *A cyclic low-rank Smith method for large sparse Lyapunov equations*, SIAM J. Sci. Comput., 21 (2000), pp. 1401–1418, https://doi.org/10.1137/S1064827598347666.

[36] A. Quarteroni, A. Manzoni, and F. Negri, *Reduced Basis Methods for Partial Differential Equations*, Unitext 92, La Mat. per il 3+2, Springer, Cham, 2016.

[37] M. C. Reed and B. Simon, *Methods of Modern Mathematical Physics* I*: Functional Analysis*, Vol. 1, Academic Press, New York, 1972.

[38] C. W. Rowley, *Model reduction for fluids, using balanced proper orthogonal decomposition*, Internat. J. Bifur. Chaos, 15 (2005), pp. 997–1013, https://doi.org/10.1142/S0218127405012429.

[39] J. Saak, M. Köhler, and P. Benner, *M-M.E.S.S. – The Matrix Equations Sparse Solvers Library*, Version 2.1, 2021, https://doi.org/10.5281/zenodo.4719688.

[40] P. J. Schmid, *Dynamic mode decomposition of numerical and experimental data*, J. Fluid Mech., 656 (2010), pp. 5–28, https://doi.org/10.1017/S0022112010001217.

[41] V. Simoncini, *Computational methods for linear matrix equations*, SIAM Rev., 58 (2016), pp. 377–441, https://doi.org/10.1137/130912839.

[42] D. C. Sorensen and Y. Zhou, *Direct methods for matrix Sylvester and Lyapunov equations*, J. Appl. Math., 2003 (2003), 245057, https://doi.org/10.1155/S1110757X03212055.

[43] A. Varga, *Balancing free square-root algorithm for computing singular perturbation approximations*, in Proceedings of the 30th IEEE Conference on Decision and Control, Brighton, UK, 1991, pp. 1062–1065, https://doi.org/10.1109/CDC.1991.261486.

[44] K. Willcox and J. Peraire, *Balanced model reduction via the proper orthogonal decomposition*, AIAA J., 40 (2002), pp. 2323–2330, https://doi.org/10.2514/2.1570.

[45] Q. Zhang, I. V. Gosea, and A. C. Antoulas, *Factorization of the Loewner Matrix Pencil and Its Consequences*, preprint, https://arxiv.org/abs/2103.09674, 2021.