SIAM J. SCI. COMPUT.
Vol. 44, No. 1, pp. A554–A582

DATA-DRIVEN BALANCING OF LINEAR DYNAMICAL SYSTEMS*

ION VICTOR GOSEA†, SERKAN GUGERCIN‡, AND CHRISTOPHER BEATTIE‡

Abstract. We present a novel reformulation of balanced truncation, a classical model reduction method. The principal innovation that we introduce comes through the use of system response data that has been either measured or computed, without reference to any prescribed realization of the original model. Data are represented by sampled values of the transfer function or the impulse response corresponding to the original model. We discuss parallels that our approach bears with the Loewner framework, another popular data-driven method. We illustrate our approach numerically in both continuous-time and discrete-time cases.

Key words. balanced truncation, numerical quadrature, data-driven modeling, transfer functions, impulse responses, Gramians

AMS subject classifications. 37M99, 41A20, 65D30, 65K99, 93A15, 93B15

DOI. 10.1137/21M1411081

1. Introduction.

Model reduction here refers to system-theoretic techniques used to create compactly represented *reduced models* that are capable of reproducing the input-output behavior of large-scale dynamical systems with high accuracy. This is accomplished by encoding fine scale dynamical features of the original systems efficiently into *reduced* dynamical systems, allowing them to closely mimic the input/output behavior of the original system for a wide range of input conditions, while having substantially lower order than the system whose behaviour they mimic. Such reduced models should be cheap to simulate and easy to manipulate and control. We refer the reader to [1, 2, 10, 47] for details on a variety of model reduction techniques.

Balanced truncation (BT) [38, 39] is one of the most successful and commonly deployed model reduction methods. For linear dynamical systems such as those that we consider in this paper, BT retains asymptotic stability of the original systems and provides a priori bounds for the model reduction error. BT effectively discards those states that are both difficult to reach and to observe, as quantified through the relative magnitude of the system's Hankel singular values. Typically, the loss of these states is expected to have little effect on the observed input-output dynamics of the system, and the resulting reduced model may replace the original system in simulations or analysis without much lost with regard to accuracy. Extensions of BT to nonlinear systems have emerged over the years, starting with [51] and continuing with more recent approaches tailored to particular classes of nonlinearities, such as bilinear systems in [7], quadratic-bilinear systems in [8], and switched systems in [24].

The major computational cost of BT is the need to solve large-scale Lyapunov equations in order to obtain the system Gramians, or better, their square-root factors. Many methods have been proposed to accomplish this efficiently; see, e.g., [11, 34,

*Submitted to the journal's Methods and Algorithms for Scientific Computing section April 9, 2021; accepted for publication (in revised form) September 22, 2021; published electronically February 28, 2022.

https://doi.org/10.1137/21M1411081

Funding: The work of the second and third authors was partially supported by the National Science Foundation under grants DMS-1923221 and DMS-1819110.

†Data-Driven System Reduction and Identification (DRI), Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, 39106, Germany (gosea@mpi-magdeburg.mpg.de).

‡Department of Mathematics and Division of Computational Modeling and Data Analytics, Virginia Tech, Blacksburg, VA 24061-0123 USA (gugercin@vt.edu, beattie@vt.edu).

35, 43, 50, 53, 61] and references therein. All such methods make central use of the system state-space representation and, in the context of BT, produce explicit state-space projections that ultimately are engaged to generate a reduced model. Thus, BT is "intrusive" to the extent that internal representations of the system dynamics play a central role in determining a final reduced model. This is to be contrasted with "nonintrusive" methods that are *data-driven*, requiring access only to system response and behavior data, e.g., transfer function or impulse response measurements.

In this paper, we develop a new, data-driven formulation for BT. We achieve this by recognizing that BT does not make independent use of the two system Gramians, which each depend on internal and hence generally inaccessible variables. BT rather makes fundamental use of their *product*, which preserves system invariants that do not depend on an underlying system realization. In the present work, we describe how this product of Gramians can be approximated directly through transfer function sampling to any accuracy desired. We explicitly derive reduced-order quantities using observed response data and, as a consequence, arrive at a novel, *nonintrusive* formulation of BT.

The rest of the paper is organized as follows: In section 2, we review the basic elements of BT and detail the usual steps involved for computing balanced reduced-order models (ROMs). We introduce our main result in section 3, a data-driven approach to BT, and discuss how one estimates key quantities through quadrature in the frequency domain that, in turn, involves quantities that are extracted from data. In section 4, we include refinements that take into account symmetries implicit in real-valued dynamics as well as extensions to MIMO, discrete-time, and infinite-dimensional systems. Even though our main focus is frequency-domain data, in section 5 we revisit the problem using time-domain data instead and make connections to earlier works. Conclusions are provided in section 6, while appendices elaborating on the quadrature rules considered and detailing the proof of Proposition 3.2 are at the end of the article. Numerical experiments and various illustrations are provided in sections 3, 4, and 5.

2. Balanced truncation. Consider the linear time-invariant (LTI) system

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

where the input mapping is given by $\mathbf{u} : \mathbb{R} \to \mathbb{R}^m$, the (generalized) state trajectory/variable is $\mathbf{x} : \mathbb{R} \to \mathbb{R}^n$, and the output mapping is $y : \mathbb{R} \to \mathbb{R}^p$. The system matrices are given by $\mathbf{E}$, $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{p\times n}$. Assume $\mathbf{E}$ is nonsingular and that the matrix pencil $(\mathbf{A}, \mathbf{E})$ is *asymptotically stable*, meaning that the eigenvalues of $(\mathbf{A}, \mathbf{E})$ (or equivalently of $\mathbf{E}^{-1}\mathbf{A}$) are located in the open left half-plane. The transfer function of the LTI system (2.1), defined as

$$\mathbf{H}(s) = \mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}, \tag{2.2}$$

is a $p \times m$ matrix-valued rational function in s.

The two fundamental quantities in BT are the reachability and observability Gramians. The reachability Gramian $\mathbf{P}$ provides a measure of how easily a state can be accessed from the zero state. In the time domain, $\mathbf{P}$ is given by

$$\mathbf{P} = \int_0^\infty e^{\mathbf{E}^{-1}\mathbf{A}t}\mathbf{E}^{-1}\mathbf{B}\mathbf{B}^T\mathbf{E}^{-T}e^{\mathbf{A}^T\mathbf{E}^{-T}t}dt. \tag{2.3}$$

Taking $\imath^2 = -1$, this can be represented equivalently in the frequency domain as

$$\mathbf{P} = \frac{1}{2\pi}\int_{-\infty}^{\infty} (\imath\zeta\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}\mathbf{B}^T(-\imath\zeta\mathbf{E}^T - \mathbf{A}^T)^{-1}d\zeta. \tag{2.4}$$

In a complementary way, the observability Gramian describes how easily a state can be observed and may be represented as $\mathbf{E}^T\mathbf{Q}\mathbf{E}$, where $\mathbf{Q}$ is defined as

$$\mathbf{Q} = \frac{1}{2\pi}\int_{-\infty}^{\infty} (-\imath\omega\mathbf{E}^T - \mathbf{A}^T)^{-1}\mathbf{C}^T\mathbf{C}(\imath\omega\mathbf{E} - \mathbf{A})^{-1}d\omega. \tag{2.5}$$

The Gramians $\mathbf{P}$ and $\mathbf{Q}$ satisfy the following generalized Lyapunov equations:

$$\mathbf{A}\mathbf{P}\mathbf{E}^T + \mathbf{E}\mathbf{P}\mathbf{A}^T + \mathbf{B}\mathbf{B}^T = \mathbf{0}, \tag{2.6}$$

$$\mathbf{A}^T\mathbf{Q}\mathbf{E} + \mathbf{E}^T\mathbf{Q}\mathbf{A} + \mathbf{C}^T\mathbf{C} = \mathbf{0}. \tag{2.7}$$

Although $\mathbf{E}^T\mathbf{Q}\mathbf{E}$ is the observability Gramian in this case, the Lyapunov equation (2.7) for $\mathbf{Q}$ enjoys a fundamental structural similarity to (2.6), and so similar computational algorithms are applicable (see, e.g., [11, 43, 57]). The solutions, $\mathbf{P}$ and $\mathbf{Q}$, to (2.6) and (2.7), respectively, are symmetric positive-semidefinite matrices, and one may compute square factors $\mathbf{L}, \mathbf{U} \in \mathbb{R}^{n\times n}$ (e.g., via a Cholesky factorization) so that

$$\mathbf{P} = \mathbf{U}\mathbf{U}^T \quad \text{and} \quad \mathbf{Q} = \mathbf{L}\mathbf{L}^T. \tag{2.8}$$

This provides the essential elements allowing us to define BT, which we summarize in Algorithm 1. The singular values of $\mathbf{L}^T\mathbf{E}\mathbf{U}$ (diagonal entries of $\mathsf{diag}(\mathbf{S}_1, \mathbf{S}_2)$ in (2.10)) are the *Hankel singular values* of the associated dynamical system, and these values are system invariants that are independent of realization. BT truncates the states that correspond to small Hankel singular values in $\mathbf{S}_2$. The rth-order reduced model resulting from BT in Algorithm 1 is asymptotically stable, and its transfer function is

$$\mathbf{H}_r(s) = \mathbf{C}_r(s\mathbf{E}_r - \mathbf{A}_r)^{-1}\mathbf{B}_r, \tag{2.9}$$

which satisfies $\|\mathbf{H} - \mathbf{H}_r\|_{\mathcal{H}_\infty} \leq 2\,\mathsf{trace}(\mathbf{S}_2)$, where $\|\cdot\|_{\mathcal{H}_\infty}$ denotes the $\mathcal{H}_\infty$-norm of a dynamical and where we assumed, for simplicity, that the Hankel singular values are distinct. For details, we refer the reader to [1].

3. A data-driven framework for balancing. The main innovation that we introduce centers on the observation that the key quantities $\mathbf{L}^T\mathbf{E}\mathbf{U}$, $\mathbf{L}^T\mathbf{A}\mathbf{U}$, $\mathbf{L}^T\mathbf{B}$, and $\mathbf{C}\mathbf{U}$, appearing in steps 1 and 3 of Algorithm 1, may be replaced by unitarily equivalent quantities that are well approximated directly from data. "Data" in our setting correspond to sampling of the transfer function $\mathbf{H}(s)$; we assume access to the values of $\mathbf{H}(\imath\hat{\omega})$ at a finite number of frequencies $\hat{\omega}$. (We consider time-domain sampling in section 5.) Our choice of frequency sampling, $\hat{\omega}$, will be associated with numerical quadratures used to approximate (2.4) and (2.5). Further details are provided below and in Appendix A; however, we note here that some quadrature rules also engage asymptotics of the integrand which in our setting may then require measuring the leading two Markov parameters of $\mathbf{H}(s)$, i.e., the leading two coefficients in the expansion of $\mathbf{H}(s)$ around $s = \infty$. For $\mathbf{H}(s) = \mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}$, $\mathbf{M}_0 = \mathbf{C}\mathbf{E}^{-1}\mathbf{B}$ is the zeroth Markov parameter (the first coefficient) and $\mathbf{M}_1 = \mathbf{C}\mathbf{E}^{-1}\mathbf{A}\mathbf{E}^{-1}\mathbf{B}$ is the first Markov parameter (the second expansion coefficient). These quantities can be measured nonintrusively, as explained in, e.g., [31], with practical applications ranging from vector network analyzers [32] to 3D laser vibrometers [62].

Algorithm 1 Balanced truncation (BT) (square-root implementation)

Input: LTI system described by matrices $\mathbf{E}$, $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, and $\mathbf{C} \in \mathbb{R}^{p\times n}$.
Output: BT reduced-system: $\mathbf{A}_r \in \mathbb{R}^{r\times r}, \mathbf{B}_r \in \mathbb{R}^{r\times m}, \mathbf{C}_r \in \mathbb{R}^{p\times r}$.

1: Compute the Lyapunov factors $\mathbf{U}, \mathbf{L} \in \mathbb{R}^{n\times n}$ from (2.8), and pick a truncation index, $1 \le r \le \min(\mathsf{rank}(\mathbf{U}), \mathsf{rank}(\mathbf{L}))$.
2: Compute the SVD of the matrix $\mathbf{L}^T\mathbf{E}\mathbf{U}$, partitioned as follows:

$$\mathbf{L}^T\mathbf{E}\mathbf{U} = \begin{bmatrix}\mathbf{Z}_1 & \mathbf{Z}_2\end{bmatrix} \begin{bmatrix}\mathbf{S}_1 & \\ & \mathbf{S}_2\end{bmatrix} \begin{bmatrix}\mathbf{Y}_1^T \\ \mathbf{Y}_2^T\end{bmatrix}, \tag{2.10}$$

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

$$\mathbf{W}_r = \mathbf{L}\mathbf{Z}_1\mathbf{S}_1^{-1/2} \text{ and } \mathbf{V}_r = \mathbf{U}\mathbf{Y}_1\mathbf{S}_1^{-1/2}. \tag{2.11}$$

4: The reduced-order system matrices are given by

$$\begin{aligned} \mathbf{E}_r &= \mathbf{W}_r^T\mathbf{E}\mathbf{V}_r = \mathbf{I}_r & \mathbf{A}_r &= \mathbf{W}_r^T\mathbf{A}\mathbf{V}_r = \mathbf{S}_1^{-1/2}\mathbf{Z}_1^T\left(\mathbf{L}^T\mathbf{A}\mathbf{U}\right)\mathbf{Y}_1\mathbf{S}_1^{-1/2}, \\ \mathbf{B}_r &= \mathbf{W}_r^T\mathbf{B} = \mathbf{S}_1^{-1/2}\mathbf{Z}_1^T\left(\mathbf{L}^T\mathbf{B}\right), & \mathbf{C}_r &= \mathbf{C}\mathbf{V}_r = \left(\mathbf{C}\mathbf{U}\right)\mathbf{Y}_1\mathbf{S}_1^{-1/2}. \end{aligned} \tag{2.12}$$

To simplify initial discussion, we consider first SISO systems, taking $\mathbf{B} = \mathbf{b}$ and $\mathbf{C} = \mathbf{c}^T$, for $\mathbf{b}, \mathbf{c} \in \mathbb{R}^n$ and the associated (scalar-valued) transfer function, $H(s)$.[1] Generalization to MIMO systems is straightforward and discussed in section 4.2.

3.1. Computing key quantities from data. For large-scale dynamical systems, solving the Lyapunov equations (2.6) and (2.7) for the Gramians $\mathbf{P}$ and $\mathbf{Q}$ is computationally demanding. Sophisticated strategies have been developed for solving these equations; see, e.g., [1, 5, 11, 34, 50, 53] and references therein. Using any of these techniques, one may find low-rank approximations to $\mathbf{U}$ and $\mathbf{L}$, and then Algorithm 1 uses instead these approximate low-rank factors. These approximate balancing approaches remain intrusive in the sense that they still will depend on explicit system realizations and model projection, i.e., $\mathbf{U}$ and $\mathbf{L}$ are approximated first, followed by an explicit projection step. Our approach will sidestep this through direct, nonintrusive, data-driven estimation of a quantity that is unitarily equivalent to $\mathbf{L}^T\mathbf{E}\mathbf{U}$.

We consider first a numerical quadrature rule that approximates the frequency integral defining $\mathbf{P}$ in (2.4), producing an approximate Gramian

$$\mathbf{P} \approx \widetilde{\mathbf{P}} = \sum_{j=1}^{\ell_p} \rho_j^2 (\imath\zeta_j\mathbf{E} - \mathbf{A})^{-1}\mathbf{b}\mathbf{b}^T(-\imath\zeta_j\mathbf{E}^T - \mathbf{A}^T)^{-1} + \rho_\infty^2\mathbf{E}^{-1}\mathbf{b}\mathbf{b}^T\mathbf{E}^{-T} \tag{3.1}$$

with ρ_j^2 and ζ_j denoting, respectively, quadrature weights and nodes. The last term involving ρ_∞ corresponds to a "node at infinity" and relates to the asymptotic decay of the integrand in (2.4) in the neighborhood of $\zeta = \infty$. The presence of this term in

[1] We distinguish matrix- and vector-valued quantities from scalar-valued quantities with boldface; hence $H(s)$ for SISO vs. $\mathbf{H}(s)$ for the MIMO counterpart, and similarly for Markov parameters, M_0 and M_1 for SISO vs. $\mathbf{M}_0$ and $\mathbf{M}_1$ for the MIMO counterparts.

(3.1) will depend on the choice of numerical quadrature; we investigate choices where this term is present and others where it is absent. In either case, we label the total number of quadrature points (including potentially a "node at infinity") as N_p. If the ρ_∞-term in (3.1) is absent, then $\ell_p = N_p$; otherwise, $\ell_p = N_p - 1$. Picking a distribution of quadrature nodes on the imaginary axis is natural in this context; we focus on two standard exemplars derived from the trapezoid rule and Clenshaw–Curtis quadrature and provide further details in Appendix A. Quadrature node distributions can be chosen on different integration contours as well (e.g., [60]), and the approach that we propose is equally applicable in these modified circumstances.

The balanced proper orthogonal decomposition (Balanced POD) approach of [61] uses this type of approximation for $\mathbf{P}$, employing the trapezoid rule as an underlying numerical quadrature approximation. A modified version of Balanced POD based on a time-domain quadrature applied to (2.3) was later proposed in [49] with an application to fluid dynamics. As is the case with other approximate balancing techniques, Balanced POD is *intrusive* in the sense that access to an explicit state-space realization is required.

Approximate balancing techniques typically approximate the Lyapunov factors $\mathbf{U}$ and $\mathbf{L}$ from (2.8), and to the extent that system response data may be utilized, its use seeks to reconstruct evolving states of the system and so tacitly requires a well-defined state-space realization. This contrasts significantly with the *nonintrusive* data-driven approach that we propose below. While there are important common themes that we adopt from Balanced POD, we do not require any knowledge of a state-space realization; we do not sample state trajectories even implicitly, and we have no need for direct approximation of the Lyapunov factors in (2.8).

Evidently, we may decompose the quadrature-based Gramian approximation as $\widetilde{\mathbf{P}} = \widetilde{\mathbf{U}}\widetilde{\mathbf{U}}^*$ with a square-root factor $\widetilde{\mathbf{U}} \in \mathbb{C}^{n\times N_p}$ defined as

$$\widetilde{\mathbf{U}} = \begin{bmatrix} \rho_1(\imath\zeta_1\mathbf{E} - \mathbf{A})^{-1}\mathbf{b} & \cdots & \rho_{\ell_p}(\imath\zeta_{\ell_p}\mathbf{E} - \mathbf{A})^{-1}\mathbf{b} & \rho_\infty\mathbf{E}^{-1}\mathbf{b} \end{bmatrix}. \tag{3.2}$$

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. A reasonable concern that may arise at this point is the potential loss of an underlying structural system symmetry, leaving us possibly with a complex-valued reduced model as an artifact of rounding errors. We address this concern in section 4.1 and show how the underlying system symmetry associated with real dynamics may be explicitly preserved and implemented in real arithmetic.

By applying a similar quadrature approximation to $\mathbf{Q}$, we obtain

$$\mathbf{Q} \approx \widetilde{\mathbf{Q}} = \sum_{k=1}^{\ell_q} \phi_k^2(-\imath\omega_k\mathbf{E}^T - \mathbf{A}^T)^{-1}\mathbf{c}\mathbf{c}^T(\imath\omega_k\mathbf{E} - \mathbf{A})^{-1} + \phi_\infty^2\mathbf{E}^{-T}\mathbf{c}\mathbf{c}^T\mathbf{E}^{-1}, \tag{3.3}$$

where ϕ_k^2 and ω_k denote, respectively, quadrature weights and nodes associated with approximating $\mathbf{Q}$ from (2.5). The corresponding square-root factor is $\widetilde{\mathbf{Q}} = \widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^*$, where

$$\widetilde{\mathbf{L}}^* = \begin{bmatrix} \phi_1\mathbf{c}^T(\imath\omega_1\mathbf{E} - \mathbf{A})^{-1} \\ \vdots \\ \phi_{\ell_q}\mathbf{c}^T(\imath\omega_{\ell_q}\mathbf{E} - \mathbf{A})^{-1} \\ \phi_\infty\mathbf{c}^T\mathbf{E}^{-1} \end{bmatrix} \in \mathbb{C}^{N_q\times n}. \tag{3.4}$$

Let us suppose for the moment that the quadratures described in (3.1) and (3.3) *exactly* recover the Gramians described in (2.4) and (2.5), respectively, so that

$$\mathbf{U}\mathbf{U}^T = \widetilde{\mathbf{U}}\widetilde{\mathbf{U}}^* \quad \text{and} \quad \mathbf{L}\mathbf{L}^T = \widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^*. \tag{3.5}$$

This, in turn, assures us of the existence of partial isometries, $\boldsymbol{\Psi}_p \in \mathbb{C}^{N_p \times n}$ and $\boldsymbol{\Psi}_q \in \mathbb{C}^{N_q \times n}$, such that

$$\widetilde{\mathbf{U}}^* = \boldsymbol{\Psi}_p \mathbf{U}^T \quad \text{and} \quad \widetilde{\mathbf{L}}^* = \boldsymbol{\Psi}_q \mathbf{L}^T.$$

Thus, $\widetilde{\mathbf{L}}^*\mathbf{E}\widetilde{\mathbf{U}} = \boldsymbol{\Psi}_q(\mathbf{L}^T\mathbf{E}\mathbf{U})\boldsymbol{\Psi}_p^*$, and so $\widetilde{\mathbf{L}}^*\mathbf{E}\widetilde{\mathbf{U}}$ is unitarily equivalent to $\mathbf{L}^T\mathbf{E}\mathbf{U}$, modulo its kernel and cokernel. More precisely, $\mathbf{L}^T\mathbf{E}\mathbf{U}$ is isomorphic to a linear transformation induced by $\widetilde{\mathbf{L}}^*\mathbf{E}\widetilde{\mathbf{U}}$ viewed as a mapping of the quotient space, $\mathbb{C}^{N_p}/\mathsf{Ker}(\widetilde{\mathbf{U}})$, onto $\mathsf{Ran}(\widetilde{\mathbf{L}}^*\mathbf{E}\widetilde{\mathbf{U}})$. This is a significant observation since the singular values of $\mathbf{L}^T\mathbf{E}\mathbf{U}$ are the *Hankel singular values* of the system which are system invariants (i.e., independent of system realization) and play a fundamental role in BT (see step 2 of Algorithm 1). Although $\mathbf{L}^T\mathbf{E}\mathbf{U}$ itself is evidently tied closely to knowledge of a state-space realization, the unitarily equivalent matrix, $\widetilde{\mathbf{L}}^*\mathbf{E}\widetilde{\mathbf{U}}$, will provide essentially equivalent information while being directly derivable from data, as we will see.

PROPOSITION 3.1. *Let* $\widetilde{\mathbf{U}}$ *and* $\widetilde{\mathbf{L}}$ *be as defined in* (3.2) *and* (3.4). *Define the matrix* $\widetilde{\mathbb{L}} = \widetilde{\mathbf{L}}^*\mathbf{E}\widetilde{\mathbf{U}} \in \mathbb{C}^{N_q \times N_p}$. *Then, for* $1 \le k \le N_q$ *and* $1 \le j \le N_p$,

$$\widetilde{\mathbb{L}}_{k,j} = \begin{cases} -\phi_k\rho_j \dfrac{H(\imath\omega_k) - H(\imath\zeta_j)}{\imath\omega_k - \imath\zeta_j} \text{ for } 1 \le k \le \ell_q,\ 1 \le j \le \ell_p, \\ \phi_k\rho_\infty H(\imath\omega_k) \text{ for } 1 \le k \le \ell_q = N_q,\ j = N_p = \ell_p + 1, \\ \phi_\infty\rho_j H(\imath\zeta_j) \text{ for } k = N_q = \ell_q + 1,\ 1 \le j \le \ell_p = N_p, \\ \phi_\infty\rho_\infty M_0 \text{ for } k = N_q = \ell_q + 1,\ j = N_p = \ell_p + 1. \end{cases} \tag{3.6}$$

Proof. Let $\mathbf{e}_k$ denote the kth canonical vector (of conforming length). From the definitions of $\widetilde{\mathbf{L}}$ and $\widetilde{\mathbf{U}}$ and for indices $1 \le k \le \ell_q$ and $1 \le j \le \ell_p$, it follows that

$$\begin{aligned} \widetilde{\mathbb{L}}_{k,j} &= \mathbf{e}_k^T\widetilde{\mathbb{L}}\mathbf{e}_j = (\mathbf{e}_k^T\widetilde{\mathbf{L}}^*)\mathbf{E}(\widetilde{\mathbf{U}}\mathbf{e}_j) = \phi_k\rho_j\mathbf{c}^T(\imath\omega_k\mathbf{E} - \mathbf{A})^{-1}\,\mathbf{E}\,(\imath\zeta_j\mathbf{E} - \mathbf{A})^{-1}\mathbf{b} \\ &= \frac{\phi_k\rho_j}{\imath\omega_k - \imath\zeta_j}\mathbf{c}^T(\imath\omega_k\mathbf{E} - \mathbf{A})^{-1}\left[(\imath\omega_k\mathbf{E} - \mathbf{A}) - (\imath\zeta_j\mathbf{E} - \mathbf{A})\right](\imath\zeta_j\mathbf{E} - \mathbf{A})^{-1}\mathbf{b} \\ &= -\frac{\phi_k\rho_j}{\imath\omega_k - \imath\zeta_j}\mathbf{c}^T\left[(\imath\omega_k\mathbf{E} - \mathbf{A})^{-1} - (\imath\zeta_j\mathbf{E} - \mathbf{A})^{-1}\right]\mathbf{b} = -\phi_k\rho_j\frac{H(\imath\omega_k) - H(\imath\zeta_j)}{\imath\omega_k - \imath\zeta_j}. \end{aligned}$$

Next, if $1 \le k \le \ell_q = N_q$ and $j = N_p = \ell_p + 1$, we can write that

$$\begin{aligned} \widetilde{\mathbb{L}}_{k,j} &= \mathbf{e}_k^T\widetilde{\mathbb{L}}\mathbf{e}_j = (\mathbf{e}_k^T\widetilde{\mathbf{L}}^*)\mathbf{E}(\widetilde{\mathbf{U}}\mathbf{e}_j) = \phi_k\mathbf{c}^T(\imath\omega_k\mathbf{E} - \mathbf{A})^{-1}\,\mathbf{E}\,\rho_\infty\mathbf{E}^{-1}\mathbf{b} \\ &= \phi_k\rho_\infty\mathbf{c}^T(\imath\omega_k\mathbf{E} - \mathbf{A})^{-1}\mathbf{b} = \phi_k\rho_\infty H(\imath\omega_k). \end{aligned}$$

Similarly, we can derive $\widetilde{\mathbb{L}}_{k,j}$ for the case $k = N_q = \ell_q + 1$ and $1 \le j \le \ell_p = N_p$. Finally, for $k = N_q = \ell_q + 1,\ j = N_p = \ell_p + 1$, it follows that

$$\widetilde{\mathbb{L}}_{k,j} = (\mathbf{e}_k^T\widetilde{\mathbf{L}}^*)\mathbf{E}(\widetilde{\mathbf{U}}\mathbf{e}_j) = \phi_\infty\mathbf{c}^T\mathbf{E}^{-1}\,\mathbf{E}\,\rho_\infty\mathbf{E}^{-1}\mathbf{b} = \phi_\infty\rho_\infty\mathbf{c}^T\mathbf{E}^{-1}\mathbf{b} = \phi_\infty\rho_\infty M_0,$$

which completes the proof. ∎

The observation that the quadrature-based approximate quantity $\widetilde{\mathbb{L}} = \widetilde{\mathbf{L}}^* \mathbf{E} \widetilde{\mathbf{U}}$ may be obtained solely from transfer function samples is consistent with it being a system invariant, producing approximations to Hankel singular values that are also system invariants and not dependent on any specific state-space realization. Note that $\widetilde{\mathbb{L}}$ has at most rank n, i.e., the McMillan degree of the underlying system.

Of course, it will not be the case that the quadratures described in (3.1) and (3.3) are exact, so before we proceed we seek some assurance that the error induced by the quadrature approximation can be controlled. We have the following result.

PROPOSITION 3.2. *Suppose the quadratures in* (3.1) *and* (3.3) *produce approximations* $\widetilde{\mathbf{P}}$ *and* $\widetilde{\mathbf{Q}}$ *to* $\mathbf{P}$ *and* $\mathbf{Q}$*, respectively, that satisfy* $\|\mathbf{Q} - \widetilde{\mathbf{Q}}\|_F \leq \frac{\delta}{1+\delta} \sigma_{\mathsf{min}}(\mathbf{Q})$ *and* $\|\mathbf{P} - \widetilde{\mathbf{P}}\|_F \leq \frac{\delta}{1+\delta} \sigma_{\mathsf{min}}(\mathbf{P})$ *for some* $\delta \in (0,1)$*, where* $\sigma_{\mathsf{min}}(\,\cdot\,)$ *denotes the smallest singular value. Then there exist isometries* $\mathbf{\Psi}_p \in \mathbb{C}^{N_p \times n}$ *and* $\mathbf{\Psi}_q \in \mathbb{C}^{N_q \times n}$ *such that*

$$\|\widetilde{\mathbf{L}}^* \mathbf{E} \widetilde{\mathbf{U}} - \mathbf{\Psi}_q (\mathbf{L}^T \mathbf{E} \mathbf{U}) \mathbf{\Psi}_p^*\|_F \leq 2\ \delta\ \|\mathbf{E}\|_2\, \|\mathbf{L}\|_2\, \|\mathbf{U}\|_2. \tag{3.7}$$

Let $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ *denote singular values of* $\mathbf{L}^T \mathbf{E} \mathbf{U}$ *(i.e., Hankel singular values of* (2.1)*), and let* $\widetilde{\sigma}_1 \geq \widetilde{\sigma}_2 \geq \cdots \geq \widetilde{\sigma}_n$ *denote the singular values of* $\widetilde{\mathbf{L}}^* \mathbf{E} \widetilde{\mathbf{U}}$*. Then,*

$$\left(\sum_{k=1}^{n} (\sigma_k - \widetilde{\sigma}_k)^2 \right)^{\frac{1}{2}} \leq 2\ \delta\ \|\mathbf{E}\|_2\, \|\mathbf{L}\|_2\, \|\mathbf{U}\|_2.$$

The proof of this proposition may be found in Appendix B.

This leads us to consider a *quadrature-based* approximation to BT that uses $\widetilde{\mathbf{U}}$ and $\widetilde{\mathbf{L}}$ in lieu of $\mathbf{U}$ and $\mathbf{L}$ in step 2 of Algorithm 1, replacing the SVD of $\mathbf{L}^T \mathbf{E} \mathbf{U}$ with the SVD of the (nearly) unitarily equivalent matrix, $\widetilde{\mathbb{L}} = \widetilde{\mathbf{L}}^* \mathbf{E} \widetilde{\mathbf{U}}$, that can be computed *nonintrusively*, generated essentially from transfer function measurements.

Recall the singular values of $\mathbf{L}^T \mathbf{E} \mathbf{U}$ (the Hankel singular values) are independent of system realization. Our nonintrusive, quadrature-based approach provides a new way of accessing the Hankel singular values without recourse to a system realization.

3.1.1. Numerical example. We use the heat model [40] (referred to as [heat] here), describing thermal response of a thin rod—the system is SISO with dimension $n = 200$.

In Figure 3.1, we depict the true Hankel singular values together with the approximate ones obtained via the nonintrusive quadrature-based approach as described above. We use two representative quadrature rules described in Appendix A, labeled

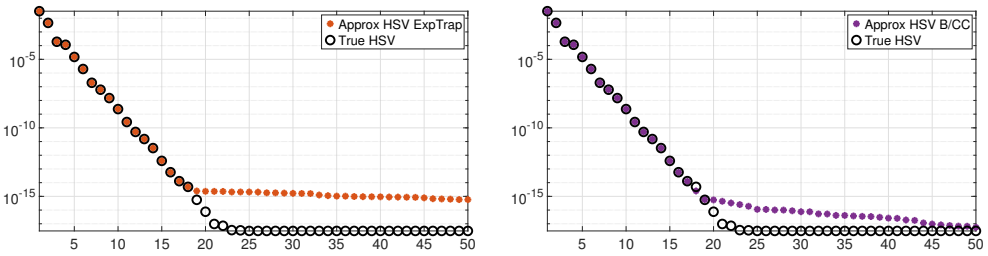

FIG. 3.1. *The computed Hankel singular values (HSV) for the NICONET* [heat] *model using a trapezoid quadrature with* 120 *nodes in the interval* $[10^{-3}, 10^{3}]i$ *(left) and using* 120 *nodes with* $L = 4$ *and* $L = 3$ *in* [B/CC] *for* $\widetilde{\mathbf{U}}$ *and* $\widetilde{\mathbf{L}}$*, respectively (right).*

as [ExpTrap] and [B/CC]. As the figures illustrate, the quadrature-based estimates for the Hankel singular values approximate the true values accurately.

3.2. Data-driven BT. Although BT yields reduced models in a manner that reflects system invariants and in that sense is independent of system realizations, BT typically does require explicit access to a state-space representation and thus, in a practical sense, remains tied to system realizations. In particular, note that in step 3 of Algorithm 1, model reduction bases $\mathbf{W}_r$ and $\mathbf{V}_r$ are constructed and then used to project state-space quantities. Through our (implicit) use of quadrature approximations for the reachability and observability Gramians, we are able to avoid any explicit reference to model reduction bases or to specific system realizations and to effect BT in a coordinate system determined solely by input-output data.

Notice that in step 4 of Algorithm 1, we have $\mathbf{A}_r = \mathbf{S}_1^{-1/2}\mathbf{Z}_1^T\left(\mathbf{L}^T\mathbf{A}\mathbf{U}\right)\mathbf{Y}_1\mathbf{S}_1^{-1/2}$, and so the construction of $\mathbf{A}_r$ makes reference to both $\mathbf{L}^T\mathbf{A}\mathbf{U}$ and the SVD of $\mathbf{L}^T\mathbf{E}\mathbf{U}$, reflecting equivalent basis transformations on both $\mathbf{A}$ and $\mathbf{E}$. Proposition 3.1 showed how we can replace $\mathbf{L}^T\mathbf{E}\mathbf{U}$ with the (nearly) unitarily equivalent matrix, $\widetilde{\mathbb{L}} = \widetilde{\mathbf{L}}^*\mathbf{E}\widetilde{\mathbf{U}}$ (which could then be computed directly from data). So, we might expect that we could construct a realization equivalent to $\mathbf{A}_r$ by replicating on $\mathbf{A}$ an equivalent change of basis that had been visited upon $\mathbf{E}$ by $\widetilde{\mathbb{L}} = \widetilde{\mathbf{L}}^*\mathbf{E}\widetilde{\mathbf{U}}$. Pleasantly, we will find this to be the case by examining $\widetilde{\mathbb{M}} = \widetilde{\mathbf{L}}^*\mathbf{A}\widetilde{\mathbf{U}}$. Similar observations hold for $\mathbf{b}_r$ and $\mathbf{c}_r$.

PROPOSITION 3.3. *Let $\widetilde{\mathbf{U}}$ and $\widetilde{\mathbf{L}}$ be as defined in* (3.2) *and* (3.4). *Define the matrix $\widetilde{\mathbb{M}} = \widetilde{\mathbf{L}}^*\mathbf{A}\widetilde{\mathbf{U}} \in \mathbb{C}^{N_q\times N_p}$. Then, the (k,j) entry of $\widetilde{\mathbb{M}}$, for $1 \le k \le N_q$ and $1 \le j \le N_p$, is given by*

$$\widetilde{\mathbb{M}}_{k,j} = \begin{cases} -\phi_k\rho_j\dfrac{\imath\omega_k H(\imath\omega_k) - \imath\zeta_j H(\imath\zeta_j)}{\imath\omega_k - \imath\zeta_j} \text{ for } 1 \le k \le \ell_q,\ 1 \le j \le \ell_p, \\ \phi_k\rho_\infty\left(\imath\omega_k H(\imath\omega_k) - M_0\right) \text{ for } 1 \le k \le \ell_q = N_q,\ j = N_p = \ell_p + 1, \\ \phi_\infty\rho_j\left(\imath\zeta_j H(\imath\zeta_j) - M_0\right) \text{ for } k = N_q = \ell_q + 1,\ 1 \le j \le \ell_p = N_p, \\ \phi_\infty\rho_\infty M_1 \text{ for } k = N_q = \ell_q + 1,\ j = N_p = \ell_p + 1. \end{cases} \tag{3.8}$$

Likewise, defining $\widetilde{\mathbb{h}} = \widetilde{\mathbf{L}}^*\mathbf{b} \in \mathbb{C}^{N_q\times 1}$ and $\widetilde{\mathbf{g}}^T = \mathbf{c}^T\widetilde{\mathbf{U}} \in \mathbb{C}^{1\times N_p}$, we find

$$\widetilde{\mathbb{h}}_k = \begin{cases} \phi_k H(\imath\omega_k) \text{ for } 1 \le k \le \ell_q, \\ \phi_\infty M_0 \text{ for } k = N_q = \ell_q + 1, \end{cases} \tag{3.9}$$

$$\widetilde{\mathbf{g}}_j = \begin{cases} \rho_j H(\imath\zeta_j) \text{ for } 1 \le j \le N_p, \\ \rho_\infty M_0 \text{ for } j = N_p = \ell_p + 1. \end{cases} \tag{3.10}$$

Proof. The proof is similar to that of Proposition 3.1. Notice first that for any square matrix, $\mathbf{T}$, and any ς, $\mu \in \mathbb{C}$ that are not eigenvalues of $\mathbf{T}$, one may verify

$$\varsigma(\varsigma\mathbf{I} - \mathbf{T})^{-1} - \mu(\mu\mathbf{I} - \mathbf{T})^{-1} = -(\varsigma - \mu)(\varsigma\mathbf{I} - \mathbf{T})^{-1}\mathbf{T}(\mu\mathbf{I} - \mathbf{T})^{-1}.$$

By setting $\mathbf{T} = \mathbf{A}\mathbf{E}^{-1}$ and simplifying, one finds

$$\varsigma(\varsigma\mathbf{E} - \mathbf{A})^{-1} - \mu(\mu\mathbf{E} - \mathbf{A})^{-1} = -(\varsigma - \mu)(\varsigma\mathbf{E} - \mathbf{A})^{-1}\mathbf{A}(\mu\mathbf{E} - \mathbf{A})^{-1}.$$

Now, using the definitions of $\widetilde{\mathbf{L}}$ and $\widetilde{\mathbf{U}}$,

$$\begin{aligned}\widetilde{\mathbb{M}}_{k,j} &= \mathbf{e}_k^T\widetilde{\mathbb{M}}\mathbf{e}_j = (\mathbf{e}_k^T\widetilde{\mathbf{L}}^*)\mathbf{A}(\widetilde{\mathbf{U}}\mathbf{e}_j) = \phi_k\rho_j\mathbf{c}^T(\imath\omega_k\mathbf{E}-\mathbf{A})^{-1}\mathbf{A}(\imath\zeta_j\mathbf{E}-\mathbf{A})^{-1}\mathbf{b}\\ &= -\frac{\phi_k\rho_j}{\imath\omega_k-\imath\zeta_j}\mathbf{c}^T\left[\imath\omega_k(\imath\omega_k\mathbf{E}-\mathbf{A})^{-1}-\imath\zeta_j(\imath\zeta_j\mathbf{E}-\mathbf{A})^{-1}\right]\mathbf{b}\\ &= -\phi_k\rho_j\frac{\imath\omega_k H(\imath\omega_k)-\imath\zeta_j H(\imath\zeta_j)}{\imath\omega_k-\imath\zeta_j}.\end{aligned}$$

Next, if $1\le k\le \ell_q = N_q$ and $j = N_p = \ell_p+1$, we can write that

$$\begin{aligned}\widetilde{\mathbb{M}}_{k,j} &= \mathbf{e}_k^T\widetilde{\mathbb{M}}\mathbf{e}_j = (\mathbf{e}_k^T\widetilde{\mathbf{L}}^*)\mathbf{A}(\widetilde{\mathbf{U}}\mathbf{e}_j) = \phi_k\mathbf{c}^T(\imath\omega_k\mathbf{E}-\mathbf{A})^{-1}\mathbf{A}\rho_\infty\mathbf{E}^{-1}\mathbf{b}\\ &= \phi_k\rho_\infty\mathbf{c}^T\left[\imath\omega_k(i\omega_k\mathbf{E}-\mathbf{A})^{-1}-\mathbf{E}^{-1}\right]\mathbf{b} = \phi_k\rho_\infty\left(\imath\omega_k H(\imath\omega_k)-M_0\right).\end{aligned}$$

Similarly, we can find the exact derivation of $\widetilde{\mathbb{L}}_{k,j}$ in (3.8) for the case $k = N_q = \ell_q+1,\ 1\le j\le \ell_p = N_p$. Next, for $k = N_q = \ell_q+1,\ j = N_p = \ell_p+1$, it follows that

$$\begin{aligned}\widetilde{\mathbb{M}}_{k,j} &= \mathbf{e}_k^T\widetilde{\mathbb{M}}\mathbf{e}_j = (\mathbf{e}_k^T\widetilde{\mathbf{L}}^*)\mathbf{A}(\widetilde{\mathbf{U}}\mathbf{e}_j) = \phi_\infty\mathbf{c}^T\mathbf{E}^{-1}\,\mathbf{A}\,\rho_\infty\mathbf{E}^{-1}\mathbf{b}\\ &= \phi_\infty\rho_\infty\mathbf{c}^T\mathbf{E}^{-1}\mathbf{A}\mathbf{E}^{-1}\mathbf{b} = \phi_\infty\rho_\infty M_1.\end{aligned}$$

Finally, one can write

$$\widetilde{\mathbb{h}}_k = \mathbf{e}_k^T\widetilde{\mathbb{h}} = (\mathbf{e}_k^T\widetilde{\mathbf{L}})\mathbf{b} = \begin{cases}\phi_k\mathbf{c}^T(\imath\omega_k\mathbf{E}-\mathbf{A})^{-1}\mathbf{b} = \phi_k H(\imath\omega_k) \text{ for } 1\le k\le\ell_q,\\ \phi_\infty\mathbf{c}^T\mathbf{E}^{-1}\mathbf{b} = \phi_\infty M_0 \text{ for } k = N_q = \ell_q+1\end{cases}$$

and

$$\widetilde{\mathbb{g}}_j = \widetilde{\mathbb{g}}^T\mathbf{e}_j = \mathbf{c}^T(\widetilde{\mathbf{U}}\mathbf{e}_j) = \begin{cases}\rho_j\mathbf{c}^T(\imath\zeta_j\mathbf{E}-\mathbf{A})^{-1}\mathbf{b} = \rho_j H(\imath\zeta_j) \text{ for } 1\le j\le N_p,\\ \rho_\infty\mathbf{c}^T\mathbf{E}^{-1}\mathbf{b} = \rho_\infty M_0 \text{ for } j = N_p = \ell_p+1,\end{cases}$$

completing the proof. ∎

We have been able to replace $\mathbf{L}^T\mathbf{E}\mathbf{U}$, $\mathbf{L}^T\mathbf{A}\mathbf{U}$, $\mathbf{L}^T\mathbf{b}$, and $\mathbf{c}^T\mathbf{U}$ in Algorithm 1 with equivalent quantities that are derivable directly from data (i.e., from transfer function samples). We assemble this together in Algorithm 2, yielding a computationally feasible strategy for quadrature-based balanced truncation (QuadBT), requiring only transfer function sampling (i.e., no access to internal properties) and fully capable of recovering models equivalent to BT reduced models to any accuracy desired.

Note the contrast in the construction of $\mathbf{A}_r$ in (2.12) with that of $\widetilde{\mathbf{A}}_r$ in (3.12). The former computes the inner term $\mathbf{L}^T\mathbf{A}\mathbf{U}$ and requires access to internal quantities characterizing the dynamics; the latter approximates this term through $\widetilde{\mathbb{M}}$ and may be constructed directly from transfer function samples. Indeed, the quantities $\widetilde{\mathbf{S}}_1^{-1/2}\widetilde{\mathbf{Z}}_1^*$ and $\widetilde{\mathbf{Y}}_1\widetilde{\mathbf{S}}_1^{-1/2}$ are available from the SVD of the data matrix $\widetilde{\mathbb{L}}$ in (3.6).

Remark 3.1. In the analysis above, we have assumed that the quadrature nodes $\{\zeta_j\}$ for $\widetilde{\mathbf{P}}$ and $\{\omega_k\}$ for $\widetilde{\mathbf{Q}}$ are distinct from each other. However, it may be more convenient to choose the same quadrature nodes (and weights) for both $\widetilde{\mathbf{P}}$ and $\widetilde{\mathbf{Q}}$. This leads to only minor changes to Algorithm 2. The vectors $\widetilde{\mathbb{h}}$ in (3.9) and $\widetilde{\mathbb{g}}$ in (3.10) stay unchanged. The principal change occurs in the diagonal entries of $\widetilde{\mathbb{L}}$ and $\widetilde{\mathbb{M}}$: if $\omega_k = \zeta_k$, then the (k,k) entry of $\widetilde{\mathbb{L}}$ in (3.6) becomes

$$\widetilde{\mathbb{L}}_{k,k} = -\phi_k\rho_k H'(\imath\omega_k),$$

Algorithm 2 Quadrature-based (data-driven) balanced truncation (QuadBT)

Input: LTI system described through a transfer function evaluation map, $H(s)$;
quadrature nodes, ζ_j, and weights, ρ_j, for $j = 1, 2, \ldots, N_p$;
quadrature nodes, ω_k, and weights, ϕ_k, for $k = 1, 2, \ldots, N_q$;
and a truncation index, $1 \leq r \leq \min(N_p, N_q)$.

Output: QuadBT reduced-system: $\widetilde{\mathbf{A}}_r \in \mathbb{R}^{r\times r}$, $\widetilde{\mathbf{b}}_r, \widetilde{\mathbf{c}}_r \in \mathbb{R}^r$.

1: Sample transfer function values, $\{H(\imath\zeta_j)\}_{j=1}^{N_p}$ and $\{H(\imath\omega_k)\}_{k=1}^{N_q}$. Using the samples and quadrature weights, $\{\rho_j\}$ and $\{\phi_k\}$, construct $\widetilde{\mathbb{L}} \in \mathbb{C}^{N_q\times N_p}$, $\widetilde{\mathbb{M}} \in \mathbb{C}^{N_q\times N_p}$, $\widetilde{\mathbb{h}}$, and $\widetilde{\mathbb{g}}$ as in (3.6), (3.8), (3.9), and (3.10), respectively,

2: Compute the SVD of $\widetilde{\mathbb{L}}$:

$$\widetilde{\mathbb{L}} = \begin{bmatrix}\widetilde{\mathbf{Z}}_1 & \widetilde{\mathbf{Z}}_2\end{bmatrix} \begin{bmatrix}\widetilde{\mathbf{S}}_1 & \\ & \widetilde{\mathbf{S}}_2\end{bmatrix} \begin{bmatrix}\widetilde{\mathbf{Y}}_1^* \\ \widetilde{\mathbf{Y}}_2^*\end{bmatrix}, \tag{3.11}$$

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

3: Construct the reduced-order matrices:

$$\begin{gathered}\widetilde{\mathbf{E}}_r = \widetilde{\mathbf{S}}_1^{-1/2}\widetilde{\mathbf{Z}}_1^*\, \widetilde{\mathbb{L}}\, \widetilde{\mathbf{Y}}_1\widetilde{\mathbf{S}}_1^{-1/2} = \mathbf{I}_r, \qquad \widetilde{\mathbf{A}}_r = \widetilde{\mathbf{S}}_1^{-1/2}\widetilde{\mathbf{Z}}_1^*\, \widetilde{\mathbb{M}}\, \widetilde{\mathbf{Y}}_1\widetilde{\mathbf{S}}_1^{-1/2}, \\ \widetilde{\mathbf{b}}_r = \widetilde{\mathbf{S}}_1^{-1/2}\widetilde{\mathbf{Z}}_1^*\widetilde{\mathbb{h}}, \qquad \text{and} \qquad \widetilde{\mathbf{c}}_r = \widetilde{\mathbb{g}}^T\widetilde{\mathbf{Y}}_1\widetilde{\mathbf{S}}_1^{-1/2}.\end{gathered} \tag{3.12}$$

where H' denotes the derivative of H. The (k,k) entry of $\widetilde{\mathbb{M}}$ in (3.8) is replaced by

$$\widetilde{\mathbb{M}}_{k,k} = -\phi_k\rho_k\left(\omega_k H'(\imath\omega_k) + H(\imath\omega_k)\right).$$

The remaining entries in $\widetilde{\mathbb{L}}$ and $\widetilde{\mathbb{M}}$ are unchanged.

3.2.1. Numerical examples: The SISO case. In this section we will test the approximation capability of our newly proposed method QuadBT and compare it with that of classical BT. We employ the following abbreviations:

1. [BT-classic]: The classical BT approach (Algorithm 1).
2. [ExpTrap-N]: QuadBT (Algorithm 2) via the exponential trapezoid rule with N points (without derivative samples); more details may be found in Appendix A.
3. [B/CC-N]: QuadBT (Algorithm 2) via the Boyd/Clenshaw–Curtis rule with N points (without derivative samples); more details may be found in Appendix A.

We first consider the [heat] model from section 3.1.1. We use first 60 and then 120 logarithmically spaced points in the interval $[10^{-3}, 10^3]\imath$ as nodes for [ExpTrap] (the left and right nodes are different). In [B/CC], choose $L = 4$ and $L = 3$ for $\widetilde{\mathbf{U}}$ and $\widetilde{\mathbf{L}}$, respectively, using again first 60 and then 120 nodes. We vary the reduction order from $r = 2$ to $r = 14$ in increments of two and collect both the $\mathcal{H}_\infty$ and the $\mathcal{H}_2$ norms of the error systems $\mathbf{H}(s) - \mathbf{H}_r(s)$ corresponding to [BT-classic] and both quadrature schemes. The results are presented in Figure 3.2, illustrating that both QuadBT models (with [ExpTrap-N] and [B/CC-N]) nearly replicate the performance of [BT-classic].

As a second example, we use the International Space Station benchmark [40] (referred to as [iss1r]) modeling the 1r component of the International Space Station—the system dimension is $n = 270$; it has three inputs and three outputs. To enforce a

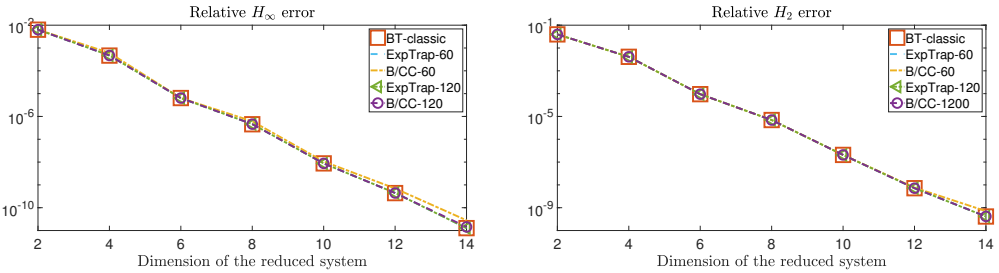

FIG. 3.2. *The relative $\mathcal{H}_\infty$ (left) and $\mathcal{H}_2$ (right) error approximation for different quadratures and nodes applied to the NICONET* [heat] *model.*

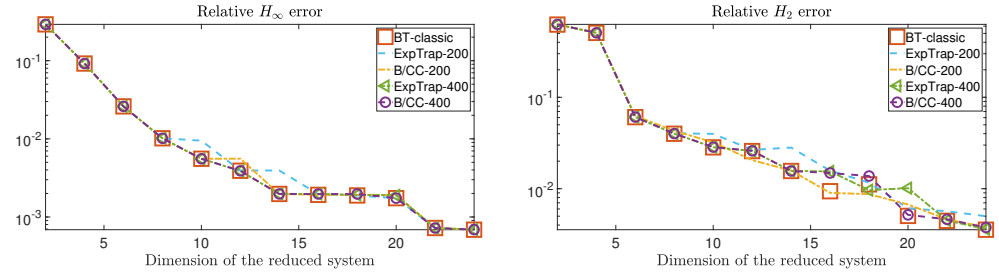

FIG. 3.3. *The relative $\mathcal{H}_\infty$ (left) and $\mathcal{H}_2$ (right) error approximation for different quadratures and nodes applied to the NICONET* [iss1r] *model.*

SISO system, we restrict our attention to the first input and first output. We choose both 200 and 400 logarithmically spaced points in the interval $[10^{-1}, 10^2]i$ as nodes for the [ExpTrap] scheme (the left and right nodes are different). In [B/CC], we choose $L = 10$ and $L = 9$ for $\widetilde{\mathbf{U}}$ and $\widetilde{\mathbf{L}}$, respectively, and use both 200 and 400 nodes. We then vary the reduction order in the interval $[2, 24]$ (in increments of two) and collect both the $\mathcal{H}_\infty$ and the $\mathcal{H}_2$ norms of the error systems corresponding to all three reduced models. The results are presented in Figure 3.3. As in the [heat] case, these figures show that transfer-function-based QuadBT very closely mimics (in many cases exactly replicates) the $\mathcal{H}_\infty$ and $\mathcal{H}_2$ performance of the projection-based BT. In this example, [B/CC] quadrature outperforms [ExpTrap].

For our last experiment, we purposefully reduce the accuracy of the underlying quadrature by using only a third as many nodes as in the previous experiment while keeping the same nodal configuration as before (i.e., 20 and 40 nodes, respectively, that are logarithmically spaced in $[10^{-3}, 10^3]i$ for [ExpTrap] and the same L-parameter choices for [B/CC]). This leads to a slightly less accurate approximation of Hankel singular values, as can be observed in the left part of Figure 3.4. Likewise, the corresponding reduced models are slightly less accurate than those computed in the first experiment; see the right part of Figure 3.4. This situation is somewhat analogous to what is observed with approximate BT methods that approach the solution of the Lyapunov equations (2.6)–(2.7) through explicit low-rank approximation of the related square-root factors, (2.8). Such approaches, exemplified by low-rank ADI methods (see, e.g., [11, 22, 35, 44]) produce errors in the low-rank factors that jointly contribute to the error in $\mathbf{L}^T\mathbf{E}\mathbf{U}$ in the first step of BT, in a way that is similar to what we see when coarsening our quadrature rules. Although the behavior of these projection-based low-rank ADI approaches is still not fully understood, there is overwhelming

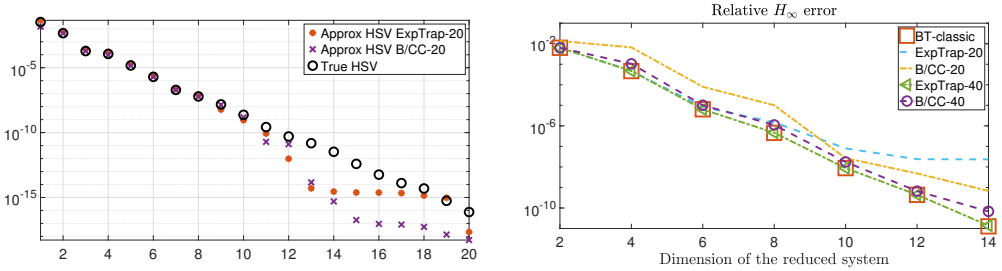

FIG. 3.4. *The computed Hankel singular values (HSV) for the NICONET* [heat] *model using using* 20 *nodes with* $L = 4$ *and* $L = 3$ *in* [B/CC] *for* $\widetilde{\mathbf{U}}$ *and* $\widetilde{\mathbf{L}}$*, respectively, and* 20 *nodes for [ExpTrap] (left) and the relative* $\mathcal{H}_\infty$ *(right) error approximation for different quadratures and nodes applied to the NICONET* [heat] *model.*

numerical evidence that projection-based approximate BT produced by such methods works very well in practice. We anticipate that this favorable situation will be seen in our current data-driven BT formulation, as well.

3.3. Connections to approximate BT methods. Due to its compelling theoretical properties, an immense amount of literature has developed around effective methods for approximating BT for large-scale dynamical systems, often focusing on efficient handling of the main computational bottleneck in BT, namely, the computation of Gramians, which requires the solution of two large-scale Lyapunov equations (2.6)–(2.7). We list a small selection of these works here [11, 34, 35, 44, 53, 57]; further references and detail can be found within these resources.

A common approach for approximate BT involves constructing low-rank approximate factors to $\mathbf{P}$ and $\mathbf{Q}$, i.e., $\mathbf{P} \approx \widehat{\mathbf{U}}\widehat{\mathbf{U}}^T$ and $\mathbf{Q} \approx \widehat{\mathbf{L}}\widehat{\mathbf{L}}^T$, with $\mathsf{rank}(\widehat{\mathbf{U}})$ and $\mathsf{rank}(\widehat{\mathbf{L}})$ much smaller than n. Then we use $\widehat{\mathbf{U}}$ and $\widehat{\mathbf{L}}$ in Algorithm 1 in place of the true square-root factors $\mathbf{U}$ and $\mathbf{L}$. Unfortunately, regardless of how $\widehat{\mathbf{U}}$ and $\widehat{\mathbf{L}}$ are computed, this construction requires access to internal dynamics and state-space quantities. The reduced model is then obtained by an explicit projection as in (2.12). This is fundamentally different from what we develop in this work; we do not compute $\widehat{\mathbf{U}}$ or $\widehat{\mathbf{L}}$, and we do not require access to state-space quantities. We only assume access to transfer function samples. Despite these major contrasts, there are striking similarities that motivate the underlying approximation theory.

The quadrature approximation (3.1) to $\mathbf{P}$ has been a major force behind the frequency-domain version of Balanced POD [61]. Indeed, the Balanced POD approach of [61] engages a numerical quadrature for the Gramians, $\mathbf{P}$ and $\mathbf{Q}$, using unit quadrature weights, $\rho_j = 1$ for $j = 1, \ldots, \ell_p$ (3.1) and $\phi_k = 1$ for $j = 1, \ldots, \ell_q$ in (3.3), and omitting the node at infinity in both cases. The approach of [61] was later referred to as Poor Man's Truncated Balanced Reduction in [45]; however, the implementation there is restricted to symmetric systems. Significantly, Balanced POD is not data-driven in the sense that we explore here, but rather it is both *projection-based* and *intrusive*, requiring explicit access to a state-space representation. Nonetheless, it is clear that one may immediately arrive at a transfer-function-based *data-driven* formulation of Balanced POD by using the QuadBT framework developed here.

Along similar lines, Gauss–Kronrod quadrature was used to approximate Hankel singular values in [12], but as with earlier methods, explicit state-space representations are still required. Similar quadrature strategies were later used in [18] to extend BT to structure-preserving model reduction for integro-differential equations.

In all these methods, frequency-domain quadrature to approximate Gramians can be viewed as a common thread shared with our framework. Significantly, we avoid explicit state-space projections by directly working with transfer function evaluations.

Evidently, numerical quadrature can also be applied to the time-domain representation of $\mathbf{P}$ as in (2.3), and this has also been considered as a strategy for approximate BT; see, e.g., [13, 14, 28, 38, 41, 54]. We explore these connections in section 5 where a time-domain formulation of this data-driven BT framework is considered.

ADI-type methods (see, e.g., [9, 11, 22, 35, 44, 53, 55]) are among the most commonly used tools for producing approximate low-rank Gramian factors $\widehat{\mathbf{U}}$ and $\widehat{\mathbf{L}}$. These methods are iterative in character and depend on shift parameters that determine the speed of convergence. Even though the resulting low-rank factor $\widehat{\mathbf{U}}$ will have a different structure than $\widetilde{\mathbf{U}}$ in (3.2), one may view either result as a low-rank factor for frequency-domain Balanced POD. Both factors span the same (Krylov) subspaces provided that the shift parameters are chosen to be the negative of the quadrature nodes [5]. Significantly, low-rank ADI solutions have also been connected to time-domain quadrature methods [13, 14], highlighting a powerful approximation-theoretic core unifying the approaches. Our approach is distinct from other methods in that it never constructs approximate solutions to $\mathbf{P}$ or $\mathbf{Q}$ nor attempts to approximate low-rank factors. Our QuadBT reduced model is constructed directly from data.

3.4. Comparison with interpolatory Loewner model reduction. For the sampling points $\{\mu_k\}_{k=1}^{N_q}$ and $\{\lambda_j\}_{j=1}^{N_p}$, consider $\mathbb{L} \in \mathbb{C}^{N_q \times N_p}$, $\mathbb{M} \in \mathbb{C}^{N_q \times N_p}$, defined elementwise as

$$\mathbb{L}_{k,j} = -\frac{H(\mu_k) - H(\lambda_j)}{\mu_k - \lambda_j} \quad \text{and} \quad \mathbb{M}_{k,j} = -\frac{\mu_k H(\mu_k) - \lambda_j H(\lambda_j)}{\mu_k - \lambda_j} \quad \text{for} \quad k = 1, 2, \ldots, N_q \quad \text{and} \quad j = 1, 2, \ldots, N_p. \tag{3.13}$$

The matrices $\mathbb{L}$ and $\mathbb{M}$ in (3.13) are called, respectively, the Loewner and shifted-Loewner matrices. Notice that $\mathbb{L}$ and $\mathbb{M}$ bear evident similarities to the key matrices $\widetilde{\mathbb{L}}$ in (3.6) and $\widetilde{\mathbb{M}}$ in (3.8) that are central to QuadBT. Indeed, assuming that the node at infinity is absent from the quadrature rules in (3.1) and (3.3) and that the quadrature nodes in QuadBT determine $\mu_k = \imath\omega_k$ for $k = 1, 2, \ldots, N_q$ and $\lambda_j = \imath\zeta_j$ for $j = 1, 2, \ldots, N_p$ in (3.13), the matrices $\widetilde{\mathbb{L}}$ in (3.6) and $\widetilde{\mathbb{M}}$ in (3.8) used in QuadBT will be diagonal scalings of $\mathbb{L}$ and $\mathbb{M}$:

$$\widetilde{\mathbb{L}} = \mathsf{diag}(\phi_1, \ldots, \phi_{N_q})\, \mathbb{L}\, \mathsf{diag}(\rho_1, \ldots, \rho_{N_p}) \text{ and } \widetilde{\mathbb{M}} = \mathsf{diag}(\phi_1, \ldots, \phi_{N_q})\, \mathbb{M}\, \mathsf{diag}(\rho_1, \ldots, \rho_{N_p}).$$

The matrices $\mathbb{L}$ and $\mathbb{M}$ of (3.13) were originally conceived to play a role in quite a different model reduction scheme which we refer to as the *interpolatory Loewner model reduction framework*. This model reduction framework was introduced in [37] as a data-driven interpolation-based system identification and complexity reduction technique. It has become a popular and powerful tool for model reduction over the last decade and has been successfully extended to parametrized linear dynamical systems [30], structured linearized systems [52], and bilinear [3] and quadratic-bilinear systems [23] and has been formulated to work with time-domain data as well [42].

The principal feature of the *interpolatory Loewner framework* is the construction of a rational interpolant matching the given sampled transfer function data. This provides immediately a realization of a reduced model that matches the system response at specified driving frequencies (or equivalently, at specified complex interpolation points). The quality of the final reduced model is strongly tied to the choice of interpolation points and in the MIMO setting, also to the choice of tangent directions.

If some additional flexibility exists in where the system may be sampled, $\mathcal{H}_2$-optimal reduced-order interpolants may be feasibly computed [6]. For a detailed account of theoretical developments as well as implementation details for Loewner-based methods relating especially to rational interpolation, we refer to [2, 4] and references therein.

Although our QuadBT framework makes use of matrices that are, at least in many cases of interest, diagonally scaled Loewner and shifted Loewner matrices, rational interpolation plays no role for us. Instead, we have observed that there are strong links among (a) quadrature rules (3.1) and (3.3), (b) the (limiting) implicit Gramian factorizations described in (3.5), and (c) the related Loewner/shifted Loewner factorizations. These links couple together the features developed in Propositions 3.1, 3.2, and 3.3 and allow us ultimately to mimic BT in a purely data-driven manner.

3.4.1. Numerical examples. We use the [heat] and [iss1r] models from sections 3.1.1 and 3.2.1 as test cases. QuadBT is implemented using only the [ExpTrap] quadrature.

We choose 120 logarithmically spaced points in the interval $[10^{-3}, 10^{3}]\dot{\imath}$ for the [heat] model and 400 logarithmically spaced points in the interval $[10^{-1}, 10^{2}]\dot{\imath}$ for the [iss1r] model. In the interpolatory Loewner framework, the singular values of $\mathbb{L}$ and/or the augmented matrix $[\ \mathbb{L}\ \ \mathbb{M}\ ]$ play a fundamental role in determining reduction order, analogous to Hankel singular values in BT; see [2, section 4.3]. Therefore, in addition to the QuadBT quantities, we construct $\mathbb{L}$ and $\mathbb{M}$ corresponding to the same transfer function samples and compute (i) the true Hankel singular values, (ii) QuadBT-based Hankel singular values, and (iii) the singular values of $\mathbb{L}$ and $[\ \mathbb{L}\ \ \mathbb{M}\ ]$. Results are depicted in Figure 3.5. While the Hankel singular values computed via [ExpTrap] approximate the true ones fairly well, the singular values corresponding to the Loewner method appear to follow the trend only up to a scaling factor. This simple example shows the importance of the quadrature weights appearing in the QuadBT framework.

Next, we consider the SISO version of the [iss1r] model and apply both QuadBT with [ExpTrap] using 200 logarithmically spaced points in the interval $[10^{-1}, 10^{2}]\dot{\imath}$ and the Loewner-based interpolation method. We vary the reduction order from $r = 2$ to $r = 24$ and collect the $\mathcal{H}_\infty$ and the $\mathcal{H}_2$ norms of the error systems for both methods. The results are presented in Figure 3.6 where the label [Loewner-200] refers to the Loewner method with $N = 200$ points. Figure 3.6 shows that *for the same transfer function data*, it pays off to use appropriate quadrature weights to perform QuadBT as QuadBT outperforms the Loewner approach and mimics BT much more closely.

For the two quadrature rules that we consider here (described in Appendix A), the quadrature nodes are closely related to spectral/pseudospectral approximations

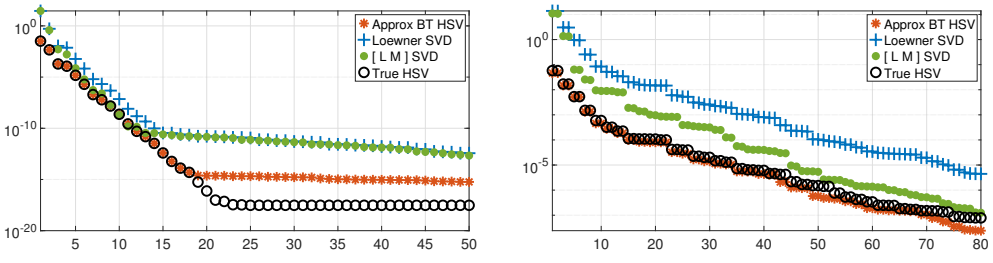

FIG. 3.5. *The computed Hankel singular values (HSV) for the heat model (left) and the* [iss1r] *model (right).*

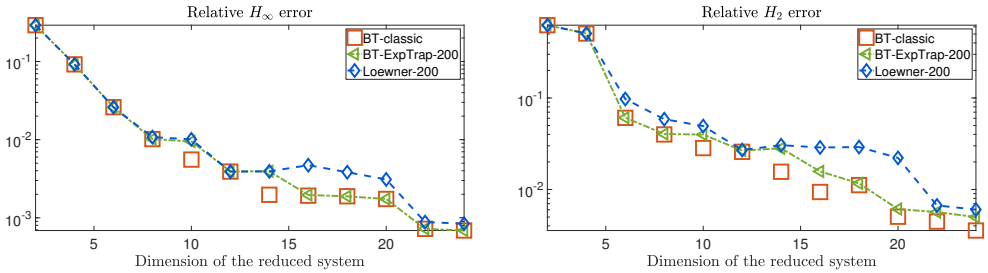

FIG. 3.6. *Relative $\mathcal{H}_\infty$ (left) and $\mathcal{H}_2$ (right) approximation errors for different methods applied to the* [iss1r] *model.*

of the integrand and so will also be effective for producing accurate interpolants for interpolation-based methods including those that typify the Loewner model reduction framework. Other choices for quadrature rules (e.g., Gauss rules) might not yield such effective interpolants.

4. Refinements and extensions.

4.1. Keeping it real. The original system has been presumed to be real, insofar as real-valued inputs and real-valued initial states assure real-valued outputs. This guarantees in turn that a *real* realization involving matrices $\mathbf{E}, \mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{p\times n}$ is possible, though not necessarily explicitly available. Equivalently, the associated system transfer function $\mathsf{H}(s) = \mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}$ displays a conjugate symmetry of system poles and residues (i.e., complex poles and residues must occur in conjugate pairs). If we utilize *symmetric quadrature rules* in the approximations given in (3.1) and (3.3)—that is, if the sets of quadrature nodes and associated weights, $\{(\zeta_j, \rho_j)\}_{j=1}^{N_p}$ and $\{(\omega_k, \phi_k\}_{k=1}^{N_q}$, are symmetrically distributed on the real axis with respect to 0—then we will be able to retain this conjugate symmetry, accomplish computations equivalent to steps 2, 3, and 4 of QuadBT using only real arithmetic, and be assured that the final reduced model will inherit the conjugate symmetry of the original system, thus representing a reduced-order real dynamical system.

For simplicity, we focus on the SISO case and suppose that both node sets have an even number of points that are symmetrically distributed across the real axis, say $N_p = 2\nu_p$ and $N_q = 2\nu_q$. (More general settings are considered in [2, page 203]). Relabel these node sets as

$$\zeta_{-\nu_p} < \zeta_{-\nu_p+1} < \cdots < \zeta_{-1} < 0 < \zeta_1 < \cdots < \zeta_{\nu_p-1} < \zeta_{\nu_p},$$
$$\omega_{-\nu_q} < \omega_{-\nu_q+1} < \cdots < \omega_{-1} < 0 < \omega_1 < \cdots < \omega_{\nu_q-1} < \omega_{\nu_q},$$

noting that $\zeta_{-j} = -\zeta_j$ and with corresponding weights, $\rho_{-j} = \rho_j$, for $j = 1, \ldots, \nu_p$. Similarly, we have $\omega_{-k} = -\omega_k$ and $\phi_{-k} = \phi_k$ for $k = 1, \ldots, \nu_q$. The nodes/weights are then reordered with respect to this $\pm$ node pairing:

$$\{\zeta_1, \zeta_{-1}, \zeta_2, \zeta_{-2}, \ldots, \zeta_{\nu_p}, \zeta_{-\nu_p}\} \quad \text{and} \quad \{\omega_1, \omega_{-1}, \omega_2, \omega_{-2}, \ldots, \omega_{\nu_q}, \omega_{-\nu_q}\}. \tag{4.1}$$

When the samples obtained in step 1 of QuadBT are reordered in the same way, they appear now as a sequence of conjugate pairs:

$$\{H(\mathbf{i}\zeta_j), \overline{H(\mathbf{i}\zeta_j)}\}_{j=1}^{\nu_p} \quad \text{and} \quad \{H(\mathbf{i}\omega_k), \overline{H(\mathbf{i}\omega_k)}\}_{k=1}^{\nu_q}.$$

Notice that one need only explicitly evaluate the first function of each pair.

Now instead of (3.6), we reconsider the construction of $\widetilde{\mathbb{L}}$ taking into account the new node ordering given in (4.1). With respect to this node ordering, $\widetilde{\mathbb{L}}$ can be partitioned into 2×2 blocks, $\widetilde{\mathbb{L}}^{(2)}_{k,j}$, conforming to the conjugate pairing of quadrature nodes in (4.1)—left data at $(\imath\omega_k, -\imath\omega_k)$; right data at $(\imath\zeta_j, -\imath\zeta_j)$:

$$\widetilde{\mathbb{L}}^{(2)}_{k,j} = -\phi_k\rho_j \begin{bmatrix} \frac{H(\imath\,\omega_k)-H(\imath\,\zeta_j)}{\imath\,\omega_k-\imath\,\zeta_j} & \frac{H(\imath\,\omega_k)-\overline{H(\imath\,\zeta_j)}}{\imath\,\omega_k-(-\imath\,\zeta_j)} \\ \frac{\overline{H(\imath\,\omega_k)}-H(\imath\,\zeta_j)}{(-\imath\,\omega_k)-\imath\,\zeta_j} & \frac{\overline{H(\imath\,\omega_k)}-\overline{H(\imath\,\zeta_j)}}{(-\imath\,\omega_k)-(-\imath\,\zeta_j)} \end{bmatrix}$$

for $j = 1,\ldots,\nu_p$ and $k = 1,\ldots,\nu_q$. Note that each $\widetilde{\mathbb{L}}^{(2)}_{k,j}$ satisfies a Sylvester equation:

$$\begin{aligned} &\widetilde{\mathbb{L}}^{(2)}_{k,j}\begin{bmatrix} \imath\,\omega_k & 0 \\ 0 & -\imath\,\omega_k \end{bmatrix} - \begin{bmatrix} \imath\,\zeta_j & 0 \\ 0 & -\imath\,\zeta_j \end{bmatrix}\widetilde{\mathbb{L}}^{(2)}_{k,j} \\ &= \phi_k\rho_j\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}\begin{bmatrix} H(\imath\,\zeta_j) & \overline{H(\imath\,\zeta_j)} \end{bmatrix} - \begin{bmatrix} H(\imath\,\omega_k) \\ \overline{H(\imath\,\omega_k)} \end{bmatrix}\begin{bmatrix} 1 & 1 \end{bmatrix}\right). \end{aligned}$$

This uniquely determines each $\widetilde{\mathbb{L}}^{(2)}_{k,j}$ (and hence, $\widetilde{\mathbb{L}}$) when the sets of quadrature nodes, $\{\zeta_j\}$ and $\{\omega_k\}$, are disjoint as we have assumed.

$\widetilde{\mathbb{L}}$ is unitarily equivalent to a *real* matrix as we next demonstrate explicitly. Define $\mathbf{J} = \frac{1}{\sqrt{2}}\left[\begin{smallmatrix} 1 & -\imath \\ 1 & \imath \end{smallmatrix}\right]$, and consider a unitary equivalence defined by

$$\widetilde{\mathbb{L}}^R = (\mathbf{I}_{\nu_q}\otimes\mathbf{J}^\star)\,\widetilde{\mathbb{L}}\,(\mathbf{I}_{\nu_p}\otimes\mathbf{J}).$$

$\widetilde{\mathbb{L}}^R$ can be partitioned into 2×2 blocks, conforming precisely to the earlier partitioning of $\widetilde{\mathbb{L}}$, and indeed one finds, for each block, $\widetilde{\mathbb{L}}^{(2)R}_{k,j} = \mathbf{J}^\star\widetilde{\mathbb{L}}^{(2)}_{k,j}\mathbf{J}$ for $j = 1,\ldots,\nu_p$ and $k = 1,\ldots,\nu_q$. One may check directly that $\widetilde{\mathbb{L}}^{(2)R}_{k,j}$ is real. Indeed, each block may be determined uniquely from a corresponding (real) Sylvester equation:

$$\widetilde{\mathbb{L}}^{(2)R}_{k,j}\begin{bmatrix} 0 & \omega_k \\ -\omega_k & 0 \end{bmatrix} - \begin{bmatrix} 0 & \zeta_j \\ -\zeta_j & 0 \end{bmatrix}\widetilde{\mathbb{L}}^{(2)R}_{k,j} = 2\phi_k\rho_j\begin{bmatrix} \mathsf{Re}\,(H(\imath\,\zeta_j)-H(\imath\,\omega_k)) & \mathsf{Im}\,(H(\imath\,\zeta_j)) \\ \mathsf{Im}\,(H(\imath\,\omega_k)) & 0 \end{bmatrix}$$

for $j = 1,\ldots,\nu_p$ and $k = 1,\ldots,\nu_q$. One may solve these Sylvester equations for $\widetilde{\mathbb{L}}^{(2)R}_{k,j}$, in real arithmetic, independently of one another in parallel.

Similar considerations may be applied to (3.8), (3.9), and (3.10), which define, respectively, the complementary matrix $\widetilde{\mathbb{M}}$, and vectors $\widetilde{\mathbb{h}}$ and $\widetilde{\mathbb{g}}$. For $\widetilde{\mathbb{M}}$, we find as before, a unitarily equivalent real matrix:

$$\widetilde{\mathbb{M}}^R = (\mathbf{I}_{\nu_q}\otimes\mathbf{J}^\star)\,\widetilde{\mathbb{M}}\,(\mathbf{I}_{\nu_p}\otimes\mathbf{J}),$$

which may be partitioned into submatrices, $\widetilde{\mathbb{M}}^{(2)R}_{k,j} = \mathbf{J}^\star\widetilde{\mathbb{M}}^{(2)}_{k,j}\mathbf{J}$, for $j = 1,\ldots,\nu_p$ and $k = 1,\ldots,\nu_q$, conforming to the partitioning for $\widetilde{\mathbb{L}}^R$. These submatrices are directly computable from real Sylvester equations:

$$\begin{aligned} &\widetilde{\mathbb{M}}^{(2)R}_{k,j}\begin{bmatrix} 0 & \omega_k \\ -\omega_k & 0 \end{bmatrix} - \begin{bmatrix} 0 & \zeta_j \\ -\zeta_j & 0 \end{bmatrix}\widetilde{\mathbb{M}}^{(2)R}_{k,j} \\ &= 2\phi_k\rho_j\begin{bmatrix} \mathsf{Im}\,(\omega_k\,H(\imath\,\omega_k)-\zeta_j\,H(\imath\,\zeta_j)) & \mathsf{Re}\,(\zeta_j\,H(\imath\,\zeta_j)) \\ \mathsf{Re}\,(\omega_k\,H(\imath\,\omega_k)) & 0 \end{bmatrix} \end{aligned}$$

for $j = 1, \ldots, \nu_p, \quad k = 1, \ldots, \nu_q$. For $\widetilde{\mathbb{h}}$ and $\widetilde{\mathbf{g}}$, we may define corresponding real vectors,

$$\widetilde{\mathbb{h}}^{(R)} = (\mathbf{I}_{\nu_q} \otimes \mathbf{J}^\star)\widetilde{\mathbb{h}} \quad \text{and} \quad \widetilde{\mathbf{g}}^{(R)T} = \widetilde{\mathbf{g}}^T(\mathbf{I}_{\nu_p} \otimes \mathbf{J}).$$

If we return to QuadBT (Algorithm 2) and replace $\widetilde{\mathbb{L}}$, $\widetilde{\mathbb{M}}$, $\widetilde{\mathbb{h}}$, and $\widetilde{\mathbf{g}}$ with their real counterparts derived above, $\widetilde{\mathbb{L}}^{(R)}$, $\widetilde{\mathbb{M}}^{(R)}$, $\widetilde{\mathbb{h}}^{(R)}$, and $\widetilde{\mathbf{g}}^{(R)}$, then the computations in steps 3 and 4 may be done entirely in real arithmetic, and the final reduced model defined in step 4 will be real, even in the presence of rounding errors.

4.2. MIMO systems. We have thus far presented an analysis framework specific to SISO dynamical systems; however, the MIMO case follows from this nearly immediately; the only change occurs in the construction of $\widetilde{\mathbb{L}}$, $\widetilde{\mathbb{M}}$, $\widetilde{\mathbb{h}}$ and $\widetilde{\mathbf{c}}_r$ and not in Algorithm 2 itself. Specifically, assume that the underlying dynamical system has m inputs and p outputs (e.g., as in (2.1)). In this case, the transfer function is a matrix-valued rational function, and we will denote it by $\mathbf{H}(s)$ so that for any frequency $\hat{s} \in \mathbb{C}$, we have $\mathbf{H}(\hat{s}) \in \mathbb{C}^{p\times m}$. Therefore, assuming the same set-up as in Proposition 3.1, in defining $\widetilde{\mathbb{L}}$, we have

$$\widetilde{\mathbb{L}}_{k,j} = -\phi_k\rho_j \frac{\mathbf{H}(\imath\omega_k) - \mathbf{H}(\imath\zeta_j)}{\imath\omega_k - \imath\zeta_j} \in \mathbb{C}^{p\times m} \tag{4.2}$$

as the (k,j)th *block* of $\widetilde{\mathbb{L}}$ and therefore $\widetilde{\mathbb{L}} \in \mathbb{C}^{(p\cdot N_q)\times(m\cdot N_p)}$. Similarly, in Proposition 3.3, we will have

$$\widetilde{\mathbb{M}}_{k,j} = -\phi_k\rho_j \frac{\imath\omega_k\mathbf{H}(\imath\omega_k) - \imath\zeta_j\mathbf{H}(i\zeta_j)}{\imath\omega_k - \imath\zeta_j} \in \mathbb{C}^{p\times m} \tag{4.3}$$

as the (k,j)th *block* of $\widetilde{\mathbb{M}}$ and thus $\widetilde{\mathbb{M}} \in \mathbb{C}^{(p\cdot N_q)\times(m\cdot N_p)}$. Since we are dealing with MIMO systems, we replace $\widetilde{\mathbb{h}}$ with $\widetilde{\mathbb{H}} = \widetilde{\mathbf{L}}^*\mathbf{B} \in \mathbb{C}^{(N_q\cdot p)\times m}$ and $\widetilde{\mathbf{g}}^T$ with $\widetilde{\mathbb{G}} = \mathbf{C}\widetilde{\mathbf{U}} \in \mathbb{C}^{p\times(N_p\cdot m)}$ with the kth row block of $\widetilde{\mathbb{H}}$ and jth columns block of $\widetilde{\mathbb{G}}$ given by

$$\widetilde{\mathbb{H}}_k = \phi_k\mathbf{H}(\imath\omega_k) \in \mathbb{C}^{p\times m} \quad \text{and} \quad \widetilde{\mathbb{G}}_j = \rho_j\mathbf{H}(\imath\zeta_j) \in \mathbb{C}^{p\times m}.$$

Then, Algorithm 2 simply proceeds with these new definitions.

4.2.1. Numerical example: The MIMO case. We examine performance for the [iss1r] numerical test case, which has $p = 3$ outputs and $m = 3$ inputs. Modifications specified above in section 4.2 are followed to adapt QuadBT to a MIMO setting.

Choose both 200 and 400 logarithmically spaced points in the interval $[10^{-1}, 10^2]\imath$ as nodes for [ExpTrap] (the left and right nodes are chosen to be distinct). For [B/CC], choose $L = 10$ and $L = 10.5$ for $\widetilde{\mathbf{U}}$ and $\widetilde{\mathbf{L}}$, respectively, using both 200 and 400 nodes. We then vary the reduction order from $r = 2$ to $r = 24$ and compute the relative $\mathcal{H}_\infty$ and $\mathcal{H}_2$ errors, corresponding to both quadrature schemes. The results are presented in Figure 4.1. As in the SISO case, QuadBT accurately mimics BT performance. In this MIMO example, [B/CC] outperforms [ExpTrap].

4.3. Discrete-time systems. Consider a discrete-time dynamical system characterized by the difference equations

$$\begin{aligned} \mathbf{E}\mathbf{x}_{k+1} &= \mathbf{A}\,\mathbf{x}_k + \mathbf{B}\,\mathbf{u}_k, \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k, \end{aligned} \tag{4.4}$$

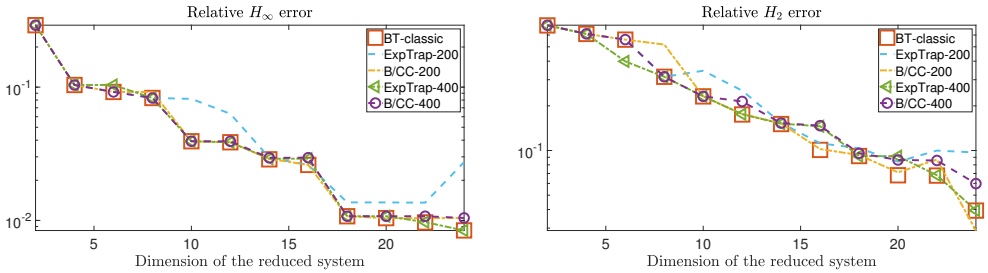

FIG. 4.1. *Relative $\mathcal{H}_\infty$ (left) and $\mathcal{H}_2$ (right) approximation errors for different quadrature schemes (*[ExpTrap] *vs.* [B/CC]*) and different numbers of nodes (*200 *vs.* 400*) for* [iss1r] *MIMO model.*

where the input mapping is given by $\mathbf{u} : \mathbb{Z} \to \mathbb{R}^m$, the state trajectory/variable is $\mathbf{x} : \mathbb{Z} \to \mathbb{R}^n$, and the output mapping is $y : \mathbb{Z} \to \mathbb{R}^p$. The system matrices, as in the continuous case, are given by $\mathbf{A}, \mathbf{E} \in \mathbb{R}^{n\times n}$ and $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{p\times n}$. We assume that the discrete system is *asymptotically stable*, i.e., the eigenvalues of the matrix pencil $(\mathbf{A}, \mathbf{E})$ are located in the open unit disc.

The reachability Gramian $\mathbf{P}$ in the frequency domain is given by

$$\mathbf{P} = \frac{1}{2\pi}\int_0^{2\pi} (e^{\boldsymbol{i}\zeta}\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}\mathbf{B}^T(e^{-\boldsymbol{i}\zeta}\mathbf{E}^T - \mathbf{A}^T)^{-1}d\zeta. \tag{4.5}$$

Similarly, the observability Gramian can be represented as

$$\mathbf{Q} = \frac{1}{2\pi}\int_0^{2\pi} (e^{-\boldsymbol{i}\omega}\mathbf{E}^T - \mathbf{A}^T)^{-1}\mathbf{C}^T\mathbf{C}(e^{\boldsymbol{i}\omega}\mathbf{E} - \mathbf{A})^{-1}d\omega. \tag{4.6}$$

$\mathbf{P}$ and $\mathbf{Q}$ satisfy the following Stein equations:

$$\mathbf{A}\mathbf{P}\mathbf{A}^T + \mathbf{B}\mathbf{B}^T = \mathbf{E}\mathbf{P}\mathbf{E}^T, \quad \mathbf{A}^T\mathbf{Q}\mathbf{A} + \mathbf{C}^T\mathbf{C} = \mathbf{E}^T\mathbf{Q}\mathbf{E}. \tag{4.7}$$

As in the continuous-time case, we employ a numerical quadrature rule to approximate the the Gramian $\mathbf{P}$ defined in (4.5) to obtain the approximant

$$\mathbf{P} \approx \widetilde{\mathbf{P}} = \sum_{j=1}^{N_p} \rho_j^2 (e^{\boldsymbol{i}\omega_j}\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}\mathbf{B}^T(e^{-\boldsymbol{i}\omega_j}\mathbf{E}^T - \mathbf{A}^T)^{-1} \tag{4.8}$$

with ρ_j^2 and $\omega_j \in [0, 2\pi]$ denoting, respectively, quadrature weights and nodes.

We apply a similar quadrature approximation for $\mathbf{Q}$ and obtain

$$\mathbf{Q} \approx \widetilde{\mathbf{Q}} = \sum_{k=1}^{N_q} \phi_k^2 (e^{-\boldsymbol{i}\zeta_k}\mathbf{E}^T - \mathbf{A}^T)^{-1}\mathbf{C}^T\mathbf{C}(e^{\boldsymbol{i}\zeta_k}\mathbf{E} - \mathbf{A})^{-1}, \tag{4.9}$$

where ϕ_k^2 and $\zeta_k \in [0, 2\pi]$ denote, respectively, quadrature weights and nodes associated with approximating $\mathbf{Q}$. The rest of the QuadBT formulation for discrete systems follows immediately as in Algorithm 2. We omit those details here for brevity.

4.3.1. Numerical example: The discrete-time case. We analyze an nth-order lowpass digital Butterworth filter with normalized cutoff frequency W_c, using the parameter values $n = 40$ and $W_c = 0.6$. (A realization can be obtained in MATLAB using the command butter.)

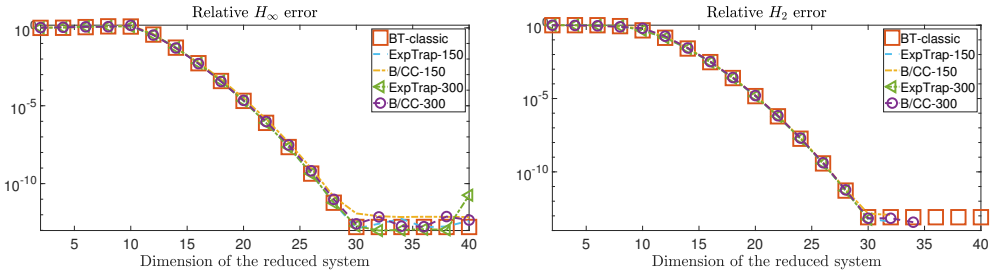

FIG. 4.2. *The relative $\mathcal{H}_\infty$ approximation error for different quadratures applied to the Butterworth digital filter model.*

Choose both 150 and 300 linearly spaced points in the interval $[0, 2\pi)$ as the ω_j's and ζ_k's in (4.8) and (4.9), for [ExpTrap]. Here, the left and right nodes are chosen to be the same. For the [B/CC] quadrature, the values of ω_j's and ζ_k's are chosen to be Chebyshev nodes instead, again both using 150 and 300 nodes. We vary the reduction order from $r = 2$ to $r = 40$ and collect the relative $\mathcal{H}_\infty$ and $\mathcal{H}_2$ errors, corresponding to both quadrature schemes and classical BT. The results are presented in Figure 4.2. QuadBT via both [ExpTrap] and [B/CC] almost exactly replicates the BT behavior, illustrating the success of our proposed method in the discrete-time setting as well.

4.4. An infinite-dimensional example. We revisit a numerical example from [6] describing the response of an infinite-dimensional linear system, in this case a one-dimensional heat equation. The evolution of the temperature distribution on a semi-infinite rod is characterized by the PDE $\frac{\partial y}{\partial t} - \frac{\partial^2 y}{\partial z^2} = 0$, where $y(z,t)$ denotes the temperature at position z and time t. The temperature is controlled at the location $z = 0$, while the temperature at $z = 1$ is the quantity of interest. By enforcing particular initial and boundary conditions as in [6], one may explicitly derive a transfer function: $H(s) = \frac{Y(1,s)}{Y(0,s)} = e^{-\sqrt{s}}$, describing the mapping from $y(0,t)$ to $y(1;t)$, where $Y(\cdot, s)$ denotes the Laplace transform of $y(\cdot, t)$. BT can be approached within an operator-theoretic framework using ideas found, for example, in [19, 48]. However, we are able to bypass these subtleties, since the transfer function $H(s)$ is immediately available, as well as its derivative $H'(s) = -\frac{e^{-\sqrt{s}}}{2\sqrt{s}}$.

We directly apply QuadBT with [ExpTrap] using 50 logarithmically spaced points in the interval $[10^{-2}, 10^{1}]\imath$. The results are displayed in Figure 4.3. In the left plot, we show the approximation error, $|\, H(\imath\omega) - H_r(\imath\omega)\,|$, for $r = 8, 12, 16, 20$. In the right side of Figure 4.3, we present the approximate Hankel singular values of the transfer function $H(s) = e^{-\sqrt{s}}$. Reduced models via QuadBT successfully approximate the original transfer function without requiring a discretization of the underlying PDE.

Remark 4.1. The numerical quadratures that we employ in our experiments converge exponentially fast with respect to the number of quadrature nodes used (see Appendix A). Thus, one might not anticipate the need for a large number of quadrature nodes in order to ensure accuracy. Nonetheless, exponential convergence is an asymptotic property that might not be readily observed in circumstances where the spectral abscissa of $\mathbf{A}$ is close to 0 or if $\mathbf{A}$ is significantly nonnormal. In such circumstances, the number of quadrature nodes that are required for accurate model reduction may grow quite large, and the cost of QuadBT may become dominated by the cost of the SVD of $\widetilde{\mathbb{L}}$ in (3.11). Indeed, with a large number, N, of quadrature

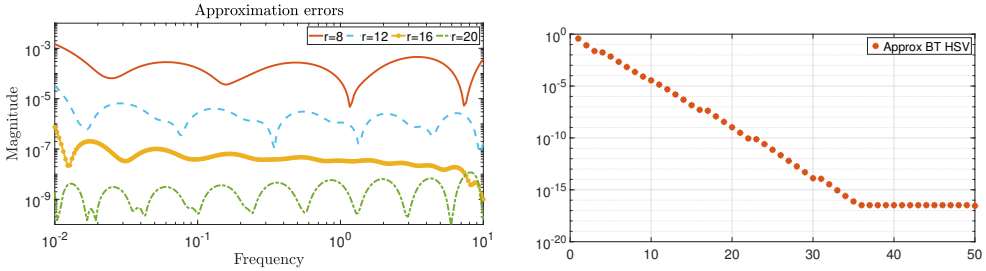

FIG. 4.3. *The approximation error for different reduction orders (left) and the approximated Hankel singular values (HSV) (right) for* $H(s) = e^{-\sqrt{s}}$.

nodes $N_q \approx N_p \approx N$, say, the cost of computing step 2 of Algorithm 2 in order to extract an order r reduced model, will grow at least as $\mathcal{O}(N^2 r)$, using standard approaches. The computational burden of this step could be moderated potentially by using randomized algorithms, e.g., randomized SVD approaches [26], and such approaches may not require that $\widetilde{\mathbb{L}}$ be explicitly available as well.

5. Time-domain formulation. Thus far, we have developed ideas that are central to QuadBT presuming access to *frequency-domain data*, i.e., data resulting from transfer function sampling at selected points in the complex plane. This formulation utilized a frequency-domain description of $\mathbf{P}$ and $\mathbf{Q}$, expressed in (2.4) and (2.5). In this section, we establish an analogous framework in the time domain, sampling the system response at selected instants in time and then using a time-domain representation of $\mathbf{P}$ and $\mathbf{Q}$ to derive a realization corresponding to an approximate BT. There are elements in common with time-domain Balanced POD [49, 54, 61], which uses time-domain state snapshots. We will show that, as in the frequency-domain version, those implementations can be performed without access to state (internal) data. In our presentation, we will adapt the notation from [49] to our formulation. We will focus on the SISO case for simplicity. However, unlike in [49], we will include the $\mathbf{E}$ term in our formulation to keep the paper consistent with the earlier frequency-domain framework of QuadBT.

5.1. Time-domain QuadBT for continuous-time systems. As illustrated in [49, 54, 61], the Balanced POD approach can be applied in the time domain by considering a numerical quadrature approximation to time domain definitions of the Gramians, as given, e.g., in (2.3) for the reachability Gramian. We also note that in the seminal paper [38] where BT was originally introduced, Moore had already mentioned the potential of quadrature-based approximation to Gramians, thus motivating time-domain Balanced POD via empirical Gramians. We refer to the recent work [29] for a comparative study on BT-related methods employing empirical Gramians. Following on to themes developed here, we will illustrate that the time-domain Balanced POD formulation can be fully implemented using only input/output data and in particular, without access to state variables or internally characterized state realizations.

Assume that the time-domain formula (2.3) for $\mathbf{P}$ is approximated via a numerical quadrature on $(0, \infty)$:

$$\mathbf{P} = \int_0^\infty e^{\mathbf{E}^{-1}\mathbf{A}t}\mathbf{E}^{-1}\mathbf{b}\mathbf{b}^T\mathbf{E}^{-T}e^{\mathbf{A}^T\mathbf{E}^{-T}t}dt \approx \sum_{j=1}^{N_p} \rho_j^2 e^{\mathbf{E}^{-1}\mathbf{A}t_j}\mathbf{E}^{-1}\mathbf{b}\mathbf{b}^T\mathbf{E}^{-T}e^{\mathbf{A}^T\mathbf{E}^{-T}t_j}, \tag{5.1}$$

where (ρ_j^2, t_j) for $j = 1, \ldots, N_p$ denote quadrature weight/node pairs on $(0, \infty)$. Define

$$\widetilde{\mathbf{U}} = \begin{bmatrix} \rho_1 e^{\mathbf{E}^{-1}\mathbf{A}t_1}\mathbf{E}^{-1}\mathbf{b} & \cdots & \rho_{N_p} e^{\mathbf{E}^{-1}\mathbf{A}t_{N_p}}\mathbf{E}^{-1}\mathbf{b} \end{bmatrix} \in \mathbb{R}^{n\times N_p}. \tag{5.2}$$

Then, $\widetilde{\mathbf{P}} = \widetilde{\mathbf{U}}\widetilde{\mathbf{U}}^*$ is an approximation to $\mathbf{P}$ with $\widetilde{\mathbf{U}}$ as the (approximate) square-root factor. As pointed out in [49], the columns of $\widetilde{\mathbf{U}}$ without the quadrature weights can be viewed as the snapshot data obtained from simulating the system

$$\mathbf{E}\dot{\mathbf{w}} = \mathbf{A}\mathbf{w} \quad \text{with} \quad \mathbf{w}(0) = \mathbf{E}^{-1}\mathbf{b}. \tag{5.3}$$

One may form a time-domain quadrature approximation to $\mathbf{Q}$ in a similar fashion:

$$\mathbf{Q} = \int_0^\infty e^{\mathbf{E}^{-T}\mathbf{A}^T t}\mathbf{E}^{-T}\mathbf{c}\mathbf{c}^T\mathbf{E}^{-1}e^{\mathbf{A}\mathbf{E}^{-1}t}dt \approx \sum_{i=1}^{N_q} \phi_i^2 e^{\mathbf{E}^{-T}\mathbf{A}^T\tau_i}\mathbf{E}^{-T}\mathbf{c}\mathbf{c}^T\mathbf{E}^{-1}e^{\mathbf{A}\mathbf{E}^{-1}\tau_i} \tag{5.4}$$

with (ϕ_i^2, τ_j) for $i = 1, \ldots, N_q$ being quadrature weight/node pairs on $(0, \infty)$. Evidently, the corresponding square-root factor is

$$\widetilde{\mathbf{L}}^T = \begin{bmatrix} \phi_1\mathbf{c}^T\mathbf{E}^{-1}e^{\mathbf{A}\mathbf{E}^{-1}\tau_1} \\ \vdots \\ \phi_{N_q}\mathbf{c}^T\mathbf{E}^{-1}e^{\mathbf{A}\mathbf{E}^{-1}\tau_{N_q}} \end{bmatrix} \in \mathbb{R}^{N_q\times n}, \tag{5.5}$$

so that $\widetilde{\mathbf{Q}} = \widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^*$ approximates $\mathbf{Q}$. Similar to $\widetilde{\mathbf{U}}$, the columns of $\widetilde{\mathbf{L}}$ without quadrature weights can be viewed as the snapshot data from simulating the dual system given by

$$\mathbf{E}^T\dot{\mathbf{z}} = \mathbf{A}^T\mathbf{z} \quad \text{with} \quad \mathbf{z}(0) = \mathbf{E}^{-T}\mathbf{c}. \tag{5.6}$$

The square-root implementation of time-domain Balanced POD [49] proceeds by using $\widetilde{\mathbf{U}}$ and $\widetilde{\mathbf{L}}$ in place of $\mathbf{U}$ and $\mathbf{L}$ in the square-root implementation of BT in Algorithm 1. So far, this still reflects a realization-*dependent* implementation, i.e., we require access to state-space representation. We will avoid this requirement below.

As we did with frequency-domain data, we inspect $\widetilde{\mathbb{L}} = \widetilde{\mathbf{L}}^T\mathbf{E}\widetilde{\mathbf{U}} \in \mathbb{R}^{N_q\times N_p}$. Using (5.2) and (5.5), the (i,j)th component of $\widetilde{\mathbb{L}}$ is obtained as

$$\widetilde{\mathbb{L}}_{ij} = \phi_i\rho_j\mathbf{c}^T\mathbf{E}^{-1}e^{\mathbf{A}\mathbf{E}^{-1}\tau_i}\mathbf{E}e^{\mathbf{E}^{-1}\mathbf{A}t_j}\mathbf{E}^{-1}\mathbf{b} = \phi_i\rho_j h(\tau_i + t_j), \tag{5.7}$$

where $h : \mathbb{R} \to \mathbb{R}$ is the impulse response of the underlying dynamical system, as

$$h(t) = \mathbf{c}^T e^{\mathbf{E}^{-1}\mathbf{A}t}\mathbf{E}^{-1}\mathbf{b}. \tag{5.8}$$

The last equality in (5.7) follows from the observation

$$\left(\mathbf{E}^{-1}e^{\mathbf{A}\mathbf{E}^{-1}\tau_i}\right)\mathbf{E}e^{\mathbf{E}^{-1}\mathbf{A}t_j} = \left(e^{\mathbf{E}^{-1}\mathbf{A}\tau_i}\mathbf{E}^{-1}\right)\mathbf{E}e^{\mathbf{E}^{-1}\mathbf{A}t_j} = e^{\mathbf{E}^{-1}\mathbf{A}(\tau_i+t_j)}, \tag{5.9}$$

which directly follows from inspection of the partial sums of the series expansion of the parenthesized expression. Using similar ideas, one can also show that

$$\widetilde{\mathbb{h}} = \widetilde{\mathbf{L}}^T\mathbf{b} = \begin{bmatrix} \phi_1\mathbf{c}^T\mathbf{E}^{-1}e^{\mathbf{A}\mathbf{E}^{-1}\tau_1}\mathbf{b} \\ \vdots \\ \phi_{N_q}\mathbf{c}^T\mathbf{E}^{-1}e^{\mathbf{A}\mathbf{E}^{-1}\tau_{N_q}}\mathbf{b} \end{bmatrix} = \begin{bmatrix} \phi_1 h(\tau_1) \\ \vdots \\ \phi_{N_q} h(\tau_{N_q}) \end{bmatrix} \tag{5.10}$$

and

$$\begin{aligned}\widetilde{\mathbf{g}}^T = \mathbf{c}^T\widetilde{\mathbf{U}} &= \begin{bmatrix}\rho_1\mathbf{c}^T e^{\mathbf{E}^{-1}\mathbf{A}t_1}\mathbf{E}^{-1}\mathbf{b} & \cdots & \mathbf{c}^T\rho_{N_p}e^{\mathbf{E}^{-1}\mathbf{A}t_{N_p}}\mathbf{E}^{-1}\mathbf{b}\end{bmatrix}\\ &= \begin{bmatrix}\rho_1 h(t_1) & \cdots & \rho_{N_p}h(t_{N_p})\end{bmatrix}.\end{aligned} \tag{5.11}$$

Now, let us consider the (i,j)th component of $\widetilde{\mathbb{M}}$:

$$\widetilde{\mathbb{M}}_{ij} = \phi_i\rho_j\mathbf{c}^T\mathbf{E}^{-1}e^{\mathbf{A}\mathbf{E}^{-1}\tau_i}\mathbf{A}e^{\mathbf{E}^{-1}\mathbf{A}t_j}\mathbf{E}^{-1}\mathbf{b} = \phi_i\rho_j h'(\tau_i+t_j), \tag{5.12}$$

where

$$h'(t) = \frac{d}{dt}h(t) = \mathbf{c}^T e^{\mathbf{E}^{-1}\mathbf{A}t}\mathbf{E}^{-1}\mathbf{A}\mathbf{E}^{-1}\mathbf{b}. \tag{5.13}$$

As with the other quantities, $\widetilde{\mathbb{M}}$ can also be obtained from the impulse response samples; yet in this case one will need to measure the derivative of $h'(t)$.

We have just shown that Balanced POD can be obtained directly from sampling the impulse response $h(t)$ and its derivative $h'(t)$ without access to state-snapshot data. A time-domain version of QuadBT follows directly as given in Algorithm 3.

Algorithm 3 Time-domain QuadBT

Input: LTI system described through an impulse response evaluation map, $h(t)$;
quadrature nodes, t_j, and weights, ρ_j, for $j = 1, 2, \ldots, N_p$;
quadrature nodes, τ_k, and weights, ϕ_k, for $k = 1, 2, \ldots, N_q$;
and a truncation index, $1 \le r \le \min(N_p, N_q)$.

Output: Time-domain QuadBT reduced-system: $\widetilde{\mathbf{A}}_r \in \mathbb{R}^{r\times r}$, $\widetilde{\mathbf{b}}_r, \widetilde{\mathbf{c}}_r \in \mathbb{R}^r$.

1: Sample the impulse response $\{h(t_j)\}_{j=1}^{N_p}$ and $\{h(\tau_k)\}_{k=1}^{N_q}$ and its derivative $\{h'(t_j)\}_{j=1}^{N_p}$ and $\{h'(\tau_k)\}_{k=1}^{N_q}$. Using the samples and quadrature weights, $\{\rho_j\}$ and $\{\phi_k\}$, construct $\widetilde{\mathbb{L}} \in \mathbb{C}^{N_q\times N_p}$, $\widetilde{\mathbb{M}} \in \mathbb{C}^{N_q\times N_p}$, $\widetilde{\mathbb{h}}$, and $\widetilde{\mathbf{g}}$ as in (5.7), (5.12), (5.10), and (5.11), respectively,

2: Compute the SVD of $\widetilde{\mathbb{L}}$:

$$\widetilde{\mathbb{L}} = \begin{bmatrix}\widetilde{\mathbf{Z}}_1 & \widetilde{\mathbf{Z}}_2\end{bmatrix}\begin{bmatrix}\widetilde{\mathbf{S}}_1 & \\ & \widetilde{\mathbf{S}}_2\end{bmatrix}\begin{bmatrix}\widetilde{\mathbf{Y}}_1^* \\ \widetilde{\mathbf{Y}}_2^*\end{bmatrix}, \tag{5.14}$$

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

3: Construct the reduced-order matrices:

$$\begin{aligned}&\widetilde{\mathbf{E}}_r = \widetilde{\mathbf{S}}_1^{-1/2}\widetilde{\mathbf{Z}}_1^*\,\widetilde{\mathbb{L}}\,\widetilde{\mathbf{Y}}_1\widetilde{\mathbf{S}}_1^{-1/2} = \mathbf{I}_r, \qquad \widetilde{\mathbf{A}}_r = \widetilde{\mathbf{S}}_1^{-1/2}\widetilde{\mathbf{Z}}_1^*\,\widetilde{\mathbb{M}}\,\widetilde{\mathbf{Y}}_1\widetilde{\mathbf{S}}_1^{-1/2},\\ &\widetilde{\mathbf{b}}_r = \widetilde{\mathbf{S}}_1^{-1/2}\widetilde{\mathbf{Z}}_1^*\widetilde{\mathbb{h}}, \qquad \text{and} \qquad \widetilde{\mathbf{c}}_r = \widetilde{\mathbf{g}}^T\widetilde{\mathbf{Y}}_1\widetilde{\mathbf{S}}_1^{-1/2}.\end{aligned} \tag{5.15}$$

Remark 5.1. Opmeer in [41] established a connection between Balanced POD and interpolatory model reduction when the state data from the simulations of (5.3) and (5.6) are obtained via a numerical ODE solver such as forward and backward Euler and other multistage implicit methods. In [14], Bertram and Faßbender have further expanded on this idea by giving explicit connections between the Butcher tableau of the underlying Runge–Kutta method and the resulting interpolation points.

Remark 5.2. We note that the impulse response measurements, $h(t_i)$ and $h'(t_i)$, which are needed to construct the data matrices in (5.7), (5.10), (5.11), and (5.12), are assumed to be available. In practical scenarios, one would need to rely on the estimation of such values either from experimental data or from numerical simulation. We refer to, e.g., [56] for various impulse response measurement approaches that are in use for acoustical systems.

Remark 5.3. In related current work [46], impulse response sampling of $h(t)$ and $h'(t)$ are employed to extend the eigenvalue realization algorithm [33] to continuous-time dynamical systems.

5.2. Time-domain QuadBT for discrete-time systems. For the (SISO) discrete-time dynamical system

$$\begin{aligned} \mathbf{E}\mathbf{x}_{k+1} &= \mathbf{A}\,\mathbf{x}_k + \mathbf{b}\,u_k, \\ y_k &= \mathbf{c}^T\mathbf{x}_k, \end{aligned} \tag{5.16}$$

the time-domain formulation simplifies drastically due to the uniform discrete time-stepping. The reachability Gramian $\mathbf{P}$ for (5.16), in time domain, is given by

$$\mathbf{P} = \sum_{k=0}^{\infty} (\mathbf{E}^{-1}\mathbf{A})^k \mathbf{E}^{-1}\mathbf{b}\mathbf{b}^T\mathbf{E}^{-T}(\mathbf{A}^T\mathbf{E}^{-T})^k. \tag{5.17}$$

Time-domain Balanced POD for discrete-time systems [36] uses

$$\widetilde{\mathbf{U}} = \begin{bmatrix} \mathbf{E}^{-1}\mathbf{b} & (\mathbf{E}^{-1}\mathbf{A})\mathbf{E}^{-1}\mathbf{b} & \cdots & (\mathbf{E}^{-1}\mathbf{A})^{N_p-1}\mathbf{E}^{-1}\mathbf{b} \end{bmatrix} \in \mathbb{R}^{n\times N_p}$$

as an approximate square-root factor for $\mathbf{P}$, i.e.,

$$\mathbf{P} \approx \widetilde{\mathbf{P}} = \widetilde{\mathbf{U}}\widetilde{\mathbf{U}}^T = \sum_{k=0}^{N_p-1} (\mathbf{E}^{-1}\mathbf{A})^k \mathbf{E}^{-1}\mathbf{b}\mathbf{b}^T\mathbf{E}^{-T}(\mathbf{A}^T\mathbf{E}^{-T})^k.$$

One may follow the same development for $\mathbf{Q}$. For (5.16), $\mathbf{Q}$ is given by

$$\mathbf{Q} = \sum_{k=0}^{\infty} (\mathbf{E}^{-T}\mathbf{A}^T)^k \mathbf{E}^{-T}\mathbf{c}\mathbf{c}^T\mathbf{E}^{-1}(\mathbf{A}\mathbf{E}^{-1})^k. \tag{5.18}$$

Then, an approximate square-root factor for $\mathbf{Q}$ is

$$\widetilde{\mathbf{L}}^T = \begin{bmatrix} \mathbf{c}^T\mathbf{E}^{-1} \\ \mathbf{c}^T\mathbf{E}^{-1}(\mathbf{A}\mathbf{E}^{-1}) \\ \vdots \\ \mathbf{c}^T\mathbf{E}^{-1}(\mathbf{A}\mathbf{E}^{-1})^{N_q-1} \end{bmatrix} \in \mathbb{R}^{N_q\times n}$$

such that

$$\mathbf{Q} \approx \widetilde{\mathbf{Q}} = \widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^T = \sum_{k=0}^{N_q-1} (\mathbf{E}^{-T}\mathbf{A}^T)^k \mathbf{E}^{-T}\mathbf{c}\mathbf{c}^T\mathbf{E}^{-1}(\mathbf{A}\mathbf{E}^{-1})^k.$$

Note that these truncated sums are also the main tool in obtaining low-rank approximations to $\mathbf{P}$ and $\mathbf{Q}$ (see, e.g., [25, 44, 55]). Then, [36] observes that $\widetilde{\mathbb{L}} = \widetilde{\mathbf{L}}^T\mathbf{E}\widetilde{\mathbf{U}}$ can be directly obtained from data: Direct computation shows that

$$\widetilde{\mathbb{L}}_{ij} = \mathbf{c}^T(\mathbf{E}^{-1}\mathbf{A})^{i+j-2}\mathbf{E}^{-1}\mathbf{b}.$$

Thus in time-domain Balanced POD for discrete-time systems, $\widetilde{\mathbb{L}}$ is the Hankel matrix of Markov parameters. Similarly, one may directly show that $\widetilde{\mathbb{M}} = \widetilde{\mathbf{L}}^T\mathbf{E}\widetilde{\mathbf{U}}$ is the shifted Hankel matrix of Markov parameters, i.e.,

$$\widetilde{\mathbb{M}}_{ij} = \mathbf{c}^T(\mathbf{E}^{-1}\mathbf{A})^{i+j-1}\mathbf{E}^{-1}\mathbf{b}.$$

Likewise, $\widetilde{\mathbf{L}}^T\mathbf{b}$ and $\mathbf{c}^T\widetilde{\mathbf{U}}$ can be directly obtained from Markov parameters. Indeed, [36] observed that for the special case of discrete-time dynamical systems, time-domain Balanced POD can be obtained from input-output data using Markov parameters. For discrete-time dynamical systems, the Markov parameters are samples of the impulse response function $h(k) = \mathbf{c}^T(\mathbf{E}^{-1}\mathbf{A})^k\mathbf{E}^{-1}\mathbf{b}$ for $k = 0, 1, 2, \ldots$. (Note that [36] also observed that this formulation of time-domain Balanced POD is equivalent to the eigenvalue realization algorithm [33].) Our formulation in section 5.1 generalizes concepts from [36] to continuous-time systems and unifies them under a larger umbrella of impulse-response sampling; our framework in section 5.1 goes farther in removing the requirement of access to state-space quantities.

6. Conclusion. We introduce here a novel reformulation of one of the most important system-theoretic model reduction tools in use today, *balanced truncation*. In its usual formulation, this method requires intrusive access to the original system realization. By contrast, our reformulation requires only system response data, either measured or computed. The central theme driving our development involves approximation of Gramian-related quantities by means of convergent numerical quadratures, permitting results as close as desired to those of classical BT. We demonstrate the application of our approach to a multitude of test cases, for both SISO and MIMO systems, and for both continuous-time and discrete-time systems. We outline time-domain extensions, connecting our approach to existing methods as well as offering promising future themes.

Appendix A: Two numerical quadrature rules. Our reformulation of BT makes central use of numerical quadrature rules, and we choose two particular quadrature rules, the exponential trapezoid rule and a refinement of the Clenshaw–Curtis quadrature rule developed by Boyd, to illustrate the approach. These strategies are effective in approximating matrix-valued integrals which generically appear as $\int_{-\infty}^{\infty}\mathbf{F}(\zeta)d\zeta$, where we assume that, for $\zeta \in \mathbb{R}$, the integrand $\mathbf{F}(\zeta)$ is continuously differentiable, $\mathbf{F}(-\zeta) = \mathbf{F}(\zeta)^T$, and $\lim_{\zeta\to\pm\infty}\zeta^2\mathbf{F}(\zeta) = \mathbf{M} = \mathbf{M}^T \in \mathbb{R}^{n\times n}$ (so that, in particular, $\|\mathbf{F}(\zeta)\| = \mathcal{O}(\frac{1}{\zeta^2})$ as $\zeta \to \pm\infty$). This assumed form covers the Gramian integrals, (2.4) and (2.5). The quadrature rules that we describe below are typical of possible choices though certainly there will be many other refined approaches that one may consider (see especially [59]).

[ExpTrap]: The exponential trapezoid rule. Note first that

$$\int_{-\infty}^{\infty}\mathbf{F}(\zeta)d\zeta = \int_0^{\infty}\mathbf{F}(\zeta) + \mathbf{F}(\zeta)^T\,d\zeta = L\int_{-\infty}^{\infty}e^{L\tau}\left(\mathbf{F}(e^{L\tau}) + \mathbf{F}(e^{L\tau})^T\right)\,d\tau = \int_{-\infty}^{\infty}\mathbf{G}(\tau)\,d\tau,$$

where $\mathbf{G}(\tau) = L\,e^{L\tau}\left(\mathbf{F}(e^{L\tau}) + \mathbf{F}(e^{L\tau})^T\right)$; the first equality has used the symmetry of $\mathbf{F}(\cdot)$, and the second equality has introduced a change of variable for $\zeta > 0$ as $\zeta = e^{L\tau}$. For any $h > 0$ and $t \in \mathbb{R}$, $\sum_{k=-\infty}^{\infty}\mathbf{G}(t + khL)$, viewed as a function of t, is well-defined, continuously differentiable, and hL-periodic. In particular, it has a pointwise convergent Fourier series with (matrix-valued) coefficients:

$$\sum_{k=-\infty}^{\infty}\mathbf{G}(t + khL) = \frac{2\pi}{hL}\sum_{\ell=0}^{\infty}\widehat{\mathbf{G}}\left(\frac{2\pi\ell}{hL}\right)\,e^{\dot{\imath}\,\frac{2\pi\ell}{hL}t},$$

where $\widehat{\mathbf{G}}(\omega)$ denotes the Fourier transform of $\mathbf{G}(t)$. Evaluating at $t = 0$, this expression may be rearranged to

$$hL \sum_{k=-\infty}^{\infty} \mathbf{G}(khL) - \int_{-\infty}^{\infty} \mathbf{G}(\tau)\, d\tau = 2\pi \sum_{\ell=1}^{\infty} \widehat{\mathbf{G}}\left(\frac{2\pi\ell}{hL}\right)$$

or equivalently,

$$hL \sum_{k=-\infty}^{\infty} e^{khL} \left(\mathbf{F}(e^{khL}) + \mathbf{F}(e^{khL})^T\right) - \int_{-\infty}^{\infty} \mathbf{F}(\zeta)\, d\tau = 2\pi \sum_{\ell=1}^{\infty} \widehat{\mathbf{G}}\left(\frac{2\pi\ell}{hL}\right). \tag{A.1}$$

The right-hand side becomes exponentially close to zero as $h \to 0$. The first sum on the left-hand side manifests the trapezoid rule, which becomes computationally tractable with suitable truncation of the sum: say, $\nu_0 \le k \le \nu_0 + \varpi$. This leads to

$$\int_{-\infty}^{\infty} \mathbf{F}(\zeta) d\zeta \approx hLe^{\nu_0 hL}\mathbf{F}(0) + hLe^{\nu_0 hL} \sum_{\ell=0}^{\varpi} e^{\ell hL} \left(\mathbf{F}(e^{(\nu_0+\ell)hL}) + \mathbf{F}(-e^{(\nu_0+\ell)hL})\right),$$

which includes a correction that weights the value of $\mathbf{F}$ at $\zeta = 0$ as well.

The right-hand side of (A.1) gives the discretization error if the sum on the left-hand side were not truncated. This discretization error is on the order of $e^{-\frac{2\pi a}{hL}}$ as $h \to 0$, where $a \approx \inf |\Re e(\lambda)|$ taken over all poles λ of $\mathbf{A}$ (e.g., see [59, Theorem 5.2]). If we assume for simplicity that the left-hand sum in (A.1) is truncated symmetrically so that $\nu_0 \approx -\frac{\varpi}{2}$ and note that $\|\mathbf{G}(\tau)\| = \mathcal{O}(e^{-L|\tau|})$ as $\tau \to \pm\infty$, then balancing discretization error with truncation error will suggest that the optimal number of quadrature nodes will grow proportionately with $\frac{1}{h^2}$ and the quadrature error will then decrease exponentially fast as the number of quadrature nodes, ϖ, increases: roughly as $\mathcal{O}(e^{-\alpha\sqrt{\varpi}})$ with $\alpha \approx \sqrt{\pi a}$.

[B/CC]: The Boyd/Clenshaw–Curtis rule. Boyd adapted the Clenshaw–Curtis quadrature rule to infinite integration domains in [17]. We summarize the application of this rule to our setting. Starting with the change of variable $\zeta = L\cot(\tau)$, we find

$$\int_{-\infty}^{\infty} \mathbf{F}(\zeta) d\zeta = \int_0^{\pi} \frac{\mathbf{F}(L\cot(\tau))L}{\sin^2(\tau)} d\tau. \tag{A.2}$$

The right-hand integrand is evidently π-periodic. With the asymptotics assumed for $\mathbf{F}$ we have that limits exist for the right-hand integrand both at 0 and π and they are real, symmetric, and equal:

$$\lim_{\tau\to 0} \frac{\mathbf{F}(L\cot(\tau))L}{\sin^2(\tau)} = \frac{1}{L} \lim_{\tau\to 0} \left(L\cot(\tau)\right)^2 \mathbf{F}(L\cot(\tau)) = \frac{1}{L} \lim_{\zeta\to\infty} \zeta^2 \mathbf{F}(\zeta) = \frac{1}{L}\mathbf{M},$$

$$\lim_{\tau\to \pi} \frac{\mathbf{F}(L\cot(\tau))L}{\sin^2(\tau)} = \frac{1}{L} \lim_{\tau\to \pi} \left(L\cot(\tau)\right)^2 \mathbf{F}(L\cot(\tau)) = \frac{1}{L} \lim_{\zeta\to-\infty} \zeta^2 \mathbf{F}(\zeta) = \frac{1}{L}\mathbf{M}.$$

The trapezoid rule for the right-hand integral in (A.2) is a compelling choice. Defining $h = \frac{\pi}{\varpi+1}$ for $\varpi \in \mathbb{N}$, take equally spaced points $\tau_\ell = \ell h$ for $\ell = 0, 1, \ldots, \varpi + 1$, giving

$$\int_{-\infty}^{\infty} \mathbf{F}(\zeta) d\zeta \approx \sum_{\ell=1}^{\varpi} hL \frac{\mathbf{F}(L\cot(\tau_\ell))}{\sin^2(\tau_\ell)} + \frac{h}{L}\mathbf{M} = \sum_{\ell=1}^{\varpi} \rho_\ell^2 \mathbf{F}(\omega_\ell) + \rho_\infty^2 \mathbf{M}, \tag{A.3}$$

where $\rho_\ell^2 = \frac{L\pi}{(\varpi+1)\sin^2(\tau_\ell)}$ and $\omega_\ell = L\cot(\tau_\ell))$ for $\ell = 1, \ldots, \varpi$ and $\rho_\infty^2 = \frac{\pi}{L(\varpi+1)}$. The last term in (A.3) is $\frac{h}{2}$ multiplying the combined evaluation of the periodicized integrand at the endpoints, 0 and π, corresponding to the "nodes at infinity" for the original integral. We note that a similar derivation was described in [20, section 3.3] for the computation of the $\mathcal{H}_2$ norm for linear dynamical systems. We refer the reader to [16] and [21] for details guiding the choice of L in practice.

Appendix B: Proof of Proposition 3.2. The hypotheses imply that

$$\|\mathbf{Q} - \widetilde{\mathbf{Q}}\|_2 < \sigma_{\mathsf{min}}(\mathbf{Q}) \quad \text{and} \quad \|\mathbf{P} - \widetilde{\mathbf{P}}\|_2 < \sigma_{\mathsf{min}}(\mathbf{P}).$$

We then use results from [58, Theorem 1.4] (see also [27, Theorem 10.8]) to build isometries $\mathbf{\Psi}_p \in \mathbb{C}^{N_p \times n}$ and $\mathbf{\Psi}_q \in \mathbb{C}^{N_q \times n}$ and perturbations $\Delta\mathbf{L}$, $\Delta\mathbf{U} \in \mathbb{C}^{n\times n}$ such that

$$\widetilde{\mathbf{U}}^* = \mathbf{\Psi}_p(\mathbf{U} + \Delta\mathbf{U})^T \quad \text{and} \quad \widetilde{\mathbf{L}}^* = \mathbf{\Psi}_q(\mathbf{L} + \Delta\mathbf{L})^T.$$

Defining $\varepsilon_q = \frac{\|\mathbf{Q}-\widetilde{\mathbf{Q}}\|_F}{\|\mathbf{Q}\|_2}$ and $\varepsilon_p = \frac{\|\mathbf{P}-\widetilde{\mathbf{P}}\|_F}{\|\mathbf{P}\|_2}$, the hypotheses ensure that both

$$\mathsf{cond}(\mathbf{Q})\varepsilon_q \le \frac{\delta}{1+\delta} < 1 \quad \text{and} \quad \mathsf{cond}(\mathbf{P})\varepsilon_p \le \frac{\delta}{1+\delta} < 1,$$

and then, referring again to [58, Theorem 1.4], we have further

$$\frac{\|\Delta\mathbf{U}\|_F}{\|\mathbf{U}\|_2} \le \left(\frac{1}{\sqrt{2}}\right)\frac{\mathsf{cond}(\mathbf{P})\varepsilon_p}{1-\mathsf{cond}(\mathbf{P})\varepsilon_p} \le \frac{\delta}{\sqrt{2}} \quad \text{and} \quad \frac{\|\Delta\mathbf{L}\|_F}{\|\mathbf{L}\|_2} \le \left(\frac{1}{\sqrt{2}}\right)\frac{\mathsf{cond}(\mathbf{Q})\varepsilon_q}{1-\mathsf{cond}(\mathbf{Q})\varepsilon_q} \le \frac{\delta}{\sqrt{2}}.$$

Now observe

$$\begin{aligned}\widetilde{\mathbf{L}}^*\mathbf{E}\widetilde{\mathbf{U}} &= \mathbf{\Psi}_q(\mathbf{L}+\Delta\mathbf{L})^T\mathbf{E}(\mathbf{U}+\Delta\mathbf{U})\mathbf{\Psi}_p^T \\ &= \mathbf{\Psi}_q(\mathbf{L}^T\mathbf{E}\mathbf{U})\mathbf{\Psi}_p^T + \mathbf{\Psi}_q\,(\Delta\mathbf{L})^T(\mathbf{E}\mathbf{U})\mathbf{\Psi}_p^T + \mathbf{\Psi}_q(\mathbf{L}^T\mathbf{E})(\Delta\mathbf{U})\,\mathbf{\Psi}_p^T + \mathbf{\Psi}_q\,(\Delta\mathbf{L})^T\,\mathbf{E}\,(\Delta\mathbf{U})\,\mathbf{\Psi}_p^T\end{aligned}$$

and as a result,

$$\begin{aligned}\|\widetilde{\mathbf{L}}^*\mathbf{E}\widetilde{\mathbf{U}} - \mathbf{\Psi}_q(\mathbf{L}^T\mathbf{E}\mathbf{U})\mathbf{\Psi}_p^T\|_F &\le \|\mathbf{\Psi}_q\,\Delta\mathbf{L}^T(\mathbf{E}\mathbf{U})\mathbf{\Psi}_p^T\|_F \quad + \\ &\qquad \|\mathbf{\Psi}_q(\mathbf{L}^T\mathbf{E})(\Delta\mathbf{U})\,\mathbf{\Psi}_p^T\|_F + \|\mathbf{\Psi}_q\,(\Delta\mathbf{L})^T\,\mathbf{E}\,(\Delta\mathbf{U})\,\mathbf{\Psi}_p^T\|_F \\ &\le \|\Delta\mathbf{L}\|_F\,\|\mathbf{E}\mathbf{U}\|_2 + \|\mathbf{L}^T\mathbf{E}\|_2\,\|\Delta\mathbf{U}\|_F + \|\Delta\mathbf{L}\|_F\,\|\mathbf{E}\|_2\|\Delta\mathbf{U}\|_F \\ &\le \frac{\delta}{\sqrt{2}}\,\|\mathbf{L}\|_2\,\|\mathbf{E}\mathbf{U}\|_2 + \frac{\delta}{\sqrt{2}}\,\|\mathbf{L}^T\mathbf{E}\|_2\,\|\mathbf{U}\|_2 + \left(\frac{\delta}{\sqrt{2}}\right)^2\|\mathbf{L}\|_2\,\|\mathbf{E}\|_2\|\mathbf{U}\|_2 \\ &\le \left(\sqrt{2}\,\delta + \frac{\delta^2}{2}\right)\|\mathbf{L}\|_2\,\|\mathbf{E}\|_2\,\|\mathbf{U}\|_2 < 2\,\delta\,\|\mathbf{L}\|_2\,\|\mathbf{E}\|_2\,\|\mathbf{U}\|_2,\end{aligned}$$

where the last inequality takes into account that $0 < \delta < 1$.

For the final assertion, we note that the singular values of $\mathbf{\Psi}_q(\mathbf{L}^T\mathbf{E}\mathbf{U})\mathbf{\Psi}_p^*$ are $\{\sigma_1, \sigma_2, \cdots, \sigma_n\}$ augmented with 0. A consequence of Lidskii's majorization theorem (e.g., [15, Theorem III.4.4]) gives

$$\sum_{k=1}^{n}(\widetilde{\sigma}_k - \sigma_k)^2 \le \|\widetilde{\mathbf{L}}^*\mathbf{E}\widetilde{\mathbf{U}} - \mathbf{\Psi}_q(\mathbf{L}^T\mathbf{E}\mathbf{U})\mathbf{\Psi}_p^*\|_F^2.$$

Then (3.7), the first assertion of Proposition 3.2, gives the conclusion. ∎

REFERENCES

[1] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, SIAM, Philadelphia, 2005.

[2] A. C. Antoulas, C. Beattie, and S. Gugercin, *Interpolatory Methods for Model Reduction*, Comput. Sci. Engrg. 21, SIAM, Philadelphia, 2020.

[3] A. C. Antoulas, I. V. Gosea, and A. C. Ionita, *Model reduction of bilinear systems in the Loewner framework*, SIAM J. Sci. Comput., 38 (2016), pp. B889–B916.

[4] A. C. Antoulas, S. Lefteriu, and A. C. Ionita, *A tutorial introduction to the Loewner framework for model reduction*, in Model Reduction and Approximation, SIAM, Philadelphia, 2017, pp. 335–376.

[5] U. Baur, P. Benner, and L. Feng, *Model order reduction for linear and nonlinear systems: A system-theoretic perspective*, Arch. Comput. Methods Eng., 21 (2014), pp. 331–358.

[6] C. Beattie and S. Gugercin, *Realization-independent $\mathcal{H}_2$-approximation*, in Proceedings of th 51st IEEE Conference on Decision and Control (CDC), IEEE, 2012, pp. 4953–4958.

[7] P. Benner and T. Damm, *Lyapunov equations, energy functionals and model order reduction of bilinear and stochastic systems*, SIAM J. Control Optim., 49 (2011), pp. 686–711.

[8] P. Benner and P. Goyal, *Balanced Truncation Model Order Reduction for Quadratic-Bilinear Control Systems*, preprint, arXiv:1705.00160 [math.OC], 2017, https://arxiv.org/abs/1705.00160.

[9] P. Benner, P. Kürschner, and J. Saak, *Self-generating and efficient shift parameters in ADI methods for large Lyapunov and Sylvester equations*, Electron. Trans. Numer. Anal., 43 (2014), pp. 142–162.

[10] P. Benner, M. Ohlberger, A. Cohen, and K. Willcox, *Model Reduction and Approximation*, SIAM, Philadelphia, 2017.

[11] P. Benner and J. Saak, *Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: A state of the art survey*, GAMM-Mitt., 36 (2013), pp. 32–52.

[12] 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–MTNS, vol. 5, 2010.

[13] C. Bertram and H. Faßbender, *Lyapunov and Sylvester Equations: A Quadrature Framework*, preprint, arXiv:1903.05383 [math.NA], 2020, https://arxiv.org/abs/1903.05383.

[14] C. Bertram and H. Faßbender, *A link between Gramian-based model order reduction and moment matching*, in Model Reduction of Complex Dynamical Systems, Springer, Cham, 2021, pp. 119–139.

[15] R. Bhatia, *Matrix Analysis*, Springer, Cham, 2013.

[16] J. P. Boyd, *The optimization of convergence for Chebyshev polynomial methods in an unbounded domain*, J. Comput. Phys., 45 (1982), pp. 43–79.

[17] J. P. Boyd, *Exponentially convergent Fourier-Chebshev quadrature schemes on bounded and infinite intervals*, J. Sci. Comput, 2 (1987), pp. 99–109.

[18] T. Breiten, *Structure-preserving model reduction for integro-differential equations*, SIAM J. Control Optim., 54 (2016), pp. 2992–3015.

[19] R. F. Curtain and K. Glover, *Balanced realisations for infinite dimensional systems*, Oper. Theory Adv. Appl., 19 (1986), pp. 87–104.

[20] Z. Drmač, S. Gugercin, and C. Beattie, *Quadrature-based vector fitting for discretized $\mathcal{H}_2$ approximation*, SIAM J. Sci. Comput., 37 (2015), pp. A625–A652.

[21] Z. Drmač, S. Gugercin, and C. Beattie, *Vector fitting for matrix-valued rational approximation*, SIAM J. Sci. Comput., 37 (2015), pp. A2346–A2379.

[22] V. Druskin, L. Knizhnerman, and V. Simoncini, *Analysis of the rational Krylov subspace and ADI methods for solving the Lyapunov equation*, SIAM J. Numer. Anal., 49 (2011), pp. 1875–1898.

[23] I. V. Gosea and A. C. Antoulas, *Data-driven model order reduction of quadratic-bilinear systems*, Numer. Linear Algebra Appl., 25 (2018), e2200.

[24] I. V. Gosea, M. Petreczky, A. C. Antoulas, and C. Fiter, *Balanced truncation for linear switched systems*, Adv. Comput. Math., 44 (2018), pp. 1845–1886.

[25] S. Gugercin, D. C. Sorensen, and A. C. Antoulas, *A modified low-rank Smith method for large-scale Lyapunov equations*, Numer. Algorithms, 32 (2003), pp. 27–55.

[26] N. Halko, P. G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288.

[27] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 2002.

[28] C. Himpe, *emgr—The empirical Gramian framework*, Algorithms, 11 (2018), p. 91.

[29] C. Himpe, *Comparing (empirical-Gramian-based) model order reduction algorithms*, in Model Reduction of Complex Dynamical Systems, P. Benner, T. Breiten, H. Faßbender, M. Hinze, T. Stykel, and R. Zimmermann, eds., Springer, Cham, 2021.
[30] A. C. Ionita and A. C. Antoulas, *Data-driven parametrized model reduction in the Loewner framework*, SIAM J. Sci. Comput., 36 (2014), pp. A984–A1007.
[31] J. Juang, *Applied System Identification*, Prentice Hall, Englewood Cliffs, NJ, 1994.
[32] Keysight Technologies, *S-Parameter Measurements: Basics for High Speed Digital Engineers*, Keysight Technologies, Santa Rosa, CA, http://literature.cdn.keysight.com/litweb/pdf/5991-3736EN.pdf.
[33] S. Kung, *A new identification and model reduction algorithm via singular value decomposition*, in Proceedings of 12th Asilomar Conference on Circuits, Systems & Computers, Pacific Grove, CA, 1978, pp. 705–714.
[34] P. Kürschner, *Efficient Low-Rank Solution of Large-Scale Matrix Equations*, PhD thesis, Otto von Guericke Universität Magdeburg, 2016.
[35] J.-R. Li and J. White, *Low rank solution of Lyapunov equations*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 260–280.
[36] Z. Ma, S. Ahuja, and C. W. Rowley, *Reduced-order models for control of fluids using the eigensystem realization algorithm*, Theoret. Comput. Fluid Dyn., 25 (2011), pp. 233–247.
[37] A. Mayo and A. Antoulas, *A framework for the solution of the generalized realization problem*, Linear Algebra Appl., 425 (2007), pp. 634–662.
[38] B. Moore, *Principal component analysis in linear systems: Controllability, observability, and model reduction*, Trans. Automat. Control, 26 (1981), pp. 17–32.
[39] C. Mullis and R. Roberts, *Synthesis of minimum roundoff noise fixed point digital filters*, IEEE Trans. Circuits Syst., 23 (1976), pp. 551–562.
[40] NICONET, *Benchmark Examples for Model Reduction of Linear Time Invariant Dynamical Systems*, http://slicot.org/20-site/126-benchmark-examples-for-model-reduction.
[41] M. R. Opmeer, *Model order reduction by balanced proper orthogonal decomposition and by rational interpolation*, IEEE Trans. Automat. Control, 57 (2011), pp. 472–477.
[42] B. Peherstorfer, S. Gugercin, and K. Willcox, *Data-driven reduced model construction with time-domain Loewner models*, SIAM J. Sci. Comput., 39 (2017), pp. A2152–A2178.
[43] T. Penzl, *Numerical solution of generalized Lyapunov equations*, Adv. Comput. Math., 8 (1998), pp. 33–48.
[44] T. Penzl, *A cyclic low-rank smith method for large sparse Lyapunov equations*, SIAM J. Sci. Comput., 21 (1999), pp. 1401–1418.
[45] J. R. Phillips and L. M. Silveira, *Poor man's TBR: A simple model reduction scheme*, IEEE Trans. Comput. Aided Design Integr. Circuits Syst., 24 (2004), pp. 43–55.
[46] I. Pontes Duff, *private communication*, 2021.
[47] A. Quarteroni, A. Manzoni, and F. Negri, *Reduced Basis Methods for Partial Differential Equations: An Introduction*, Springer, Cham, 2015.
[48] T. Reis and T. Selig, *Balancing transformations for infinite-dimensional systems with nuclear Hankel operator*, Integral Equations Operator Theory, 79 (2014), pp. 67–105.
[49] C. W. Rowley, *Model reduction for fluids, using balanced proper orthogonal decomposition*, Internat. J. Bifur. Chaos, 15 (2005), pp. 997–1013.
[50] J. Sabino, *Solution of Large-Scale Lyapunov Equations via the Block Modified Smith Method*, PhD thesis, Rice University, 2007.
[51] J. M. A. Scherpen, *Balancing for nonlinear systems*, Systems Control Lett., 21 (1993), pp. 143–153.
[52] P. Schulze, B. Unger, C. A. Beattie, and S. Gugercin, *Data-driven structured realization*, Linear Algebra Appl., 537 (2018), pp. 250–286.
[53] V. Simoncini, *Computational methods for linear matrix equations*, SIAM Rev., 58 (2016), pp. 377–441.
[54] J. R. Singler and B. A. Batten, *A proper orthogonal decomposition approach to approximate balanced truncation of infinite dimensional linear systems*, Int. J. Comput. Math., 86 (2009), pp. 355–371.
[55] R. Smith, *Matrix equation* $\mathbf{XA} + \mathbf{BX} = \mathbf{C}$, SIAM J. Appl. Math., 16 (1968), pp. 198–201.
[56] G.-B. V. Stan, J. J. Embrechts, and D. Archambeau, *Comparison of different impulse response measurement techniques*, J. Audio Engrg. Soc., 50 (2002), pp. 249–262.
[57] T. Stykel, *Low-rank iterative methods for projected generalized Lyapunov equations*, Electron. Trans. Numer. Anal., 30 (2008), pp. 187–202.
[58] J.-G. Sun, *Perturbation bounds for the Cholesky and QR factorizations*, BIT, 31 (1991), pp. 341–352.

[59] L. N. Trefethen and J. Weideman, *The exponentially convergent trapezoidal rule*, SIAM Rev., 56 (2014), pp. 385–458.
[60] J. Weideman and L. Trefethen, *Parabolic and hyperbolic contours for computing the Bromwich integral*, Math. Comp., 76 (2007), pp. 1341–1356.
[61] K. Willcox and J. Peraire, *Balanced model reduction via the proper orthogonal decomposition*, AIAA J., 40 (2002), pp. 2323–2330.
[62] S. Yang and M. S. Allen, *Harmonic transfer function to measure translational and rotational velocities with continuous-scan laser Doppler vibrometry*, J. Vib. Acoust., 136 (2014), p. 021025.