ESAIM: M2AN 55 (2021) 561–594
https://doi.org/10.1051/m2an/2021001

**ESAIM: Mathematical Modelling and Numerical Analysis**
www.esaim-m2an.org

# ON ERROR ESTIMATION FOR REDUCED-ORDER MODELING OF LINEAR NON-PARAMETRIC AND PARAMETRIC SYSTEMS

Lihong Feng[1,*] and Peter Benner[1,2]

**Abstract.** Motivated by a recently proposed error estimator for the transfer function of the reduced-order model of a given linear dynamical system, we further develop more theoretical results in this work. Moreover, we propose several variants of the error estimator, and compare those variants with the existing ones both theoretically and numerically. It is shown that some of the proposed error estimators perform better than or equally well as the existing ones. All the error estimators considered can be easily extended to estimate the output error of reduced-order modeling for steady linear parametric systems.

**Mathematics Subject Classification.** 37M05, 65P99, 65L80.

Received April 1, 2020. Accepted January 12, 2021.

## 1. Introduction

Many model order reduction (MOR) methods have been proposed during the last decades [1–3, 5, 7–12]. For many problems, especially parametric time-dependent problems, efficient error estimation of the reduced-order model (ROM) is still critical.

It is well-known that many a-posteriori error bounds/estimators [17, 22, 26, 30, 31] need compute the inf-sup constant, which appears as the denominator of the error estimator. In the numerically discretized space, the inf-sup constant corresponds to the smallest singular value of a large matrix. For many models arising from, *e.g.*, circuit simulation or Micro-Electro-Mechanical Systems (MEMS) simulation, the smallest singular value can be zero at some samples of the parameter due to resonances [21], making the error bound unavailable at those samples. Besides, computing the smallest singular value at many samples of the parameter is time-consuming for large-scale problems. Although some algorithms are proposed to compute a lower bound of the inf-sup constant [24], they are found to be inefficient for many problems [29]. The error bound often overestimates the true error, especially for those systems whose smallest singular values are close to zero at many parameter values. Often, the residual of the solution to the ROM is simply used to form a heuristic error indicator, see, *e.g.*, [13, 14, 18, 25] to name a few.

In recent work [16], a new estimator for the transfer function error, as well as for the output error of MOR for steady parametric systems, is proposed. The proposed error estimator avoids computing the singular values of

*Keywords and phrases.* Model order reduction, error estimation.

[1] Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstrasse 1, D-39106 Magdeburg, Germany.

[2] Fakultät für Mathematik, Otto-von-Guericke Universität Magdeburg, Magdeburg, Germany.

*Corresponding author: feng@mpi-magdeburg.mpg.de

Article published by EDP Sciences

any matrix, and depends mainly on the ROM. It is applicable to any system whose ROMs are computed using a projection based MOR method. It is illustrated by the numerical results that the error estimator is much sharper than the error bound in [17] for those systems with small inf-sup constants. Using the proposed error estimation, the adaptive greedy algorithm converges much faster than using the error bound from [17]. The error estimator was used in the greedy algorithm to select expansion/sampling points for MOR in frequency domain. Searching good expansion/sampling points can also be tackled by optimization with respect to system-theoretic measures, such as the $\mathcal{H}_2$-norm and $\mathcal{H}_2 \times \mathcal{L}_2$-norm, see, *e.g.*, [6, 19, 23].

Error estimation based on randomized residual for parametric steady systems is proposed in [29]. The output error estimation proposed there is also free of computing the inf-sup constant and can be used to estimate the transfer function error in frequency domain. We will show in this work that the error estimator in [29] more likely underestimates the true error as compared with the error estimators in [16] and the proposed error estimators.

Another error estimation, which is independent of the inf-sup constant, is proposed in [20]. This error estimation is used to estimate the error of the state (solution vector). It simply uses the error between two approximate solutions computed from two ROMs divided by a saturation constant as the error estimator. As for estimation of the transfer function error or output error, trivially multiplying the output matrix norm $\|C(\mu)\|$ ($C(\mu)$ will be defined precisely in the next section) with the error estimator could also estimate the output error, but may lead to slow error decay if $\|C(\mu)\|$ is large. Moreover, a saturation constant needs to be estimated for the error estimator in [20], which needs extra computations and may cause inefficiency of the error estimator if computed without sufficient accuracy.

The error bound in [28] is proposed for nonlinear systems and also requires the computation of the inf-sup constant or its lower bound. Numerical issues concerning computing these quantities remain. Moreover, some assumptions on the magnitude of the inf-sup constant is needed in order to derive the error estimator. For problems whose inf-sup constants are close to zero, *e.g.* $O(10^{-12})$, as for the examples presented in this work, the error bound might not be tight anymore. From Lemma 2 in [28], it is not difficult to check that for linear problems, the error bound in [28] is an upper bound of the error estimator $\Delta_1^{pr}$ (will be detailed in Sect. 4.2) proposed in this work when the output matrix satisfies $C(\mu) = I$, the identity matrix. The residual system needed for computing $\Delta_1^{pr}$ is called *error equation* in [28], where a ROM of the error equation needs also to be constructed. This ROM is constructed by running a *separate* greedy algorithm at *each* iteration of the main greedy algorithm. In contrast, we simultaneously construct the ROM of the residual system (error equation) and that of the original system in one greedy algorithm.

In this work, we further explore the property of the error estimator in [16] and propose some variants of it. Sensitivity analyses are presented to show that the proposed error estimators may behave as error bounds when influenced with a small constant. The proposed variants are theoretically and numerically analyzed, and compared with the existing ones. Furthermore, the more general MOR framework based on Petrov–Galerkin projection is used to analyze the error estimators and to explore the corresponding theoretical and numerical properties. In the next section, we first review the error estimator in [16] and develop more theoretical results. Variants of the error estimator and corresponding theoretical analyses are provided in Section 4. Section 5 theoretically compares the new error estimators with the existing ones. Section 6 provides greedy algorithms for constructing the ROMs based on the error estimators. Numerical results of all the error estimators for various problems are presented and compared in Section 7. Conclusions are drawn in the end.

## 2. Preliminaries and notation

We use the following form of a linear system (with or without parameters) as an example to introduce projection based MOR,

$$\begin{aligned} E(\mu)\tfrac{\mathrm{d}}{\mathrm{d}t}x(t,\mu) &= A(\mu)x(t,\mu) + B(\mu)u(t), \\ y(t,\mu) &= C(\mu)x(t,\mu). \end{aligned} \tag{2.1}$$

Here $x(t,\mu) \in \mathbb{R}^n$ is the state vector, $n$ is often referred to as the *order* of the system. The vector $\mu := (\mu_1, \dots, \mu_m) \in \mathbb{R}^{1\times m}$ includes all of the geometrical and physical parameters. The system matrices $E(\mu), A(\mu) \in \mathbb{R}^{n\times n}$, and $B(\mu) \in \mathbb{R}^{n\times n_I}$, $C(\mu) \in \mathbb{R}^{n_O\times n}$ may depend on the parameters.

The reduced-order model (ROM) of the original system can be obtained *via* Petrov–Galerkin projection and can be written as

$$\begin{aligned} \hat{E}(\mu)\tfrac{\mathrm{d}}{\mathrm{d}t}z(t,\mu) &= \hat{A}(\mu)z(t,\mu) + \hat{B}(\mu)u(t), \\ \hat{y}(t,\mu) &= \hat{C}(\mu)z(t,\mu), \end{aligned} \tag{2.2}$$

where $\hat{E}(\mu) = W^T E(\mu)V \in \mathbb{R}^{r\times r}$, $\hat{A}(\mu) = W^T A(\mu)V \in \mathbb{R}^{r\times r}$, $\hat{B}(\mu) = W^T B(\tilde{\mu}) \in \mathbb{R}^{r\times n_I}$, $\hat{C}(\mu) = C(\tilde{\mu})V \in \mathbb{R}^{n_O\times r}$, and $z(t,\mu) \in \mathbb{R}^r$ with $r \ll n$. Then $x(t,\mu)$ can be recovered by $x(t,\mu) \approx Vz(t,\mu)$. Here, range$(V)$ and range$(W)$ are the trial space and the test space for Petrov–Galerkin projection, respectively.

The transfer function of the original system is defined as

$$H(\tilde{\mu}) = C(\mu)Q(\tilde{\mu})^{-1}B(\mu), \tag{2.3}$$

where $Q(\tilde{\mu}) = sE(\mu) - A(\mu)$ and $\tilde{\mu} := (\mu, s)$ with $s \in \mathbb{C}$, the Laplace variable in frequency domain. Similarly, the transfer function of the ROM is

$$\hat{H}(\tilde{\mu}) = \hat{C}(\mu)\hat{Q}(\tilde{\mu})^{-1}\hat{B}(\mu),$$

where $\hat{Q}(\tilde{\mu}) = s\hat{E}(\mu) - \hat{A}(\mu)$.

Note that the error estimators discussed in this work does not require that the transfer function must be derived from the linear system given in (2.1). In fact, the proposed error estimators apply to any system with transfer function in the form of $H(\tilde{\mu}) = C(\tilde{\mu})Q(\tilde{\mu})^{-1}B(\tilde{\mu})$[4, 5]. For example, the parametric model in Section 7 has a transfer function $H(\tilde{\mu}) = CQ(\tilde{\mu})^{-1}B$, with $Q(\tilde{\mu}) = s^2M(\mu) + sD(\mu) + T(\mu)$ being derived from a second-order parametric system. Time-delay systems have transfer functions with $Q(\tilde{\mu}) = Q(s)$ being a non-rational matrix function of $s$. Many systems in computational electromagnetics need to compute output quantities $y(\tilde{\mu}) = C(\tilde{\mu})Q(\tilde{\mu})^{-1}B(\tilde{\mu})$, where $\tilde{\mu} = (f_1(s), \dots, f_l(s))^T$ and $f_i(s), i = 1, \dots, l$, are (rational) functions of $s$. The error of the reduced output $\hat{y}(\tilde{\mu})$ can also be estimated by the proposed error estimators. Therefore, in the following parts of the paper, our discussions are based on the transfer function in the general form $H(\tilde{\mu}) = C(\tilde{\mu})Q(\tilde{\mu})^{-1}B(\tilde{\mu})$. The corresponding reduced transfer function is given as $\hat{H}(\tilde{\mu}) = \hat{C}(\tilde{\mu})\hat{Q}(\tilde{\mu})^{-1}\hat{B}(\tilde{\mu})$.

We define a primal system and a dual system, whose solutions depict the right part $Q(\tilde{\mu})^{-1}B(\tilde{\mu})$ and the left part $C(\tilde{\mu})Q(\tilde{\mu})^{-1}$ of the transfer function $H(\tilde{\mu})$, respectively. A primal system in frequency domain is defined as

$$Q(\tilde{\mu})x_{pr}(\tilde{\mu}) = B(\tilde{\mu}). \tag{2.4}$$

The reduced primal system is then defined as

$$\hat{Q}(\tilde{\mu})z_{pr}(\tilde{\mu}) = \hat{B}(\tilde{\mu}), \tag{2.5}$$

where $\hat{Q}(\tilde{\mu}) = W^T Q(\tilde{\mu})V$, which is $\hat{Q}(\tilde{\mu}) = s\hat{E}(\mu) - \hat{A}(\mu)$ for the ROM of (2.1). Define a dual system

$$Q^T(\tilde{\mu})x_{du}(\tilde{\mu}) = C^T(\tilde{\mu}), \tag{2.6}$$

where $x_{du}(\tilde{\mu})$ solves the dual system. The ROM of the dual system is

$$\hat{Q}_{du}(\tilde{\mu})z_{du}(\tilde{\mu}) = \hat{C}_{du}(\tilde{\mu}), \tag{2.7}$$

where $\hat{Q}_{du}(\tilde{\mu}) = W_{du}^T Q^T(\tilde{\mu})V_{du}$, $\hat{C}_{du}(\tilde{\mu}) = W_{du}^T C^T(\tilde{\mu})$, such that $\hat{x}_{du}(\tilde{\mu}) := V_{du}z_{du}(\tilde{\mu})$ well approximates $x_{du}(\tilde{\mu})$. Analogously, range$(V_{du})$ and range$(W_{du})$ are the trial space and the test space for Petrov–Galerkin projection of the dual system (2.6), respectively. The ROMs of the primal and the dual systems introduce two residuals, respectively, *i.e.*, the *primal residual*

$$r_{pr}(\tilde{\mu}) = B(\tilde{\mu}) - Q(\tilde{\mu})\hat{x}_{pr}(\tilde{\mu}) \tag{2.8}$$

and the *dual residual*

$$r_{du}(\tilde{\mu}) = C^T(\tilde{\mu}) - Q^T(\tilde{\mu})\hat{x}_{du}(\tilde{\mu}). \tag{2.9}$$

Here, $\hat{x}_{pr}(\tilde{\mu}) := Vz_{pr}(\tilde{\mu})$ is obtained from the reduced primal system (2.5) and approximates $x_{pr}(\tilde{\mu})$. In the following, we first review the error estimator in [16], then develop more theoretical results. Several variants of the error estimator and corresponding theoretical analyses are proposed afterwards. We only consider single-input single-output (SISO) systems. Extension of the error estimator to multiple-input multiple-output (MIMO) systems as well as to output error estimation for steady linear parametric systems is detailed in [16] and will not be repeated in this work. $|\cdot|$ denotes the absolute value of a scalar.

## 3. Error estimator in [16] and extensions

It is not difficult to obtain the following proposition, based on which several error estimators are proposed.

**Proposition 3.1.** *The error of the reduced transfer function $\hat{H}(\tilde{\mu})$ satisfies*

$$|H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| = |x_{du}^T(\tilde{\mu})r_{pr}(\tilde{\mu})|,$$

*where $x_{du}(\tilde{\mu})$ solves the dual system in (2.6) and $r_{pr}(\tilde{\mu})$ is defined in (2.8).*

*Proof.* It follows from the definition of $\hat{H}(\tilde{\mu})$ and the reduced primal system (2.5) that

$$\begin{aligned} |H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| &= |C(\tilde{\mu})(Q^{-1}(\tilde{\mu})B(\tilde{\mu}) - V\hat{Q}^{-1}(\tilde{\mu})\hat{B}(\tilde{\mu}))| \\ &= |C(\tilde{\mu})Q^{-1}(\tilde{\mu})(B(\tilde{\mu}) - Q(\tilde{\mu})\underbrace{V\hat{Q}^{-1}(\tilde{\mu})\hat{B}(\tilde{\mu}))}_{\hat{x}_{pr}(\tilde{\mu}):=Vz_{pr}(\tilde{\mu})|} \\ &= |C(\tilde{\mu})Q^{-1}(\tilde{\mu})r_{pr}(\tilde{\mu})| \\ &= |x_{du}^T(\tilde{\mu})r_{pr}(\tilde{\mu})|. \end{aligned} \tag{3.1}$$

□

Note that computing $x_{du}^T(\tilde{\mu})$ in the last equality of (3.1) needs to solve the dual system of original large scale $n$. If we solve the ROM of the dual system instead, then $x_{du}^T(\tilde{\mu})$ can be approximated by $x_{du}(\tilde{\mu}) \approx \hat{x}_{du}(\tilde{\mu}) = V_{du}z_{du}(\tilde{\mu})$. Consequently, the error of $\hat{H}(\tilde{\mu})$ can be estimated as

$$|H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| \approx \Delta_1(\tilde{\mu}) := |\hat{x}_{du}^T(\tilde{\mu})r_{pr}(\tilde{\mu})|. \tag{3.2}$$

Clearly, the error estimator $\Delta_1(\tilde{\mu})$ might underestimate the true error. To reduce the probability of underestimation, a more robust error estimator is proposed in [16], which is based on the following error bound.

**Theorem 3.2** ([16]). *The error of the reduced transfer function $\hat{H}(\tilde{\mu})$ can be bounded as*

$$|H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| \leq \Delta_1(\tilde{\mu}) + |x_{r_{du}}^T(\tilde{\mu})r_{pr}(\tilde{\mu})|, \tag{3.3}$$

*where $x_{r_{du}}(\tilde{\mu})$ is the solution to the dual-residual system defined as*

$$Q^T(\tilde{\mu})x_{r_{du}}(\tilde{\mu}) = r_{du}(\tilde{\mu}). \tag{3.4}$$

*Proof.* See [16]. □

Again, computing $x_{r_{du}}(\tilde{\mu})$ in (3.3) requires solving a large system in (3.4). Instead, we compute the ROM of (3.4),

$$\tilde{Q}(\tilde{\mu})z_{r_{du}}(\tilde{\mu}) = \tilde{r}_{du}(\tilde{\mu}), \tag{3.5}$$

where $\tilde{Q}(\tilde{\mu}) = W_{r_{du}}^T Q^T(\tilde{\mu})V_{r_{du}}$, $\tilde{r}_{du}(\tilde{\mu}) = W_{r_{du}}^T r_{du}(\tilde{\mu})$. Then $x_{r_{du}}(\tilde{\mu}) \approx \hat{x}_{r_{du}}(\tilde{\mu}) := V_{r_{du}} z_{r_{du}}(\tilde{\mu})$. Finally we replace $x_{r_{du}}(\tilde{\mu})$ in the error bound (3.3) with $\hat{x}_{r_{du}}(\tilde{\mu})$, and get the error estimator:

$$|H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| \lesssim \Delta_1(\tilde{\mu}) + |\hat{x}_{r_{du}}^T(\tilde{\mu})r_{pr}(\tilde{\mu})| =: \Delta_2(\tilde{\mu}).$$

**Theorem 3.3.** *The error of the reduced transfer function $\hat{H}(\tilde{\mu})$ can be bounded as*

$$\Delta_1(\tilde{\mu}) - \varepsilon_1 \leq |H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| \leq \Delta_1(\tilde{\mu}) + \varepsilon_1, \tag{3.6}$$

*where* $\varepsilon_1 := |(x_{du}(\tilde{\mu}) - \hat{x}_{du}(\tilde{\mu}))^T r_{pr}(\tilde{\mu})| \geq 0.$

*Proof.* On the one hand, by Proposition 3.1

$$\begin{aligned} |H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| &= |x_{du}^T(\tilde{\mu})r_{pr}(\tilde{\mu})| + |\hat{x}_{du}^T(\tilde{\mu})r_{pr}(\tilde{\mu})| - |\hat{x}_{du}^T(\tilde{\mu})r_{pr}(\tilde{\mu})| \\ &= \Delta_1\tilde{\mu}) + |x_{du}^T(\tilde{\mu})r_{pr}(\tilde{\mu})| - |\hat{x}_{du}^T(\tilde{\mu})r_{pr}(\tilde{\mu})| \\ &\leq \Delta_1(\tilde{\mu}) + \varepsilon_1. \end{aligned} \tag{3.7}$$

On the other hand,

$$\begin{aligned} \Delta_1(\tilde{\mu}) &= |\hat{x}_{du}^T(\tilde{\mu})r_{pr}(\tilde{\mu})| + |x_{du}^T(\tilde{\mu})r_{pr}(\tilde{\mu})| - |x_{du}^T(\tilde{\mu})r_{pr}(\tilde{\mu})| \\ &= |H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| + |\hat{x}_{du}^T(\tilde{\mu})r_{pr}(\tilde{\mu})| - |x_{du}^T(\tilde{\mu})r_{pr}(\tilde{\mu})| \\ &\leq |H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| + \varepsilon_1. \end{aligned} \tag{3.8}$$

□

Here, $\varepsilon_1$ can be very small if the reduced solution $\hat{x}_{du}(\tilde{\mu})$ approximates the solution $x_{du}(\tilde{\mu})$ to the original dual system well and/or if the primal system solution $x_{pr}(\tilde{\mu})$ is well approximated by the reduced primal solution $\hat{x}_{pr}(\tilde{\mu})$, so that $||r_{pr}(\tilde{\mu})||_2$ is small. Both approximations can be made as accurate as possible through a greedy process using the proposed error estimators, which will be discussed in detail in Section 6. Theorem 3.3 shows that the true error is both lower bounded and upper bounded by $\Delta_1(\tilde{\mu})$ with the influence of a small-valued $\varepsilon_1$.

**Theorem 3.4.** *The error of the reduced transfer function $\hat{H}(\tilde{\mu})$ can be bounded as*

$$\Delta_2(\tilde{\mu}) - \delta_2 - \varepsilon_1 \leq |H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| \leq \Delta_2(\tilde{\mu}) + \varepsilon_2 \tag{3.9}$$

*where* $\varepsilon_2 := |(x_{r_{du}}(\tilde{\mu}) - \hat{x}_{r_{du}}(\tilde{\mu}))^T r_{pr}(\tilde{\mu})| \geq 0$, $\delta_2 := |\hat{x}_{r_{du}}(\tilde{\mu}))^T r_{pr}(\tilde{\mu})|$.

*Proof.* From (3.3),

$$\begin{aligned} |H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| &\leq \Delta_1(\tilde{\mu}) + |x_{r_{du}}^T(\tilde{\mu})r_{pr}(\tilde{\mu})| \\ &= \Delta_1(\tilde{\mu}) + |\hat{x}_{r_{du}}^T(\tilde{\mu})r_{pr}(\tilde{\mu})| - |\hat{x}_{r_{du}}^T(\tilde{\mu})r_{pr}(\tilde{\mu})| + |x_{r_{du}}^T(\tilde{\mu})r_{pr}(\tilde{\mu})| \\ &= \Delta_2(\tilde{\mu}) + |x_{r_{du}}^T(\tilde{\mu})r_{pr}(\tilde{\mu})| - |\hat{x}_{r_{du}}^T(\tilde{\mu})r_{pr}(\tilde{\mu})| \\ &\leq \Delta_2(\tilde{\mu}) + \varepsilon_2. \end{aligned} \tag{3.10}$$

The proof of the lower bound is a direct result from the lower bound of Theorem 3.3 and the relation between $\Delta_1(\tilde{\mu})$ and $\Delta_2(\tilde{\mu})$. □

$\varepsilon_1$ and $\varepsilon_2$ involve computing the solutions $x_{du}(\tilde{\mu})$ and $x_{r_{du}}(\tilde{\mu})$ to the original dual system and the original dual-residual system, respectively. Since the original dual system and the original dual-residual system are of high dimension $n$, computing the lower bounds and upper bounds in (3.3) and (3.4) is as expensive as computing the true error. Instead of the lower and upper bounds, the error estimators $\Delta_1(\tilde{\mu})$, $\Delta_2(\tilde{\mu})$ can be computed and used in practice, as they involve only computations in the reduced $r$ dimensions. Theorem 3.4 shows that the error estimator $\Delta_2(\tilde{\mu})$ cannot underestimate the true error too much, since $\Delta_2(\tilde{\mu}) \geq |H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| - \varepsilon_2$ and $\varepsilon_2$ can be made very small by letting $\hat{x}_{r_{du}}(\tilde{\mu})$ approximate $x_{r_{du}}(\tilde{\mu})$ well. On the other hand, when $\varepsilon_2$ is small, Theorem 3.4 implies that $\Delta_2(\tilde{\mu})$ is a tight error estimator from above. Furthermore, Theorem 3.4 also provides a lower bound for the true error using $\Delta_2(\tilde{\mu})$ and two small valued variables $\varepsilon_2$ and $\delta_2$. Here, $\delta_2$ cannot be large when both $r_{pr}(\tilde{\mu})$ and $r_{du}(\tilde{\mu})$ become small. Note that $r_{du}(\tilde{\mu})$ appears on the right-hand side of the reduced dual-residual system (3.5) from which $\hat{x}^T_{r_{du}}(\tilde{\mu})$ in $\delta_2$ is computed.

## 4. Error estimator variants

In the following, we derive some error estimators, which can be seen as variants of the error estimators $\Delta_1(\tilde{\mu})$ and $\Delta_2(\tilde{\mu})$, respectively. Again, the lower bounds and upper bounds proved in the following theorems are used to show how close the proposed error estimators are to the true error and are not computed in practice. As mentioned at the end of the previous section, computing them is as expensive as computing the true error. This is the main motivation for proposing the error estimators.

### 4.1. Variant 1

From the error bound in (3.3) and (3.4), we get

$$|H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| \leq \Delta_1(\tilde{\mu}) + |r^T_{du}(\tilde{\mu})Q^{-1}(\tilde{\mu})r_{pr}(\tilde{\mu})|. \tag{4.1}$$

We see that instead of solving the dual-residual system (3.4), one can also solve the primal-residual system as below,

$$Q(\tilde{\mu})x_{r_{pr}}(\tilde{\mu}) = r_{pr}(\tilde{\mu}). \tag{4.2}$$

Replacing $Q^{-1}(\tilde{\mu})r_{pr}(\tilde{\mu})$ in (4.1) with $x_{r_{pr}}(\tilde{\mu})$ in (4.2), we obtain

$$|H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| \leq \Delta_1(\tilde{\mu}) + |r^T_{du}(\tilde{\mu})x_{r_{pr}}(\tilde{\mu})|. \tag{4.3}$$

If we construct the ROM of the primal-residual system in (4.2), *i.e.*,

$$W^T_{r_{pr}}Q(\tilde{\mu})V_{r_{pr}}z_{r_{pr}}(\tilde{\mu}) = W^T_{r_{pr}}r_{pr}(\tilde{\mu}), \tag{4.4}$$

then we obtain a variant of $\Delta_2(\tilde{\mu})$,

$$|H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| \lesssim \Delta_1(\tilde{\mu}) + |r^T_{du}(\tilde{\mu})\hat{x}_{r_{pr}}(\tilde{\mu})| =: \Delta_2^{pr}(\tilde{\mu}),$$

where $\hat{x}_{r_{pr}}(\tilde{\mu}) = V_{r_{pr}}z_{r_{pr}}$ is computed from (4.4), the ROM of the primal-residual system and approximates the state vector $x_{r_{pr}}(\tilde{\mu})$ of the primal-residual system. We obtain a similar sensitivity analysis for $\Delta_2^{pr}(\tilde{\mu})$ presented in Theorem 4.1.

**Theorem 4.1.** *The error of the reduced transfer function $\hat{H}(\tilde{\mu})$ can be bounded as*

$$\Delta_2^{pr}(\tilde{\mu}) - \delta_2^{pr} - \varepsilon_1 \leq |H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| \leq \Delta_2^{pr}(\tilde{\mu}) + \varepsilon_2^{pr} \tag{4.5}$$

*where $\varepsilon_2^{pr} := |r^T_{du}(\tilde{\mu})(x_{r_{pr}}(\tilde{\mu}) - \hat{x}_{r_{pr}}(\tilde{\mu}))| \geq 0$ and $\delta_2^{pr} := |r^T_{du}(\tilde{\mu})\hat{x}_{r_{pr}}(\tilde{\mu})|$.*

*Proof.* The result can be obtained by using (4.3) and following similar steps as in the proof of Theorem 3.4. □

Note that $\varepsilon_2^{pr}$ will be of small value once the reduced solution $\hat{x}_{r_{pr}}(\tilde{\mu})$ approximates $x_{r_{pr}}(\tilde{\mu})$, the solution to the primal-residual system (4.2), well.

## 4.2. Variant 2

From (3.1), we know

$$|H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| = |C(\tilde{\mu})Q^{-1}(\tilde{\mu})r_{pr}(\tilde{\mu})|.$$

Similarly, if we use the solution to the primal-residual system (4.2) to replace $Q^{-1}(\tilde{\mu})r_{pr}(\tilde{\mu})$, then we get

$$|H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| = |C(\tilde{\mu})x_{r_{pr}}(\tilde{\mu})|. \tag{4.6}$$

If further using the ROM (4.4) to compute an approximate state, then $x_{r_{pr}}(\tilde{\mu})$ in (4.6) can be approximated by $\hat{x}_{r_{pr}}(\tilde{\mu})$. We obtain the following error estimation

$$|H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| \approx |C(\tilde{\mu})\hat{x}_{r_{pr}}(\tilde{\mu})| =: \Delta_1^{pr}(\tilde{\mu}),$$

that can be considered as a variant of $\Delta_1(\tilde{\mu})$.

**Theorem 4.2.** *The error of the reduced transfer function $\hat{H}(\tilde{\mu})$ can be bounded as*

$$\Delta_1^{pr}(\tilde{\mu}) - \varepsilon_1^{pr} \leq |H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| \leq \Delta_1^{pr}(\tilde{\mu}) + \varepsilon_1^{pr}, \tag{4.7}$$

*where $\varepsilon_1^{pr} := |C(\tilde{\mu})(x_{r_{pr}}(\tilde{\mu}) - \hat{x}_{r_{pr}}(\tilde{\mu}))| \geq 0$.*

*Proof.* The proof is similar to that of Theorem 3.3 and therefore not be repeated here. □

## 4.3. Variant 3

The next theorem presents an error bound based on $\Delta_1^{pr}(\tilde{\mu})$, from which we get another variant of $\Delta_2(\tilde{\mu})$.

**Theorem 4.3.** *The error of the reduced transfer function $\hat{H}(\tilde{\mu})$ can be bounded as*

$$|H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| \leq \Delta_1^{pr}(\tilde{\mu}) + |x_{du}^T(\tilde{\mu})r_{r_{pr}}(\tilde{\mu})|,$$

*where $r_{r_{pr}}$ is the residual of the approximate solution $\hat{x}_{r_{pr}}(\tilde{\mu})$ to the primal-residual system in (4.2), i.e., $r_{r_{pr}} = r_{pr}(\tilde{\mu}) - Q\hat{x}_{r_{pr}}(\tilde{\mu})$.*

*Proof.* From (4.6), the true error can be presented as

$$|H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| = |C(\tilde{\mu})x_{r_{pr}}(\tilde{\mu})|. \tag{4.8}$$

We check the distance between the true error $|C(\tilde{\mu})x_{r_{pr}}(\tilde{\mu})|$ and its estimator $\Delta_1^{pr}(\tilde{\mu}) = |C(\tilde{\mu})\hat{x}_{r_{pr}}(\tilde{\mu})|$:

$$\begin{aligned} |C(\tilde{\mu})x_{r_{pr}}(\tilde{\mu})| - |C(\tilde{\mu})\hat{x}_{r_{pr}}(\tilde{\mu})| &\leq |C(\tilde{\mu})Q^{-1}r_{pr}(\tilde{\mu}) - C(\tilde{\mu})\hat{x}_{r_{pr}}(\tilde{\mu})| \\ &= |C(\tilde{\mu})Q^{-1}\underbrace{[r_{pr}(\tilde{\mu}) - Q\hat{x}_{r_{pr}}(\tilde{\mu})]}_{=:r_{r_{pr}}(\tilde{\mu})}|. \end{aligned} \tag{4.9}$$

Combining (4.8) and (4.9), we get

$$\begin{aligned} |H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| &\leq |C(\tilde{\mu})\hat{x}_{r_{pr}}(\tilde{\mu})| + |C(\tilde{\mu})Q^{-1}r_{r_{pr}}(\tilde{\mu})| \\ &= \Delta_1^{pr}(\tilde{\mu}) + |x_{du}^T(\tilde{\mu})r_{r_{pr}}(\tilde{\mu})|. \end{aligned} \tag{4.10}$$

□

Similarly, we get the following error estimator by approximating $x_{du}(\tilde{\mu})$ with $\hat{x}_{du}(\tilde{\mu})$.

$$|H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| \lesssim \Delta_1^{pr}(\tilde{\mu}) + |\hat{x}_{du}^T(\tilde{\mu})r_{r_{pr}}(\tilde{\mu})| =: \Delta_3(\tilde{\mu}).$$

**Theorem 4.4.** *The error of the reduced transfer function $\hat{H}(\tilde{\mu})$ can be bounded as*

$$\Delta_3(\tilde{\mu}) - \delta_3 - \varepsilon_1^{pr} \leq |H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| \leq \Delta_3(\tilde{\mu}) + \varepsilon_3 \tag{4.11}$$

*where $\varepsilon_3 := |(x_{du}(\tilde{\mu}) - \hat{x}_{du}(\tilde{\mu}))^T r_{pr}(\tilde{\mu})| \geq 0$ and $\delta_3 = |\hat{x}_{du}^T(\tilde{\mu}) r_{r_{pr}}(\tilde{\mu})|$.*

*Proof.* The result can be obtained by using (4.10), the relation between $\Delta_3(\tilde{\mu})$ and $\Delta_1^{pr}(\tilde{\mu})$, and the lower bound of Theorem 4.2, then following similar steps as in the proof of Theorem 3.4. □

Analogously, $\varepsilon_3$ is also a small number, since $\hat{x}_{du}(\tilde{\mu})$ is close enough to $x_{du}(\tilde{\mu})$ if it is a good approximation computed from the ROM of the dual system.

## 4.4. Variant 4

In (4.10), if we consider $Q^{-1} r_{r_{pr}}$ and seek the solution to the primal-residual-residual system,

$$Q(\tilde{\mu}) x_{r_{rpr}}(\tilde{\mu}) = r_{r_{pr}}(\tilde{\mu}), \tag{4.12}$$

then the error bound in (4.10) becomes

$$|H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| \leq \Delta_1^{pr}(\tilde{\mu}) + |C(\tilde{\mu}) x_{r_{rpr}}(\tilde{\mu})|. \tag{4.13}$$

Certainly, we can compute the ROM of (4.12),

$$W_{r_{rpr}}^T Q(\tilde{\mu}) V_{r_{rpr}} z_{r_{rpr}}(\tilde{\mu}) = W_{r_{rpr}}^T r_{r_{pr}}(\tilde{\mu}), \tag{4.14}$$

and replace $x_{r_{rpr}}(\tilde{\mu})$ in (4.13) with its approximation $\hat{x}_{r_{rpr}}(\tilde{\mu}) = V_{r_{rpr}} z_{r_{rpr}}(\tilde{\mu})$ computed from the ROM. Finally, we get the error estimator

$$|H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| \lesssim |\Delta_1^{pr}(\tilde{\mu})| + |C(\tilde{\mu}) \hat{x}_{r_{rpr}}(\tilde{\mu})| =: \Delta_3^{pr}(\tilde{\mu}).$$

From (4.13), we can get the following lower and upper bound using the error estimator $\Delta_3^{pr}(\tilde{\mu})$.

**Theorem 4.5.** *The error of the reduced transfer function $\hat{H}(\tilde{\mu})$ can be bounded as*

$$\Delta_3^{pr}(\tilde{\mu}) - \delta_3^{pr} - \varepsilon_1^{pr} \leq |H(\tilde{\mu}) - \hat{H}(\tilde{\mu})| \leq \Delta_3^{pr}(\tilde{\mu}) + \varepsilon_3^{pr} \tag{4.15}$$

*where $\varepsilon_3^{pr} := |C(\tilde{\mu})(x_{r_{rpr}}(\tilde{\mu}) - \hat{x}_{r_{rpr}}(\tilde{\mu}))| \geq 0$ and $\delta_3^{pr} := |C(\tilde{\mu}) \hat{x}_{r_{rpr}}(\tilde{\mu})|$.*

*Proof.* The result can be obtained by using (4.13) and following similar steps as in the proof of Theorem 4.4. □

## 4.5. Relations among the error estimators

In this section we explore relations among the error estimators discussed in the previous two sections and present the following propositions.

**Proposition 4.6.** *If $W_{du} = V$, and $V_{du} = W$, then $\Delta_1(\tilde{\mu}) = 0$.*

*Proof.*

$$\begin{aligned}\Delta_1(\tilde{\mu}) &= |\hat{x}_{du}^T(\tilde{\mu}) r_{pr}(\tilde{\mu})| \\ &= |\hat{x}_{du}^T(\tilde{\mu})(B(\tilde{\mu}) - Q(\tilde{\mu}) V (W^T Q(\tilde{\mu}) V)^{-1} W^T B(\tilde{\mu})| \\ &= |\hat{x}_{du}^T(\tilde{\mu}) B(\tilde{\mu}) - \hat{x}_{du}^T(\tilde{\mu}) Q(\tilde{\mu}) V (W^T Q(\tilde{\mu}) V)^{-1} W^T B(\tilde{\mu})|.\end{aligned} \tag{4.16}$$

The first part of the last equation in (4.16) is

$$\begin{aligned}\hat{x}_{du}^T(\tilde{\mu})B(\tilde{\mu}) &= [V_{du}(W_{du}^T Q^T(\tilde{\mu})V_{du})^{-1}W_{du}^T C^T(\tilde{\mu})]^T B(\tilde{\mu})\\ &= C(\tilde{\mu})V(W^T Q(\tilde{\mu})V)^{-1}W^T B(\tilde{\mu}) \text{ (if } W_{du}=V \text{ and } V_{du}=W).\end{aligned} \tag{4.17}$$

If $W_{du}=V$ and $V_{du}=W$, the second part of the last equation in (4.16) becomes

$$\begin{aligned}\hat{x}_{du}^T(\tilde{\mu})V(W^T Q(\tilde{\mu})V)^{-1}W^T B(\tilde{\mu}) &= \left[V_{du}(W_{du}^T Q^T(\tilde{\mu})V_{du})^{-1}W_{du}^T C^T(\tilde{\mu})\right]^T Q(\tilde{\mu})V(W^T Q(\tilde{\mu})V)^{-1}W^T B(\tilde{\mu})\\ &= C(\tilde{\mu})V(W^T Q(\tilde{\mu})V)^{-1}W^T Q(\tilde{\mu})V(W^T Q(\tilde{\mu})V)^{-1}W^T B(\tilde{\mu})\\ &= C(\tilde{\mu})V(W^T Q(\tilde{\mu})V)^{-1}W^T B(\tilde{\mu}).\end{aligned} \tag{4.18}$$

Comparing (4.17) and (4.18), we get the conclusion. □

**Remark 4.7.** Proposition 4.6 points out that if $W_{du}=V$ and $V_{du}=W$, then $\Delta_1(\tilde{\mu})$ is always zero, and cannot be a good error estimator. This is not the case for most problems. However, if the system is symmetric, *i.e.*, $Q(\tilde{\mu})=Q^T(\tilde{\mu})$, and $B(\tilde{\mu})=C^T(\tilde{\mu})$, this will likely happen, since in this case, the primal system and the dual system are identical. We will show later that for systems that are almost symmetric, *i.e.*, $Q(\tilde{\mu})\approx Q^T(\tilde{\mu})$ and/or $B(\tilde{\mu})\approx C^T(\tilde{\mu})$, $\Delta_1(\tilde{\mu})$ also behaves badly. One possibility of avoiding $\Delta_1(\tilde{\mu})$ being zero or improving the performance of $\Delta_1(\tilde{\mu})$ is to construct $(W_{du},V_{du})$ and $(W,V)$ from different subspaces of the solution (state) manifold. More specifically, when using time domain methods, different snapshots should be chosen for $(W_{du},V_{du})$ and $(W,V)$, respectively; or different expansion points should be taken if using frequency domain methods, *e.g.*, moment-matching.

**Remark 4.8.** Using Galerkin projection, *i.e.*, $W=V$, $W_{du}=V_{du}$, then $V_{du}=V$ leads to $\Delta_1(\tilde{\mu})=0$.

**Proposition 4.9.** *If $W_{r_{du}}=W_{du}$, then the second part of $\Delta_2(\tilde{\mu})$ is always zero, i.e.,* $|\hat{x}_{r_{du}}^T(\tilde{\mu})r_{pr}(\tilde{\mu})|=0$.

*Proof.*

$$\begin{aligned}\hat{x}_{r_{du}}^T(\tilde{\mu})r_{pr}(\tilde{\mu}) &= [V_{r_{du}}\tilde{Q}^{-1}(\tilde{\mu})(W_{r_{du}}^T r_{du}(\tilde{\mu}))]^T r_{pr}(\tilde{\mu})\\ &= r_{du}^T(\tilde{\mu})W_{du}\tilde{Q}^{-T}(\tilde{\mu})V_{r_{du}}^T r_{pr}(\tilde{\mu}) \text{ (if } W_{r_{du}}=W_{du}).\end{aligned} \tag{4.19}$$

Considering the first two terms in the last equation, we get

$$\begin{aligned}(r_{du}^T(\tilde{\mu})W_{du})^T &= W_{du}^T(C^T(\tilde{\mu})-Q^T(\tilde{\mu})V_{du}z_{du}(\tilde{\mu}))\\ &= 0 \quad \text{(due to (2.7))}.\end{aligned} \tag{4.20}$$

□

**Remark 4.10.** Proposition 4.9 points out that if $W_{r_{du}}=W_{du}$, then $\Delta_2(\tilde{\mu})$ reduces to $\Delta_1(\tilde{\mu})$, and cannot be more robust than $\Delta_1(\tilde{\mu})$. Therefore, $W_{r_{du}}$ should be carefully constructed to avoid being equal to $W_{du}$. For example, it could be computed from different snapshots if the reduced basis method is used for MOR; when (multi-)moment-matching is used for MOR, different expansion points should be chosen for $W_{r_{du}}$ and $W_{du}$, respectively. More details can be found in Section 6. In the case of Galerkin projection, *i.e.*, $W_{r_{du}}=V_{r_{du}}$ and $W_{du}=V_{du}$, $V_{r_{du}}=V_{du}$ leads to the same result in Proposition 4.9.

**Proposition 4.11.** *If $W_{r_{pr}}=W$, then $\hat{x}_{r_{pr}}(\tilde{\mu})=0$.*

*Proof.* From the ROM of the primal-residual system in (4.4),

$$\begin{aligned}\hat{x}_{r_{pr}}(\tilde{\mu}) &= V_{r_{pr}}(W_{r_{pr}}^T Q(\tilde{\mu})V_{r_{pr}})^{-1}(W_{r_{pr}}^T r_{pr}(\tilde{\mu}))\\ &= V_{r_{pr}}(W_{r_{pr}}^T Q(\tilde{\mu})V_{r_{pr}})^{-1}(W^T r_{pr}(\tilde{\mu})) \quad \text{(if } W_{r_{pr}}=W)\\ &= V_{r_{pr}}(W_{r_{pr}}^T Q(\tilde{\mu})V_{r_{pr}})^{-1}W^T(B(\tilde{\mu})-Q(\tilde{\mu})Vz_{pr}(\tilde{\mu}))\\ &= V_{r_{pr}}(W_{r_{pr}}^T Q(\tilde{\mu})V_{r_{pr}})^{-1}[W^T B(\tilde{\mu})-W^T Q(\tilde{\mu})Vz_{pr}(\tilde{\mu})]\\ &= 0. \quad \text{(due to (2.5))}.\end{aligned} \tag{4.21}$$

□

**Remark 4.12.** Proposition 4.11 implicates that if $W_{r_{pr}} = W$, then the second part of $\Delta_2^{pr}(\tilde{\mu})$ is always zero, *i.e.*, $|r_{du}^T(\tilde{\mu})\hat{x}_{r_{pr}}(\tilde{\mu})| = 0$, and $\Delta_2^{pr}(\tilde{\mu})$ equals to $\Delta_1(\tilde{\mu})$. Also, $\hat{x}_{r_{pr}}(\tilde{\mu}) = 0$ makes $\Delta_1^{pr}(\tilde{\mu})$ zero, meaning the first part of $\Delta_3(\tilde{\mu})$ and the first part of $\Delta_3^{pr}(\tilde{\mu})$ are all zeros. Therefore, $W_{r_{pr}}$ should also be carefully constructed to avoid being equal to $W$. For Galerkin projection, *i.e.*, $W_{r_{pr}} = V_{r_{pr}}$ and $W = V$, Proposition 4.11 reads: If $V_{r_{pr}} = V$, then $\hat{x}_{r_{pr}}(\tilde{\mu}) = 0$.

**Proposition 4.13.** *If* $W_{r_{rpr}} = W_{r_{pr}}$, *then* $\hat{x}_{r_{rpr}}(\tilde{\mu}) = 0$.

*Proof.* From the ROM of the primal-residual-residual system in (4.14),

$$\hat{x}_{r_{rpr}}(\tilde{\mu}) = V_{r_{rpr}}(W_{r_{rpr}}^T Q(\tilde{\mu})V_{r_{rpr}})^{-1}(W_{r_{rpr}}^T r_{r_{pr}}(\tilde{\mu})). \tag{4.22}$$

The last tow terms of the right-hand side of (4.22) are

$$\begin{aligned} W_{r_{rpr}}^T r_{r_{pr}}(\tilde{\mu}) &= W_{r_{pr}}^T r_{r_{pr}}(\tilde{\mu}) \quad (\text{if } W_{r_{rpr}} = W_{r_{pr}}) \\ &= W_{r_{pr}}^T (r_{pr}(\tilde{\mu}) - QV_{r_{pr}} z_{r_{pr}}(\tilde{\mu})) \\ &= 0 \quad (\text{due to } (4.4)). \end{aligned}$$

□

**Remark 4.14.** From Proposition 4.13, we see that if $W_{r_{rpr}} = W_{r_{pr}}$, then the second part of $\Delta_3^{pr}(\tilde{\mu})$ is always zero, *i.e.*, $|C(\tilde{\mu})\hat{x}_{r_{rpr}}(\tilde{\mu})| = 0$, so that $\Delta_3^{pr}(\tilde{\mu})$ is no better than $\Delta_1^{pr}(\tilde{\mu})$ in underestimating the true error. Similarly, in case of Galerkin projection, *i.e.*, $W_{r_{pr}} = V_{r_{pr}}$ and $W_{r_{rpr}} = V_{r_{rpr}}$, Proposition 4.13 reads: If $V_{r_{rpr}} = V_{r_{pr}}$, then $\hat{x}_{r_{rpr}}(\tilde{\mu}) = 0$.

## 4.6. Constructing projection matrices for the ROMs

The key components for computing the error estimators are the projection matrix pairs $(W, V)$, $(W_{du}, V_{du})$, $(W_{r_{du}}, V_{r_{du}})$ or $(W_{r_{pr}}, V_{r_{pr}})$, $(W_{r_{rpr}}, V_{r_{rpr}})$, which are used to construct the reduced systems in (2.5), (2.7), (3.5) or in (4.4), (4.14), respectively. For simplicity and clarity of analysis, we only use Galerkin projection for all the reduced systems, so that only one projection matrix $V, V_{du}$, $V_{r_{du}}$ or $V_{r_{pr}}$, $V_{r_{rpr}}$ needs to be computed for each reduced system. The analysis in this subsection can be extended to Petrov–Galerkin projection without many difficulties and could be addressed in a future work.

By definition of the reduced primal system (2.5), $V$ is also the projection matrix for constructing the ROM of the original model. Since the proposed error estimator does not depend on the MOR method, $V$ can be computed either using time-domain MOR methods, such as the reduced basis (RB) method, the proper orthogonal decomposition (POD) method [8,10], which use the snapshots in time domain (trajectories of the state vector $x$) to obtain $V$ or using frequency domain methods, such as multi-moment-matching [15].

The dual system (2.6), the dual-residual system (3.4), as well as the primal-residual system (4.2), the primal-residual-residual system (4.12) are parametric systems in frequency domain, with $\tilde{\mu} = s$ or $\tilde{\mu} = (\mu, s)$ being the vector of parameters. Similarly, we can compute the projection matrices for MOR of these systems either through snapshot based methods, or the multi-moment-matching method. The snapshots do not represent the trajectory of the solution in time domain, instead, they are the solution vectors at different samples of the parameter $\tilde{\mu}$.

In order to be consistent with the previous work in [16, 17], and to be comparable with existing results, we apply the frequency domain method, *i.e.*, the multi-moment-matching method [15] to derive the ROMs for all the systems contributing to the error estimator. To be self-contained, we also review the construction of $V, V_{du}$ and $V_{r_{du}}$, though it is detailed in [16]. It is illustrated in [16] that the reduced basis method can be seen as a special case of the multi-moment-matching method for systems in frequency domain.

### *4.6.1. Constructing $V$ using the multi-moment-matching method [15]*

When using the multi-moment-matching method proposed in [15] to construct the ROM, then $V$ can be computed as follows. We first consider the state vector $x(t,\mu)$ in frequency domain, *i.e.*, the state vector $x(\tilde{\mu})$ of the primal system. Assume that $Q(\tilde{\mu})$ has the following affine decomposition

$$Q(\tilde{\mu}) = Q_0 + h_1(\tilde{\mu})Q_1 + \ldots + h_p(\tilde{\mu})Q_p,$$

where $h_j(\tilde{\mu}) : \mathbb{C}^m \mapsto \mathbb{C}, j = 1,\ldots,p$ are scalar functions of $\tilde{\mu}$. $Q_0,\ldots,Q_p \in \mathbb{R}^{n\times n}$ are constant matrices, so are parameter independent. From the series expansion of $x(\tilde{\mu})$,

$$\begin{aligned} x(\tilde{\mu}) &= [Q(\tilde{\mu})]^{-1}B(\tilde{\mu}) \\ &= [Q_0 + h_1(\tilde{\mu})Q_1 + \ldots + h_p(\tilde{\mu})Q_p]^{-1}B(\tilde{\mu}) \\ &= [I - (\sigma_1 M_1 + \ldots + \sigma_p M_p)]^{-1}B_M \\ &= \sum_{k=0}^{\infty}(\sigma_1 M_1 + \ldots + \sigma_p M_p)^k B_M, \end{aligned} \tag{4.23}$$

where $\sigma_j = h_j(\tilde{\mu}) - h_j(\tilde{\mu}^i)$, $B_M = [Q(\tilde{\mu}^i)]^{-1}B(\tilde{\mu})$, $M_j = -[Q(\tilde{\mu}^i)]^{-1}Q_j$, $j = 1,2,\ldots,p$; $h(\tilde{\mu}^i) := (h_1(\tilde{\mu}^i),\ldots,h_p(\tilde{\mu}^i))$ is the expansion point at which the above power series of $x(\tilde{\mu})$ is derived. Since $h(\tilde{\mu}^i)$ is uniquely determined by $\tilde{\mu}^i$, we call $\tilde{\mu}^i$ the expansion point in the following text, for simplicity. The superscript $i$ fixes the parameter $\tilde{\mu}$ to a specific value $\tilde{\mu}^i$. It corresponds to the $i$th expansion point used in (4.26), $i = 1,\ldots,l$. There exist recursions between the coefficients of the series expansion as below,

$$\begin{aligned} R_0 &= \tilde{B}_M, \\ R_1 &= [M_1R_0,\ldots,M_pR_0], \\ R_2 &= [M_1R_1,\ldots,M_pR_1], \\ &\vdots \\ R_q &= [M_1R_{q-1},\ldots,M_pR_{q-1}], \\ &\vdots \end{aligned} \tag{4.24}$$

Here, $\tilde{B}_M = B_M$, if $B(\tilde{\mu})$ does not depend on $\mu$, *i.e.*, $B(\tilde{\mu}) = B$. Otherwise, $\tilde{B}_M = [B_{M_1},\ldots,B_{M_p}]$, $B_{M_j} = [Q(\tilde{\mu}^i)]^{-1}B_j$, $j = 1,\ldots,p$, if $B(\tilde{\mu})$ can be written in an affine form, *e.g.*, $B(\tilde{\mu}) = B_1\alpha_1(\mu) + \ldots + B_p\alpha_p(\mu)$, $\alpha_i(\mu) : \mathbb{C}^m \mapsto \mathbb{C}$. Then $V_{\tilde{\mu}^i}$ is computed as

$$\text{range}(V_{\tilde{\mu}^i}) = \text{span}\{R_0, R_1,\ldots,R_q\}_{\tilde{\mu}^i}, \tag{4.25}$$

where usually we require $q \le 1$ to avoid exponential increase of column dimension. The matrix $V_{\tilde{\mu}^i}$ depends on the expansion point $\tilde{\mu}^i$. Finally, $V$ can be constructed as

$$V = \text{orth}\left\{V_{\tilde{\mu}^1},\ldots,V_{\tilde{\mu}^l}\right\}. \tag{4.26}$$

The matrices $R_j, j = 0,\ldots,$ in (4.24) depend on the expansion point $\mu^i$, so that they can be written as $R_j(\tilde{\mu}^i)$. The multi-moments corresponding to the expansion point $\mu^i$ are defined as $C(\tilde{\mu}^i)R_j(\tilde{\mu}^i), j = 0,\ldots,$ where $R_j$ includes the $j$th order multi-moments. Corresponding multi-moments of the ROM (2.2) can also be defined using the projected reduced system matrices, *i.e.*, $\hat{C}(\tilde{\mu}^i)\hat{R}_j(\tilde{\mu}^i), j = 0,\ldots,$ where $\hat{R}_j = V^TR_jV$. When $C(\tilde{\mu})$ is a constant matrix $C$, it is proved in [15] that the multi-moments $\hat{C}\hat{R}_j(\tilde{\mu}^i), j = 0,\ldots q, i = 1,\ldots,l$, of the ROM (2.2) are equal to those of the FOM (2.1), if the projection matrix $V$ is constructed as in (4.26). An analogous proof applies to the case of parameter dependent $C(\tilde{\mu})$ to show a similar moment-matching property: $\hat{C}(\tilde{\mu}^i)\hat{R}_j(\tilde{\mu}^i) = C(\tilde{\mu}^i)R_j(\tilde{\mu}^i), j = 0,\ldots q, i = 1,\ldots,l$. In this sense, the multi-moments of the FOM are matched by the ROM till order $q$ at each expansion point $\tilde{\mu}^i$, for $i = 1,\ldots,l$.

*4.6.2. Constructing $V_{du}$ using multi-moment-matching*

If using the multi-moment-matching method, $V_{du}$ can also be constructed similarly as $V$. Considering the dual system in (2.6), $x_{du}(\tilde{\mu})$ can be written as

$$\begin{aligned} x_{du}(\tilde{\mu}) &= [Q(\tilde{\mu})]^{-T}C^T(\tilde{\mu}) \\ &= [Q_0^T + h_1(\tilde{\mu})Q_1^T + \ldots + h_p(\tilde{\mu})Q_p^T]^{-1}C^T(\tilde{\mu}) \\ &= [I - (\sigma_1\tilde{M}_1 + \ldots + \sigma_p\tilde{M}_p)]^{-1}C_M \\ &= \sum_{k=0}^{\infty}(\sigma_1\tilde{M}_1 + \ldots + \sigma_p\tilde{M}_p)^k C_M, \end{aligned} \tag{4.27}$$

where $C_M = [Q(\tilde{\mu}^i)]^{-T}C^T(\tilde{\mu})$, $\tilde{M}_j = -[Q(\tilde{\mu}^i)]^{-T}Q_j^T$, $j = 1, 2, \ldots, p$. The recursions between the coefficients of the series expansion in (4.27) are

$$\begin{aligned} \tilde{R}_0 &= \tilde{C}_M, \\ \tilde{R}_1 &= [\tilde{M}_1\tilde{R}_0, \ldots, \tilde{M}_p\tilde{R}_0], \\ \tilde{R}_2 &= [\tilde{M}_1\tilde{R}_1, \ldots, \tilde{M}_p\tilde{R}_1], \\ &\vdots \\ \tilde{R}_q &= [\tilde{M}_1\tilde{R}_{q-1}, \ldots, \tilde{M}_p\tilde{R}_{q-1}], \\ &\vdots \end{aligned} \tag{4.28}$$

Here, $\tilde{C}_M = C_M$, if $C(\tilde{\mu})$ does not depend on $\mu$, *i.e.*, $C(\tilde{\mu}) = C$. Otherwise, $\tilde{C}_M = [C_{M_1}, \ldots, C_{M_p}]$, $C_{M_i} = [Q(\tilde{\mu}^i)]^{-1}C_j$, $j = 1, \ldots, p$, if $C(\tilde{\mu})$ can be written in an affine form, *e.g.*, $C(\tilde{\mu}) = C_1\beta_1(\mu) + \ldots + C_p\beta_p(\mu)$. Then $V_{\tilde{\mu}^i}^{du}$ is computed as

$$\text{range}(V_{\tilde{\mu}^i}^{du}) = \text{span}\left\{\tilde{R}_0, \tilde{R}_1, \ldots, \tilde{R}_q\right\}_{\tilde{\mu}^i}. \tag{4.29}$$

Finally, $V_{du}$ can be constructed as

$$V_{du} = \text{orth}\left\{V_{\tilde{\mu}^1}^{du}, \ldots, V_{\tilde{\mu}^l}^{du}\right\}. \tag{4.30}$$

*4.6.3. Constructing $V_{r_{du}}$*

$V_{r_{du}}$ is used to construct the ROM of the dual-residual system and the error estimator $\Delta_2(\tilde{\mu})$. From the state vector of the dual-residual system (3.4), we see that

$$\begin{aligned} x_{r_{du}}(\tilde{\mu}) &= Q^{-T}(\tilde{\mu})r_{du}(\tilde{\mu}) \\ &= Q^{-T}(\tilde{\mu})C^T(\tilde{\mu}) - \hat{x}_{du}(\tilde{\mu}) \\ &= Q^{-T}(\tilde{\mu})C^T(\tilde{\mu}) - V_{du}z_{du}(\tilde{\mu}), \end{aligned} \tag{4.31}$$

where $Q^{-T}(\tilde{\mu})C^T(\tilde{\mu})$ is nothing but the state vector $x_{du}(\tilde{\mu})$ of the dual system.

Considering the series expansion of $x_{du}(\tilde{\mu})$ in (4.27), we see that taking the same expansion point as in (4.27), the series expansion leads to the subspace range($V_{du}$). Finally, $Q^{-T}(\tilde{\mu})C^T(\tilde{\mu})$ in the last equality of (4.31) provides no new information than $V_{du}$, so that we can use range($V_{du}$) as the subspace for approximating the trajectory space of $x_{r_{du}}(\tilde{\mu})$, *i.e.*, $V_{r_{du}} = V_{du}$. However, from Proposition 4.9, we know that $V_{r_{du}}$ should be different from $V_{du}$. Therefore, if we use expansion points different from those used for $V_{du}$ to obtain a second projection matrix $V_{r_{du}}^1$, which is different from $V_{du}$, then the projection matrix $V_{r_{du}} := \text{orth}\left\{V_{r_{du}}^1, V_{du}\right\}$ should represent the trajectory of $x_{r_{du}}(\tilde{\mu})$ well.

$V_{r_{du}}^1$ can be computed using the multi-moment-matching method as in (4.29) and (4.30), by choosing expansion points that are different from those used there, *i.e.*,

$$\text{range}(V_{\tilde{\mu}^j}^{r_{du}}) = \text{span}\left\{\tilde{R}_0, \tilde{R}_1, \ldots, \tilde{R}_q\right\}_{\tilde{\mu}^j}, j = 1, \ldots, l. \tag{4.32}$$

Finally,

$$\mathrm{range}(V_{r_{du}}) = \mathrm{orth}\left\{V_{\tilde{\mu}^1}^{r_{du}}, \dots, V_{\tilde{\mu}^l}^{r_{du}}, V_{du}\right\}. \tag{4.33}$$

The $\tilde{\mu}^j$ in (4.32) can be selected by a greedy algorithm searching the maximum of $|\hat{x}_{r_{du}}^T(\tilde{\mu})r_{pr}(\tilde{\mu})|$, the first part of $\Delta_2(\tilde{\mu})$ associated with $\hat{x}_{r_{du}}$, and are usually different from $\tilde{\mu}^i$ used for computing $V_{du}$.

### *4.6.4. Constructing* $V_{r_{pr}}$

From the state vector of the primal-residual system (4.2), we get

$$\begin{aligned} x_{r_{pr}}(\tilde{\mu}) &= Q^{-1}(\tilde{\mu})r_{pr}(\tilde{\mu}) \\ &= Q^{-1}(\tilde{\mu})B(\tilde{\mu}) - \hat{x}_{pr}(\tilde{\mu}) \\ &= Q^{-1}(\tilde{\mu})B(\tilde{\mu}) - Vz_{pr}(\tilde{\mu}), \end{aligned} \tag{4.34}$$

where $Q^{-1}(\tilde{\mu})B(\tilde{\mu})$ is exactly the state vector $x(\tilde{\mu})$ of the primal system.

Similarly to constructing $V_{r_{du}}$, we use expansion points different from those used for $V$ to obtain a second projection matrix $V_{r_{pr}}^1$ whose columns span a subspace that is different from the one spanned by the columns of $V$, then the projection matrix

$$V_{r_{pr}} := \mathrm{orth}\left\{V_{r_{pr}}^1, V\right\} \tag{4.35}$$

should represent the trajectory of $x_{r_{pr}}(\tilde{\mu})$ well.

### *4.6.5. Constructing* $V_{r_{rpr}}$

From the state vector of the primal-residual-residual system (4.12), we see that

$$\begin{aligned} x_{r_{rpr}}(\tilde{\mu}) &= Q^{-1}(\tilde{\mu})r_{r_{pr}}(\tilde{\mu}) \\ &= Q^{-1}(\tilde{\mu})(r_{pr}(\tilde{\mu}) - Q(\tilde{\mu})V_{r_{pr}}z_{r_{pr}}(\tilde{\mu})) \\ &= Q^{-1}(\tilde{\mu})(r_{pr}(\tilde{\mu}) - V_{r_{pr}}z_{r_{pr}}(\tilde{\mu})) \\ &= Q^{-1}(\tilde{\mu})(B(\tilde{\mu}) - QVz_{pr}(\tilde{\mu})) - V_{r_{pr}}z_{r_{pr}}(\tilde{\mu}) \\ &= Q^{-1}(\tilde{\mu})B(\tilde{\mu}) - Vz_{pr}(\tilde{\mu}) - V_{r_{pr}}z_{r_{pr}}(\tilde{\mu}). \end{aligned} \tag{4.36}$$

Taking the same expansion point as in (4.23), the series expansion of $Q^{-1}(\tilde{\mu})B(\tilde{\mu})$ in the last equation of (4.36) gives rise to the projection matrix $V$. Consequently, the subspace for $x_{r_{rpr}}(\tilde{\mu})$ is $\mathrm{range}(V, V_{r_{pr}})$, which is equivalent with $\mathrm{range}(V_{r_{pr}})$, since $V$ is already included in $V_{r_{pr}}$ in (4.35). This is in contradiction with Proposition 4.13 that $V_{r_{rpr}}$ should be different from $V_{r_{pr}}$. Therefore, $Q^{-1}(\tilde{\mu})B(\tilde{\mu})$ in the last equation of (4.36) cannot be expanded using the same expansion points as those for both $V$ and $V_{r_{pr}}$. Recall that $V_{r_{rpr}}$ is used to construct the ROM of the primal-residual-residual system (4.12) and contributes to the error estimator $\Delta_3^{pr}(\tilde{\mu})$. Then the expansion points for series expansion of $Q^{-1}(\tilde{\mu})B(\tilde{\mu})$ in the last equation of (4.36) can be iteratively chosen by searching the maximum of $|C(\mu)\hat{x}_{r_{rpr}}(\tilde{\mu})|$, the second part of $\Delta_3^{pr}(\tilde{\mu})$, which purely depends on the ROM built by $V_{r_{rpr}}$. Greedy algorithms computing the projection matrices are presented in Section 6.

## 5. Comparing the proposed error estimators with the existing ones

In this section, the proposed error estimators are compared with the existing and closely related error estimators: the randomized error estimator in [29], where some dual residual systems are employed to construct the error estimator, and the error bound in [17] using the inf-sup constant. We first briefly review the randomized error estimator in Section 5.1, then compare it with the proposed error estimators in the next subsection, where the proposed error estimators are also compared with regard to their similarities and differences. The comparison is only done for error estimators with high relevance or with comparability.

### 5.1. Review of the error estimator in [29]

State error estimation as well as output error estimation for parametric linear steady systems is proposed in [29] based on randomized residuals. Given the system has only a single input, the output error estimation can be used to estimate the transfer function error in frequency domain. The transfer function error $e_H(\tilde{\mu}) := H(\tilde{\mu}) - \hat{H}(\tilde{\mu})$ can be measured using the 2-norm $\|e_H(\tilde{\mu})\|_2$. The error estimator is given as

$$\|e_H(\tilde{\mu})\|_2 \approx \frac{1}{K}\left(\sum_{i=1}^{K} \delta_i^2\right)^{1/2} =: \Delta_r(\tilde{\mu}), \tag{5.1}$$

where $\delta_i = (x_{du}^i(\tilde{\mu}))^T r_{pr}(\tilde{\mu})$, and $x_{du}^i(\tilde{\mu})$ solves the $i$th random dual system,

$$Q(\tilde{\mu})^T x_{du}^i(\tilde{\mu}) = z_i, i = 1, \ldots, K, \tag{5.2}$$

where $z_i \sim \mathcal{N}(0, C^T(\tilde{\mu})C(\tilde{\mu}))$ is a random vector following the normal distribution with zero mean and covariance matrix $C^T(\tilde{\mu})C(\tilde{\mu}) \in \mathbb{R}^{n\times n}$. According to Remark 2.6 in [29], the random dual systems reduce to

$$Q(\tilde{\mu})^T x_{du}^i(\tilde{\mu}) = \xi_i C^T(\tilde{\mu}), i = 1, \ldots, K, \tag{5.3}$$

where $\xi_i \sim \mathcal{N}(0,1)$ is a random variable (scalar) with standard normal random distribution. Therefore, $x_{du}^i(\tilde{\mu})$ can be obtained by first solving the dual system in (2.6) to get $x_{du}(\tilde{\mu})$ and then multiplying $x_{du}(\tilde{\mu})$ with $\xi_i$, *i.e.*, $x_{du}^i(\tilde{\mu}) = \xi_i x_{du}(\tilde{\mu})$.

It is stated in [29] (Corollary 2.5) that under certain conditions, $\Delta_r$ is an error estimator of the true error with the probability

$$\mathbb{P}\{w^{-1}\Delta_r(\tilde{\mu}) \le \|e_H(\tilde{\mu})\|_2 \le w\Delta_r(\tilde{\mu}), \forall \tilde{\mu} \in \Xi\} \ge 1 - \delta, \tag{5.4}$$

where $w > \sqrt{e}$, $e$ is the Euler number, and $\Xi$ is a finite set of parameter samples, $0 < \delta < 1$. Note that the dual system (2.6) with large size $n$ needs to be solved at least once for every parameter to obtain $x_{du}^i$, this is still costly. Therefore, for single output systems, $x_{du}$ is replaced by $\hat{x}_{du}$, so that only the reduced dual system in (2.7) needs to be solved. For multiple output systems, each of the random dual systems in (5.2) is first reduced to a small system and then $x_{du}^i$ is approximated by the approximate solutions $\hat{x}_{du}^i$ computed from the reduced random dual systems. Finally, we have

$$\begin{aligned} \|e(\tilde{\mu})\|_2 &\approx \tfrac{1}{K}\left(\sum_{i=1}^{K} \delta_i^2\right)^{1/2} =: \Delta_r(\tilde{\mu}) \\ &\approx \tfrac{1}{K}\left(\sum_{i=1}^{K} \tilde{\delta}_i^2\right)^{1/2} =: \tilde{\Delta}_r(\tilde{\mu}), \end{aligned} \tag{5.5}$$

where $\tilde{\delta}_i = (\hat{x}_{du}^i(\tilde{\mu}))^T r_{pr}(\tilde{\mu})$.

### 5.2. Robustness comparison

This subsection gives more insights into the proposed error estimators by comparing them with the existing ones and by comparing highly relevant ones among them. The comparison mainly focuses on the robustness of the error estimators in predicting the true error in the following sense: we discuss whether some are more likely to underestimate or overestimate the true error than others.

– $\Delta_1(\tilde{\mu})$ *vs.* $\Delta_1^{pr}(\tilde{\mu})$: To compute $\Delta_1(\tilde{\mu})$, we need reduce both a primal system and a dual system. Whereas, the primal system and the primal-residual system are reduced to obtain $\Delta_1^{pr}(\tilde{\mu})$. Although it is not clear which one better estimates the true error theoretically, numerical results nevertheless show obvious superiority of $\Delta_1^{pr}(\tilde{\mu})$ over $\Delta_1(\tilde{\mu})$.

- $\Delta_1(\tilde{\mu})$ *vs.* $\Delta_2(\tilde{\mu})$: it is clear that $\Delta_2(\tilde{\mu})$ is an upper bound of $\Delta_1(\tilde{\mu})$, though it is not an upper bound of the true error. This means, $\Delta_1(\tilde{\mu})$ is more likely to underestimate the true error than $\Delta_2(\tilde{\mu})$, if $W_{r_{du}} \neq W_{du}$ due to Proposition 4.9.
- $\Delta_1(\tilde{\mu})$ *vs.* $\Delta_2^{pr}(\tilde{\mu})$: analogously, $\Delta_1(\tilde{\mu})$ is more likely to underestimate the true error than $\Delta_2^{pr}(\tilde{\mu})$, if $W_{r_{pr}} \neq W$ due to Proposition 4.11.
- $\Delta_1^{pr}(\tilde{\mu})$ *vs.* $\Delta_3(\tilde{\mu})$: $\Delta_1^{pr}(\tilde{\mu})$ is more likely to underestimate the true error than $\Delta_3(\tilde{\mu})$.
- $\Delta_1^{pr}(\tilde{\mu})$ *vs.* $\Delta_3^{pr}(\tilde{\mu})$: $\Delta_1^{pr}(\tilde{\mu})$ is more likely to underestimate the true error than $\Delta_3^{pr}(\tilde{\mu})$, if $W_{r_{rpr}} \neq W_{r_{pr}}$ due to Proposition 4.13.
- $\Delta_2(\tilde{\mu})$ *vs.* $\Delta_2^{pr}(\tilde{\mu})$: the only difference between $\Delta_2$ and $\Delta_2^{pr}(\tilde{\mu})$ is the difference between their second parts, where the ROM of the dual residual system $(\hat{x}_{du}(\tilde{\mu}))$ is used for $\Delta_2(\tilde{\mu})$, whereas the ROM of the primal-residual system $(\hat{x}_{r_{pr}}(\tilde{\mu}))$ is used for $\Delta_2^{pr}(\tilde{\mu})$. They also behave similarly in the numerical experiments.
- $\Delta_2(\tilde{\mu})$ *vs.* $\Delta_3(\tilde{\mu})$: the first term $|\hat{x}_{r_{du}}^T(\tilde{\mu})r_{pr}(\tilde{\mu})|$ of $\Delta_2(\tilde{\mu})$ results from the ROM of the primal system and that of the dual system. The first term $|C(\tilde{\mu})\hat{x}_{r_{pr}}(\tilde{\mu})|$ of $\Delta_3(\tilde{\mu})$ results from reducing the primal system and the primal-residual system. As for their second terms: $|\hat{x}_{du}^T(\tilde{\mu})r_{pr}(\tilde{\mu})|$ of $\Delta_2(\tilde{\mu})$ and $|\hat{x}_{du}^T(\tilde{\mu})r_{r_{pr}}(\tilde{\mu})|$ of $\Delta_3(\tilde{\mu})$, $r_{pr}(\tilde{\mu})$ is the residual from the ROM of the primal system, but $r_{r_{pr}}(\tilde{\mu})$ is the residual from the ROM of the primal-residual system. $r_{r_{pr}}(\tilde{\mu})$ is the result of two-step model reduction, whereas $r_{pr}$ results from one step of MOR. Numerical results show that $\Delta_2(\tilde{\mu})$ is more robust than $\Delta_3(\tilde{\mu})$, when $\Delta_2(\tilde{\mu})$ is computed properly, especially for near symmetric systems.
- $\Delta_3(\tilde{\mu})$ *vs.* $\Delta_3^{pr}(\tilde{\mu})$: The only difference between $\Delta_3(\tilde{\mu})$ and $\Delta_3^{pr}(\tilde{\mu})$ is the difference between their second parts, where $\hat{x}_{du}(\tilde{\mu})$, the quantity computed from the ROM of the dual system is used for $\Delta_3(\tilde{\mu})$, whereas, $\hat{x}_{r_{rpr}}(\tilde{\mu})$, the quantity computed from the ROM of the primal-residual-residual system is used for $\Delta_3^{pr}(\tilde{\mu})$. Numerical results in the next section show little difference between their effectivities.
- $\Delta_0(\tilde{\mu})$ *vs.* $\Delta_2(\tilde{\mu})$ in [17]: It is shown in [16] that $\Delta_0(\tilde{\mu})$ has motivated the derivation of $\Delta_2(\tilde{\mu})$ and can be seen as an upper bound of $\Delta_2(\tilde{\mu})$. Although $\Delta_0(\tilde{\mu})$ is an error bound of the transfer function error, it is much more time consuming to compute as compared with $\Delta_2(\tilde{\mu})$, since the smallest singular value of a large matrix (of the original model size $n$) needs to be solved for every parameter value in a given training set. $\Delta_2(\tilde{\mu})$ avoids this computational issue. Numerical tests on several models in [16] have shown that $\Delta_2(\tilde{\mu})$ is much tighter than $\Delta_0(\tilde{\mu})$ and behaves as an error bound, except for very small true errors close to machine precision.
- $\tilde{\Delta}_r(\tilde{\mu})$ in [29] *vs.* $\Delta_1(\tilde{\mu})$: From the proof of Theorem 1, we see that the quantity $|x_{du}^T(\tilde{\mu})r_{pr}(\tilde{\mu})|$ in (3.1) is exactly the true error. Using a similar description as in (5.4), $|x_{du}^T(\tilde{\mu})r_{pr}|$ satisfies

$$\mathbb{P}\{w^{-1}|x_{du}^T(\tilde{\mu})r_{pr}| \leq \|e_H(\tilde{\mu})\|_2 \leq w|x_{du}^T(\tilde{\mu})r_{pr}(\tilde{\mu})|, \forall \tilde{\mu} \in \Xi, \forall \Xi \in \mathcal{D}\} = 1, \tag{5.6}$$

with $w = 1$, which is an exact estimation of the true error not only for any $\tilde{\mu}$ in a given $\Xi$ as in (5.4), but also for any $\tilde{\mu}$ in $\mathcal{D}$. Here, $\mathcal{D}$ is the continuous parameter domain. Comparing (5.6) with (5.4), we know that $\Delta_r(\tilde{\mu})$ in (5.1) and (5.4) is an error estimator, whereas $|x_{du}^T(\tilde{\mu})r_{pr}(\tilde{\mu})|$ in (3.1) and (5.6) is the true error. Furthermore, the error estimator $\Delta_1(\tilde{\mu})$ is derived based on $|x_{du}^T(\tilde{\mu})r_{pr}(\tilde{\mu})|$ by replacing the true dual solution $x_{du}(\tilde{\mu})$ in $|x_{du}^T(\tilde{\mu})r_{pr}(\tilde{\mu})|$ with the approximate dual solution $\hat{x}_{du}(\tilde{\mu})$; whereas $\tilde{\Delta}_r(\tilde{\mu})$ is derived based on $\Delta_r(\tilde{\mu})$ in (5.1) also by replacing $x_{du}(\tilde{\mu})$ in $\Delta_r(\tilde{\mu})$ with $\hat{x}_{du}(\tilde{\mu})$. In summary, $\Delta_1(\tilde{\mu})$ is only a one-step approximation of the true error, whereas, $\tilde{\Delta}_r(\tilde{\mu})$ is a two-step approximation of the true error. It is therefore not difficult to see that $\Delta_1(\tilde{\mu})$ should be tighter than $\tilde{\Delta}_r(\tilde{\mu})$. Simulation results also show that $\tilde{\Delta}_r(\tilde{\mu})$ is often not as tight as $\Delta_1(\tilde{\mu})$. From the previous analyses, $\Delta_1(\tilde{\mu})$ is less accurate than all the other proposed error estimators, which can also be seen from the numerical results in Section 7. Therefore, it appears to be unnecessary to compare $\tilde{\Delta}_r(\tilde{\mu})$ with the other estimators.

## 5.3. Computational complexity comparison

Computing any of the error estimators discussed in this work needs to construct a ROM of the primal system. It is noticed that the projection matrix $V$ used to construct the ROM of the primal system (2.4) is the same matrix used to derive the ROM of the original system. Therefore, the ROM of the primal system can be derived *for free* in the sense that $V$ is obtained without additional computation. Except for constructing the ROM of the primal system, we list the following additional costs required by different error estimators.

- Computing $\Delta_0(\tilde{\mu})$ involves constructing the ROM of the dual system (2.6), and computing the inf-sup constant at each $\tilde{\mu}$ in the training set $\Xi$.
- Computing $\Delta_1(\tilde{\mu})$ or $\Delta_1^{pr}(\tilde{\mu})$ involves constructing the ROM of the dual system or the ROM of the primal-residual system (4.2).
- Computing $\Delta_2(\tilde{\mu})$, $\Delta_2^{pr}(\tilde{\mu})$ or $\Delta_3(\tilde{\mu})$ involves constructing the ROM of the dual system (2.6), and additionally the ROM of a corresponding residual system needs to be constructed: the ROM of the dual-residual system (3.4) or the ROM of the primal-residual system (4.2).
- Computing $\Delta_3^{pr}(\tilde{\mu})$ involves constructing the ROMs of two residual systems: the primal-residual system (4.2) and the primal-residual-residual system (4.12).
- Computing $\Delta_r(\tilde{\mu})$ involves constructing the ROM of the dual system if the output matrix $C$ is a vector, otherwise, $K$ ROMs of the $K$ random dual systems in (5.2) must be constructed.

From Section 4.6, we see that to construct the ROMs of the dual system, or any of the residual systems, one only has to solve several linear systems to compute the coefficients in the series expansion of the corresponding solution vector. For interpolatory MOR methods in frequency domain, the cost of constructing the ROM of any of the above mentioned system is equivalent to the cost of constructing the ROM of the original system. This means, in order to compute any of the error estimators, one or two additional ROMs need to be constructed at each iteration of the greedy algorithm. Except for constructing the ROMs, where linear systems of the original dimension need to be solved at the expansion points, *no extra* large-scale computations are needed to compute the error estimators. However, the error bound $\Delta_0(\tilde{\mu})$ has the highest computational cost, since computing the inf-sup constant means solving a large eigenvalue problem at each $\tilde{\mu}$ in $\Xi$ per iteration. Furthermore, from the proposed greedy algorithms in the next section, the additional ROMs are constructed simultaneously with the ROM (2.2) of the original system, no separate greedy algorithms are required as in [28].

## 6. Greedy algorithms for constructing the projection matrices

The aim of an efficient error estimator is to construct a ROM of the original system with satisfying accuracy and high reliability. In the following, we show algorithms for constructing the ROM of the original system, where an error estimator acts as a guidance for greedy constructing the projection matrix $V$ for the ROM. Again, we use Galerkin projection to compute the ROM of the original systems and the ROMs of the other systems, which are involved in computing the error estimators. To compute any of the proposed error estimators, corresponding projection matrices $V_{du}$, $V_{r_{du}}$, $V_{r_{pr}}$, $V_{r_{rpr}}$ need to be constructed simultaneously with $V$.

As compared with the algorithms in [16], we have included the proposed variants of the error estimator and computation of their corresponding projection matrices into the algorithms. The performance of the proposed error estimators as well as the existing ones are compared in the next section.

We first present the greedy scheme for non-parametric systems in Algorithm 1. The standard moment-matching method [7] is used to compute the projection matrices. The variable $\epsilon_{\text{tol}}$ is defined as the tolerance for the error of the reduced transfer function. Once the maximal error estimator over the whole sample set $\Xi$ is below the tolerance, the greedy algorithm stops. In every iteration, the $s$ sample corresponding to the maximal error estimator is chosen as the next expansion point $s_i$ (Step 21). Steps 5, 8, 12, 15 and Step 19 orthogonalize the vectors in $V(s_i)$ and $V_{du}(s_i)$, $V_{r_{du}}(s_i^\alpha)$, $V_{r_{pr}}(s_i^\alpha)$, $V_{r_{rpr}}\left(s_i^\beta\right)$ against the existing vectors in $V$ and $V_{du}$, $V_{r_{du}}$, $V_{r_{pr}}$, $V_{r_{rpr}}$, respectively.

In Algorithm 1, some steps are only implemented for certain error estimators, depending on which error estimator is being used. The expansion point $s_i^\alpha$ is chosen to iteratively construct $V_{r_{du}}$ or $V_{r_{pr}}$, while $s_i^\beta$ is chosen to construct $V_{r_{rpr}}$. The choice of the expansion points $s_i^\alpha$ or $s_i^\beta$ depends on the part of the error estimator that is solely decided by the corresponding projection matrices $V_{r_{du}}$, $V_{r_{pr}}$, or $V_{r_{rpr}}$. As for $\Delta_1^{pr}(\tilde{\mu})$, since $s_i$ is chosen according to $\Delta_1^{pr}$, $s_i^\alpha$ is chosen according to the norm of $r_{r_{pr}}$ to avoid $V_{r_{pr}}$ being identical with $V$ due to Proposition 4.11.

**Algorithm 1.** Greedy ROM construction for non-parametric systems (2.1).

```
Input: System matrices $E, A, B, C, \epsilon_{\text{tol}}$, $\Xi$: a set of samples of $s$ covering the interesting frequency range.
Output: The projection matrix $V$ for constructing the ROM in (2.2).
1: $V = [], V_{du} = [], V_{r_{du}} = [], V_{r_{pr}} = [], V_{r_{rpr}} = []$, set $\epsilon = \epsilon_{\text{tol}} + 1, q > 1$.
2: Initial expansion point: $s_i \in \Xi$, for $V, V_{du}$;  $s_i^\alpha \in \Xi$, for $V_{r_{du}}$(or $V_{r_{pr}}$);  $s_i^\beta \in \Xi$, for $V_{r_{rpr}}$, $i = 1$.
3: while $\epsilon > \epsilon_{\text{tol}}$ do
4:     $\text{range}(V(s_i)) = \text{span}\left\{\tilde{B}(s_i), \dots, (\tilde{A}(s_i))^{q-1}\tilde{B}(s_i)\right\}$, where $\tilde{A}(s) = (sE - A)^{-1}E$, $\tilde{B}(s) = (sE - A)^{-1}B$, and $q \ll n$
5:     $V = \text{orth}\{V, V(s_i)\}$.
6:     if $\Delta(s) \in \{\Delta_1(s), \Delta_2(s), \Delta_2^{pr}(s), \Delta_3(s)\}$ then
7:         $\text{range}(V_{du}(s_i)) = \text{span}\left\{\tilde{C}(s_i), \dots, (\tilde{A}_c(s_i))^{q-1}\tilde{C}(s_i)\right\}$, where $\tilde{A}_c(s) = (sE - A)^{-T}E^T$, $\tilde{C}(s) = (sE - A)^{-T}C^T$.
8:         $V_{du} = \text{orth}\{V_{du}, V_{du}(s_i)\}$.
9:     end if
10:    if  $\Delta(s) = \Delta_2(s)$ then
11:        $\text{range}(V_{r_{du}}(s_i^\alpha)) = \text{span}\left\{\tilde{C}(s_i^\alpha), \dots, (\tilde{A}_c(s_i^\alpha))^{q-1}\tilde{C}(s_i^\alpha)\right\}$.
12:        $V_{r_{du}} = \text{orth}\{V_{du}, V_{r_{du}}, V_{r_{du}}(s_i^\alpha)\}$.
13:    else if $\Delta(s) \in \{\Delta_1^{pr}(s), \Delta_2^{pr}(s), \Delta_3(s), \Delta_3^{pr}(s)\}$ then
14:        $\text{range}(V_{r_{pr}}(s_i^\alpha)) = \text{span}\left\{\tilde{B}(s_i^\alpha), \dots, (\tilde{A}(s_i^\alpha))^{q-1}\tilde{B}(s_i^\alpha)\right\}$.
15:        $V_{r_{pr}} = \text{orth}\left\{V, V_{r_{pr}}, V_{r_{pr}}(s_i^\alpha)\right\}$.
16:    end if
17:    if  $\Delta(s) = \Delta_3^{pr}(s)$ then
18:        $\text{range}\left(V_{r_{rpr}}\left(s_i^\beta\right)\right) = \text{span}\left\{\tilde{B}\left(s_i^\beta\right), \dots, \left(\tilde{A}\left(s_i^\beta\right)\right)^{q-1}\tilde{B}\left(s_i^\beta\right)\right\}$.
19:        $V_{r_{rpr}} = \text{orth}\left\{V, V_{r_{pr}}, V_{r_{rpr}}, V_{r_{rpr}}\left(s_i^\beta\right)\right\}$.
20:    end if
21:    $i = i + 1$, $s_i = \arg\max_{s \in \Xi} \Delta(s)$.
22:    if $\Delta(s) = \Delta_2(s)$ then
23:        $s_i^\alpha = \arg\max_{s \in \Xi} |\hat{x}_{r_{du}}^T(s) r_{pr}(s)|$.                    %second part of $\Delta_2(s)$
24:    else if $\Delta(s) = \Delta_2^{pr}(s)$ then
25:        $s_i^\alpha = \arg\max_{s \in \Xi} |r_{du}^T(s)\hat{x}_{r_{pr}}(s)|$.                    %second part of $\Delta_2^{pr}(s)$
26:    else if $\Delta(s) = \Delta_1^{pr}(s)$ then
27:        $s_i^\alpha = \arg\max_{s \in \Xi} \|r_{r_{pr}}(s)\|_2$.  $r_{r_{pr(s)}}$ is defined in (4.9).
28:    else if $\Delta(s) = \Delta_3(s)$, or $\Delta_3^{pr}(s)$ then
29:        $s_i^\alpha = \arg\max_{s \in \Xi} \Delta_1^{pr}(s)$.                    %first part of $\Delta_3(s)$ or $\Delta_3^{pr}(s)$
30:    end if
31:    if $\Delta(s) = \Delta_3^{pr}(s)$ then
32:        $s_i^\beta = \arg\max_{s \in \Xi} |C\hat{x}_{r_{rpr}}(s)|$.                    %second part of $\Delta_3^{pr}(s)$
33:    end if
34:    $\epsilon = \Delta(s_i)$.
35: end while
```

Algorithm 2 shows the adaptive scheme for linear parametric systems. Algorithm 2 is similar with Algorithm 1. Its only difference from Algorithm 1 is in computing the projection matrices at a chosen expansion point in Steps 4, 7, 11, 14 and Step 18, where the multi-moment-matching method instead of the moment-matching method is used.

In greedy algorithms, large linear systems are solved *only* at the expansion points *selected* from a training set $\Xi$, to construct the corresponding projection matrices for MOR and for computing the error estimators. For each projection matrix, only a single expansion point is selected at each iteration step of the greedy algorithm, requiring a quite limited number of large-scale linear solves. Finally, if the greedy algorithm converges in a few iterations, large-scale linear solves are performed only at a few samples in the training set $\Xi$.

**Algorithm 2.** Greedy ROM construction for parametric systems (2.1).

```
Input: System matrices $E(\mu), A(\mu), B(\mu), C(\mu), \epsilon_{\mathrm{tol}}$, $\Xi$: a set of samples of $\tilde{\mu}$ covering the interesting parameter domain.
Output: The projection matrix $V$ for constructing the ROM in (2.2).
$V = []$, $V_{du} = []$,$V_{r_{du}} = []$, $V_{r_{pr}} = []$, $V_{r_{r_{pr}}} = []$, set $\epsilon = \epsilon_{\mathrm{tol}} + 1$.
Initial expansion point: $\tilde{\mu}^i \in \Xi$ for $V, V_{du}$;  $\tilde{\mu}^i_\alpha$ for $V_{r_{du}}$(or $V_{r_{pr}}$);  $\tilde{\mu}^i_\beta$ for $V_{r_{r_{pr}}}$, $i = 1$.
while $\epsilon > \epsilon_{\mathrm{tol}}$ do
    compute $V_{\tilde{\mu}^i}$ following (4.25).
    $V = \mathrm{orth}\left\{V, V_{\tilde{\mu}^i)}\right\}$.
    if $\Delta(\tilde{\mu}) \in \{\Delta_1(\tilde{\mu}), \Delta_2(\tilde{\mu}), \Delta_2^{pr}(\tilde{\mu}), \Delta_3(\tilde{\mu})\}$ then
        compute $V_{\tilde{\mu}^i}^{du}$ following (4.29).
        $V_{du} = \mathrm{orth}\left\{V_{du}, V_{\tilde{\mu}^i}^{du}\right\}$.
    end if
    if $\Delta(\tilde{\mu}) = \Delta_2(\tilde{\mu})$ then
        compute $V_{\tilde{\mu}^i_\alpha}^{r_{du}}$ following (4.32).
        $V_{r_{du}} = \mathrm{orth}\left\{V_{du}, V_{r_{du}}, V_{\tilde{\mu}^i_\alpha}^{r_{du}}\right\}$.
    else if $\Delta(\tilde{\mu}) \in \{\Delta_1^{pr}(\tilde{\mu}), \Delta_2^{pr}(\tilde{\mu}), \Delta_3(\tilde{\mu}), \Delta_3^{pr}(\tilde{\mu})\}$ then
        compute $V_{\tilde{\mu}^i_\alpha}^{r_{pr}}$ following (4.25).
        $V_{r_{pr}} = \mathrm{orth}\left\{V, V_{r_{pr}}, V_{\tilde{\mu}^i_\alpha}^{r_{pr}}\right\}$.
    end if
    if $\Delta(\tilde{\mu}) = \Delta_3^{pr}(\tilde{\mu})$ then
        compute $V_{\tilde{\mu}^i_\beta}^{r_{r_{pr}}}$ following (4.25).
        $V_{r_{r_{pr}}} = \mathrm{orth}\left\{V, V_{r_{pr}}, V_{r_{r_{pr}}}, V_{\tilde{\mu}^i_\beta}^{r_{r_{pr}}}\right\}$.
    end if
    $i = i + 1$, $\tilde{\mu}^i = \arg\max_{\tilde{\mu} \in \Xi} \Delta(\tilde{\mu})$.
    if $\Delta(\tilde{\mu}) = \Delta_2(\tilde{\mu})$ then
        $\tilde{\mu}^i_\alpha = \arg\max_{\tilde{\mu} \in \Xi} |\hat{x}^T_{r_{du}}(\tilde{\mu}) r_{pr}(\tilde{\mu})|$.        %second part of $\Delta_2(\tilde{\mu})$
    else if $\Delta(s) = \Delta_2^{pr}(\tilde{\mu})$ then
        $\tilde{\mu}^i_\alpha = \arg\max_{s \in \Xi} |r^T_{du}(\tilde{\mu}) \hat{x}_{r_{pr}}(\tilde{\mu})|$.        %second part of $\Delta_2^{pr}(\tilde{\mu})$
    else if $\Delta(\tilde{\mu}) = \Delta_1^{pr}(\tilde{\mu})$ then
        $\tilde{\mu}^i_\alpha = \arg\max_{\tilde{\mu} \in \Xi} \|r_{r_{pr}}(\tilde{\mu})\|_2$.        % $r_{r_{pr}}(\tilde{\mu})$ is defined in (4.9).
    else if $\Delta(\tilde{\mu}) = \Delta_3(\tilde{\mu})$, or $\Delta_3^{pr}(\tilde{\mu})$ then
        $\tilde{\mu}^i_\alpha = \arg\max_{\tilde{\mu} \in \Xi} \Delta_1^{pr}(\tilde{\mu})$.
    end if
    if $\Delta(\tilde{\mu}) = \Delta_3^{pr}(\tilde{\mu})$ then
        $\tilde{\mu}^i_\beta = \arg\max_{\tilde{\mu} \in \Xi} |C(\mu) \hat{x}_{r_{r_{pr}}}(\tilde{\mu})|$.        %second part of $\Delta_3^{pr}(\tilde{\mu})$
    end if
    $\epsilon = \Delta(\tilde{\mu}^i)$.
end while
```

We point out in Remark 4.7, Section 4.5 that when a system is almost symmetric, $\Delta_1(\tilde{\mu})$ performs badly, which will in turn, affect the behavior of $\Delta_2(\tilde{\mu})$ and $\Delta_2^{pr}(\tilde{\mu})$. From the simulation results in the next section, we will see that, except for the CD player model, $\Delta_1(\tilde{\mu})$ is not a good estimator. It is observed that for the RLCtree model, $Q(s)$ is symmetric, and only two elements are different between the input vector $B$ and the transpose of the output vector $C$. For the MIMO example, the matrix $E$ is symmetric and $B = C^T$. For the parametric example, the mass matrix $M(\mu)$ is symmetric. The stiffness matrix is unsymmetric, but the maximal magnitude of the elements in the matrix $T^T(\mu) - T(\mu)$ is $O(10^{-18})$ for all the parameters. This implicates that $T(\mu)$ is very likely symmetric in theory, and the small differences between $T(\mu)$ and its transpose are probably caused by numerical errors. The damping matrix is unsymmetric, but the maximal magnitude of the elements in the damping matrix $D(\mu)$ is $O(10^{-11})$. Whereas, the maximal magnitude of the elements in $T(\mu)$ is $O(1)$. Since the matrix $Q(\tilde{\mu})$ is composed of $M(\mu)$, $T(\mu)$ and $D(\mu)$, *i.e.*, $Q(\tilde{\mu}) = s^2M(\mu) + sD(\mu) + T(\mu)$, and $s = 2\pi f, f \in [50, 250]$, it can be concluded that $Q(\tilde{\mu})$ is almost symmetric. All the three examples are close to the symmetric case indicated in Remark 4.7.

In the following, we propose two algorithms: Algorithms 3 and 4, aiming at improving the behavior of $\Delta_1(\tilde{\mu})$, $\Delta_2(\tilde{\mu})$ and $\Delta_2^{pr}(\tilde{\mu})$ for nearly symmetric systems. Their main difference from Algorithm 1 and 2 is that instead of using the same expansion point for $V_{du}$ and $V$, different expansion points ($s_i^\gamma$ or $\tilde{\mu}_\gamma^i$) are chosen for $V_{du}$ according to a different error criterion that directly depends on $V_{du}$, see Steps 21–25 in Algorithm 3 and Algorithm 4, respectively.

## 7. Simulation results

In this section, we show the performance of the proposed error estimators and the existing ones. Detailed analyses for each of them are presented accordingly. Since the error bound $\Delta_0(\tilde{\mu})$ in [17] has been compared in detail with the error estimator $\Delta_2(\tilde{\mu})$ in a recent work [16], we do not repeat this comparison. Furthermore, since $\Delta_0(\tilde{\mu})$ was shown to be less tight than $\Delta_2(\tilde{\mu})$, it will not be compared with other error estimators either, as it will be clear from the results below that $\Delta_0(\tilde{\mu})$ may not outperform most of the error estimators.

We use the same four models as in [16] to show the robustness of the error estimators. The first two are non-parametric SISO systems. One is a well-known MOR benchmark example, the model of a CD player (with order $n = 120$), the other is a model of an RLC tree circuit with order $n = 6134$. The third example is a circuit model with $n = 980$. It has 4 inputs and 4 outputs, and no parameters. Both the CD player model and the third multi-input multi-output (MIMO) circuit model are from the SLICOT benchmark collection[1]. The last one is the model of a butterfly-shaped micro-gyroscope, available from the MOR benchmark collection[2]. It is a second-order parametric system with $n = 17\,931$.

The interesting frequency band for the CD player model is $[0, 1\,\mathrm{MHz}]$. The frequency range of interest for the second and the third models is $[0, 3\,\mathrm{GHz}]$. The Gyroscope model is a low frequency problem with $f \in [50\,\mathrm{Hz}, 250\,\mathrm{Hz}]$.

The error tolerance $\epsilon_{\mathrm{tol}}$ used in the greedy algorithms, *i.e.*, the error tolerance for the error of the ROM of the original system, is set as $1 \times 10^{-3}$ for the first three examples, while for the last example, we set $\epsilon_{\mathrm{tol}} = 1 \times 10^{-7}$, since the transfer function $H(\mu)$ has the smallest magnitude of $2.8 \times 10^{-7}$.

For all the non-parametric examples, we use $q = 3$ (order of moments matched) in Algorithm 1 and Algorithm 3. For the parametric model, we use $R_0, R_1$ ($\tilde{R}_0, \tilde{R}_1$) to generate the matrices $V_{\tilde{\mu}^i}$, $V_{\tilde{\mu}^i}^{du}$, $V_{\tilde{\mu}_\alpha^i}^{r_{du}}$, $V_{\tilde{\mu}_\alpha^i}^{r_{pr}}$ and $V_{\tilde{\mu}_\beta^i}^{r_{rpr}}$ in Algorithms 2 and 4. At each iteration, the maximal error estimator in $\Xi$, is computed, and is used as the error control for the ROM (2.2) of the original system. Therefore, the maximal true error $\varepsilon_{\max} = \max_{\mu^i \in \Xi} \varepsilon(\mu^i)$ is used for comparison, where $\varepsilon(\mu^i)$ is the true error of the ROM evaluated at $\mu^i$, at the current iteration of the algorithm. Different error estimators produce ROMs with different accuracy at each

[1] http://www.icm.tu-bs.de/NICONET/benchmodred.html.

[2] https://morwiki.mpi-magdeburg.mpg.de/morwiki.

**Algorithm 3.** Improving $\Delta_1(\tilde{\mu})$, $\Delta_2(\tilde{\mu})$ and $\Delta_2^{pr}(\tilde{\mu})$ for nearly symmetric and non-parametric systems (2.1).

**Input:** System matrices $E, A, B, C, \epsilon_{\text{tol}}$, $\Xi$: a set of samples of $s$ covering the interesting frequency range.
**Output:** The projection matrix $V$ for constructing the ROM in (2.2).

```
$V = [], V_{du} = [], V_{r_{du}} = [], V_{r_{pr}} = []$ set $\epsilon = \epsilon_{\text{tol}} + 1, q > 1$.
Initial expansion point: $i = 1$, $s_i \in \Xi$ for $V$;  $s_i^{\alpha} \in \Xi$ for $V_{r_{du}}$(or $V_{r_{pr}}$);  $s_i^{\gamma} \in \Xi$ for $V_{du}$.
while $\epsilon > \epsilon_{\text{tol}}$ do
    range$(V(s_i))$ = span$\left\{\tilde{B}(s_i), \dots, (\tilde{A}(s_i))^{q-1}\tilde{B}(s_i)\right\}$.
    $V$ = orth$\{V, V(s_i)\}$
    range$(V_{du}(s_i^{\gamma}))$ = span$\left\{\tilde{C}(s_i^{\gamma}), \dots, (\tilde{A}_c(s_i^{\gamma}))^{q-1}\tilde{C}(s_i^{\gamma})\right\}$.
    $V_{du}$ = orth$\{V_{du}, V_{du}(s_i^{\gamma})\}$.
    if  $\Delta(s) = \Delta_2(s)$ then
        range$(V_{r_{du}}(s_i^{\alpha}))$ = span$\left\{\tilde{C}(s_i^{\alpha}), \dots, (\tilde{A}_c(s_i^{\alpha}))^{q-1}\tilde{C}(s_i^{\alpha})\right\}$.
        $V_{r_{du}}$ = orth$\{V_{du}, V_{r_{du}}, V_{r_{du}}(s_i^{\alpha})\}$.
    else if $\Delta(s) = \Delta_2^{pr}(s)$ then
        range$(V_{r_{pr}}(s_i^{\alpha}))$ = span$\left\{\tilde{B}(s_i^{\alpha}), \dots, (\tilde{A}(s_i^{\alpha}))^{q-1}\tilde{B}(s_i^{\alpha})\right\}$.
        $V_{r_{pr}}$ = orth$\{V, V_{r_{pr}}, V_{r_{pr}}(s_i^{\alpha})\}$.
    end if
    $i = i + 1$, $s_i = \arg\max_{s \in \Xi} \Delta(s)$.
    if $\Delta(s) = \Delta_2(s)$ then
        $s_i^{\alpha} = \arg\max_{s \in \Xi} |\hat{x}_{r_{du}}^T(s) r_{pr}(s)|$.                %second part of $\Delta_2(s)$
    else if $\Delta(s) = \Delta_2^{pr}(s)$ then
        $s_i^{\alpha} = \arg\max_{s \in \Xi} |r_{du}^T(s) \hat{x}_{r_{pr}}(s)|$.                %second part of $\Delta_2^{pr}(s)$
    end if
    if $\Delta(s) = \Delta_1(s)$ then
        $s_i^{\gamma} = \arg\max_{s \in \Xi} \|r_{du}(s)\|_2$.
    else if $\Delta(s) \in \{\Delta_2(s), \Delta_2^{pr}\}$ then
        $s_i^{\gamma} = \arg\max_{s \in \Xi} \Delta_1(s)$.                %first part of $\Delta_2(s)$ or $\Delta_2^{pr}$
    end if
    $\epsilon = \Delta(s_i)$.
end while
```

iteration of the greedy algorithm. In the tables below, we write $\varepsilon_{\max}(\Delta)$ to indicate the maximal true error corresponding to a specific error estimator $\Delta$. Here, $\Delta$ represents any of the error estimators listed in the tables.

For Algorithms 1 and 2, the initial expansion point $s_1$ or $\tilde{\mu}^1$ for computing $V, V_{du}$ is taken as the first sample in $\Xi$, and the initial expansion point $s_1^{\alpha}$ or $\tilde{\mu}_{\alpha}^1$ for computing $V_{r_{du}}$, $V_{r_{pr}}$ is taken as the last sample in $\Xi$ to make the two expansion points different from each other. The expansion point $s_1^{\beta}$ or $\tilde{\mu}_{\beta}^1$ is for $V_{r_{rpr}}$. It is taken as the midpoint in $\Xi$. Algorithms 3 and 4 are for (nearly) symmetric systems, and the initial expansion points $s_1, \tilde{\mu}^1$ for $V$ are different from $s_1^{\gamma}, \tilde{\mu}_{\gamma}^1$ for $V_{du}$. Therefore, $s_1$ or $\tilde{\mu}^1$ is taken as the first sample in $\Xi$ and $s_1^{\gamma}$ or $\tilde{\mu}_{\gamma}^1$ is taken as the midpoint in $\Xi$. The initial point $s_1^{\alpha}$ or $\tilde{\mu}_{\alpha}^1$ for $V_{r_{du}}$, $V_{r_{pr}}$ is taken as the last point in $\Xi$.

### 7.1. The CD player model

The training set $\Xi$ for this model contains 60 samples of $s$, and then the finally obtained ROM in (2.2) is validated at 600 samples of $s$ covering the whole interesting frequency range. The samples are taken from the interval $[0, 1\,\text{MHz}]$ using the MATLAB function "logspace". The results of Algorithm 1 using different error estimators are shown in Tables 1–3, where the error estimators and the corresponding true errors $\epsilon_{\max}$ of the ROMs at each iteration of the Algorithm, are listed. Note that different ROMs are derived by using different

**Algorithm 4.** Improving $\Delta_1(\tilde{\mu})$, $\Delta_2(\tilde{\mu})$ and $\Delta_2^{pr}(\tilde{\mu})$ for nearly symmetric and parametric systems (2.1).

```
Input: System matrices $E(\mu), A(\mu), B(\mu), C(\mu), \epsilon_{\text{tol}}$, $\Xi$: a set of samples of $\tilde{\mu}$ covering the interesting frequency range.
Output: The projection matrix $V$ for constructing the ROM in (2.2).
$V = []$, $V_{du} = []$, $V_{r_{du}} = []$, $V_{r_{pr}} = []$, set $\epsilon = \epsilon_{\text{tol}} + 1$.
Initial expansion point: $\tilde{\mu}^i \in \Xi$ for $V$;  $\tilde{\mu}_\alpha^i \in \Xi$ for $V_{r_{du}}$(or $V_{r_{pr}}$);  $\tilde{\mu}_\gamma^i \in \Xi$ for $V_{du}$; $i = 1$.
while $\epsilon > \epsilon_{\text{tol}}$ do
    compute $V_{\tilde{\mu}^i}$ following (4.25).
    $V = \text{orth}\left\{V, V_{\tilde{\mu}^i)}\right\}$.
    compute $V_{\tilde{\mu}_\gamma^i}^{du}$ following (4.29).
    $V_{du} = \text{orth}\left\{V_{du}, V_{\tilde{\mu}_\gamma^i}^{du}\right\}$.
    if $\Delta(\tilde{\mu}) = \Delta_2(\tilde{\mu})$ then
        compute $V_{\tilde{\mu}_\alpha^i}^{r_{du}}$ following (4.32).
        $V_{r_{du}} = \text{orth}\left\{V_{du}, V_{r_{du}}, V_{\tilde{\mu}_\alpha^i}^{r_{du}}\right\}$.
    else if $\Delta(\tilde{\mu}) = \Delta_2^{pr}(\tilde{\mu})$ then
        compute $V_{\tilde{\mu}_\alpha^i}^{r_{pr}}$ following (4.25).
        $V_{r_{pr}} = \text{orth}\left\{V, V_{r_{pr}}, V_{\tilde{\mu}_\alpha^i}^{r_{pr}}\right\}$.
    end if
    $i = i + 1$, $\tilde{\mu}^i = \arg\max_{\tilde{\mu} \in \Xi} \Delta(\tilde{\mu})$.
    if $\Delta(\tilde{\mu}) = \Delta_2(\tilde{\mu})$ then
        $\tilde{\mu}_\alpha^i = \arg\max_{\tilde{\mu} \in \Xi} |\hat{x}_{r_{du}}^T(\tilde{\mu}) r_{pr}(\tilde{\mu})|$.          %second part of $\Delta_2(\tilde{\mu})$
    else if $\Delta(\tilde{\mu}) = \Delta_2^{pr}(\tilde{\mu})$ then
        $\tilde{\mu}_\alpha^i = \arg\max_{\tilde{\mu} \in \Xi} |r_{du}^T(\tilde{\mu}) \hat{x}_{r_{pr}}(\tilde{\mu})|$.          %second part of $\Delta_2^{pr}(\tilde{\mu})$
    end if
    if $\Delta(\tilde{\mu}) = \Delta_1(\tilde{\mu})$ then
        $\tilde{\mu}_\gamma^i = \arg\max_{\tilde{\mu} \in \Xi} \|r_{du}(\tilde{\mu})\|$.
    else if $\Delta(\tilde{\mu}) \in \{\Delta_2(\tilde{\mu}), \Delta_2^{pr}(\tilde{\mu})\}$ then
        $\tilde{\mu}_\gamma^i = \arg\max_{\tilde{\mu} \in \Xi} \Delta_1(\tilde{\mu})$.          %first part of $\Delta_2(\tilde{\mu})$ or $\Delta_2^{pr}(\tilde{\mu})$
    end if
    $\epsilon = \Delta(\tilde{\mu}^i)$.
end while
```

error estimators, therefore the true errors depend on the error estimators and are usually different. This also applies to analogous results listed in the other tables for other examples.

In Table 1, we also show the results for $\tilde{\Delta}_r(s)$ from [29], where $K$ in (5.5) is taken as $K = 20$, which is shown to produce better results than $K = 10$ [29]. In this table and other tables that follow, "–" means the algorithm terminated at the previous iteration and no further results are given. During the greedy iteration, $\tilde{\Delta}_r(s)$ always underestimates the maximal true error. $\Delta_1^{pr}$underestimates the true error at the first 5 iterations, but then becomes an accurate estimator at the last two iterations. $\Delta_1(s)$ is better than $\tilde{\Delta}_r(s)$, but is no better than the other estimators. $\Delta_2(s)$ and its primal version $\Delta_2^{pr}(s)$ behave like error bounds. $\Delta_1^{pr}(s)$, $\Delta_3(s)$ and $\Delta_3^{pr}(s)$ have underestimation only at the first several iterations. In general, once they bound error from above, they are very tight.

We further validate the ROM obtained by the error estimators at samples in $\Xi_{\text{ver}}$ including 600 samples randomly taken from $[0, 1\,\text{MHz}]$, the results are presented in Table 4 and plotted in Figures 1–4. In Table 4, we compare the effectivity defined as $\mathsf{eff}(s) := \Delta(s)/\varepsilon(s)$, the ratio between the given error estimator and its corresponding true error. $\tilde{\Delta}_r(s)$ still underestimates the true error at most samples. $\Delta_1^{pr}(s)$, $\Delta_2(s)$, $\Delta_3(s)$, $\Delta_3^{pr}(s)$ are equally well, whereas $\Delta_1(s)$ and $\Delta_2^{pr}(s)$ underestimate the true error too much ($\min(\mathsf{eff}) < 0.1$) at some samples. However, we observe that underestimation happens only at samples with very small true errors

TABLE 1. CD player, $\varepsilon_{\text{tol}} = 10^{-3}$, $q = 3$, $r = 44(\Delta_r)$, $r = 52(\Delta_1)$, $r = 56(\Delta_1^{pr})$.

| Iteration $i$ | $\varepsilon_{\max}(\tilde{\Delta}_r)$ | $\tilde{\Delta}_r(s_i)$ | $\varepsilon_{\max}(\Delta_1)$ | $\Delta_1(s_i)$ | $\varepsilon_{\max}(\Delta_1^{pr})$ | $\Delta_1^{pr}(s_i)$ |
|---|---|---|---|---|---|---|
| 1 | 61.63 | 21.88 | 40.75 | 34.93 | 40.75 | 2.56 |
| 2 | 51.98 | 18.46 | 19.34 | 33.92 | 19.34 | 1.07 |
| 3 | 14.49 | 5.14 | 0.59 | 1.47 | 14.48 | 0.64 |
| 4 | 0.76 | 0.27 | 0.31 | 0.26 | 14.45 | 5.46 |
| 5 | 0.11 | 0.04 | 0.06 | 0.11 | 0.26 | 0.26 |
| 6 | 0.0016 | $5.86 \times 10^{-4}$ | 0.04 | 0.04 | 0.0024 | 0.0024 |
| 7 | – | – | $6.81 \times 10^{-4}$ | $7.65 \times 10^{-4}$ | $1.28 \times 10^{-5}$ | $1.28 \times 10^{-5}$ |

TABLE 2. CD player, $\varepsilon_{\text{tol}} = 10^{-3}$, $q = 3$, $r = 52$.

| Iteration $i$ | $\varepsilon_{\max}(\Delta_2)$ | $\Delta_2(s_i)$ | $\varepsilon_{\max}(\Delta_2^{pr})$ | $\Delta_2^{pr}(s_i)$ |
|---|---|---|---|---|
| 1 | 40.75 | 51 | 40.75 | 46.1 |
| 2 | 30.16 | 35.75 | 19.34 | 52.2 |
| 3 | 0.75 | 5.41 | 0.59 | 1.95 |
| 4 | 0.32 | 0.4 | 0.31 | 0.38 |
| 5 | 0.03 | 0.03 | 0.06 | 0.19 |
| 6 | 0.002 | 0.002 | 0.04 | 0.04 |
| 7 | $8.28 \times 10^{-4}$ | $8.38 \times 10^{-4}$ | $6.82 \times 10^{-4}$ | $8.48 \times 10^{-4}$ |

TABLE 3. CD player, $\varepsilon_{\text{tol}} = 10^{-3}$, $q = 3$, $r = 52$.

| Iteration $i$ | $\varepsilon_{\max}(\Delta_3)$ | $\Delta_3(s_i)$ | $\varepsilon_{\max}(\Delta_3^{pr})$ | $\Delta_3^{pr}(s_i)$ |
|---|---|---|---|---|
| 1 | 40.75 | 35.45 | 40.75 | 34.95 |
| 2 | 19.34 | 35.19 | 16.81 | 51.76 |
| 3 | 0.59 | 0.84 | 9.1 | 9.1 |
| 4 | 0.31 | 0.4 | 0.21 | 0.24 |
| 5 | 0.05 | 0.05 | 0.03 | 0.03 |
| 6 | 0.002 | 0.002 | 0.0016 | 0.0016 |
| 7 | $8.27 \times 10^{-4}$ | $8.27 \times 10^{-4}$ | $7.57 \times 10^{-4}$ | $7.57 \times 10^{-4}$ |

$\varepsilon(s)$ being smaller than $10^{-11}$, which may be caused by rounding errors. If we check the error estimators only at true errors larger than $10^{-11}$, then we obtain the last two columns in the table, which show that except for $\tilde{\Delta}_r(s)$ the other estimators are tight.

Figure 1 further shows the inaccuracy of $\tilde{\Delta}_r(s)$ validated at the 600 samples in $\Xi_{\text{ver}}$. $\Delta_1(s)$ in Figure 1 behaves slightly worse than the other proposed estimators, see Figures 2–4. In the following, we will omit the results of $\tilde{\Delta}_r(\tilde{\mu})$ for the other examples, since it is always worse than the others.

## 7.2. The RLC tree model

We use a training set $\Xi$ with 90 frequency samples covering the whole frequency range $[0, 3\,\text{GHz}]$. The samples $s_i$ are taken using the function $f_i = 3 \times 10^{i/10}$, $s_i = 2\pi \jmath$, $i = 1, \dots, 90$. Here, $\jmath$ is the imaginary unit. The results of the greedy algorithm using different error estimators are listed in Tables 5–7. $\Delta_1(s)$ always underestimates the true error, and finally it makes the greedy algorithm stop before the true error $\varepsilon_{\max}$ is below the tolerance.

TABLE 4. CD player, effectivity of the error estimators.

| Estimator | For all $\varepsilon(s)$ | | For $\varepsilon(s) \geq 10^{-11}$ | |
|---|---|---|---|---|
| | $\min\limits_{s\in\Xi_{\text{ver}}}$ (eff) | $\max\limits_{s\in\Xi_{\text{ver}}}$ (eff) | $\min\limits_{s\in\Xi_{\text{ver}}}$ (eff) | $\max\limits_{s\in\Xi_{\text{ver}}}$ (eff) |
| $\tilde{\Delta}_r$ | 0.09 | 1.82 | 0.26 | 0.26 |
| $\Delta_1$ | 0.02 | 80 | 0.9211 | 1.1785 |
| $\Delta_1^{pr}$ | 0.28 | 20.39 | 0.9988 | 1.0046 |
| $\Delta_2$ | 0.12 | 17.32 | 0.9987 | 1.1653 |
| $\Delta_2^{pr}$ | 0.02 | 80 | 1.0000 | 1.3643 |
| $\Delta_3$ | 0.12 | 10.97 | 0.9993 | 1.0004 |
| $\Delta_3^{pr}$ | 0.1 | 9.13 | 0.9998 | 5.31 |

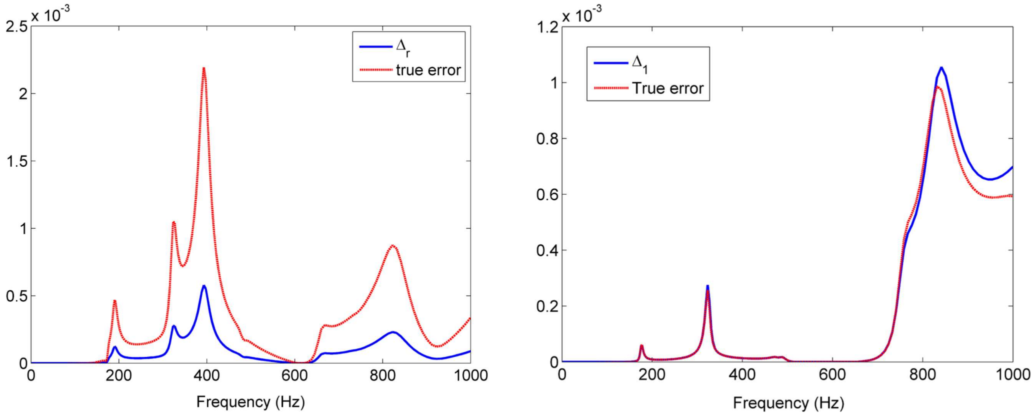

FIGURE 1. CD player: $\Delta_r(s)$ and $\Delta_1(s)$ *vs.* the respective true errors at 600 frequency samples.

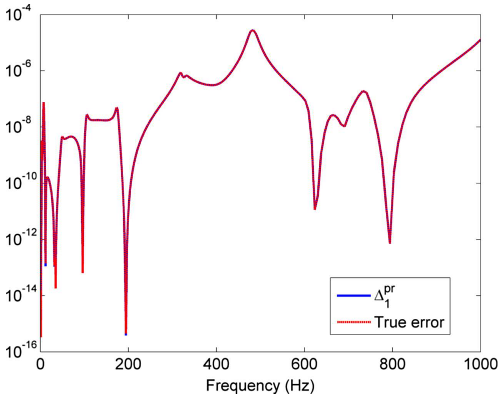

FIGURE 2. CD player: $\Delta_1^{pr}(s)$ *vs.* the true errors at 600 frequency samples.

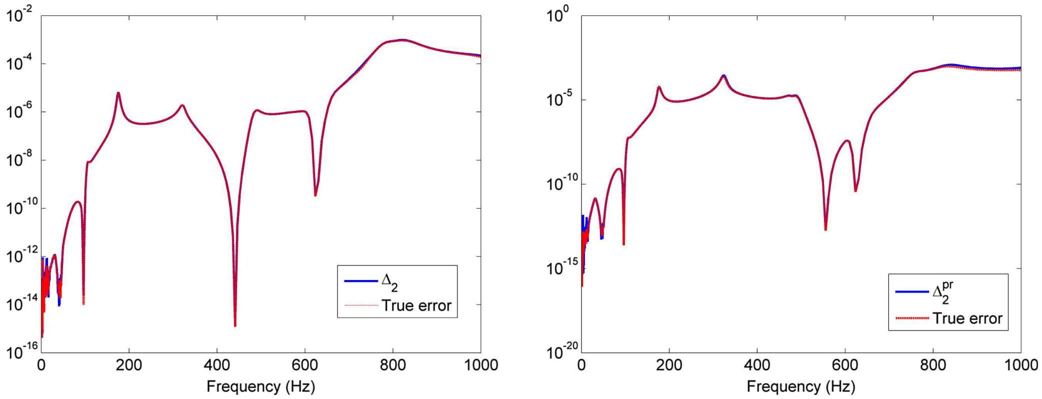

FIGURE 3. CD player: $\Delta_2(s)$ and $\Delta_2^{pr}(s)$ *vs.* the respective true errors at 600 frequency samples.

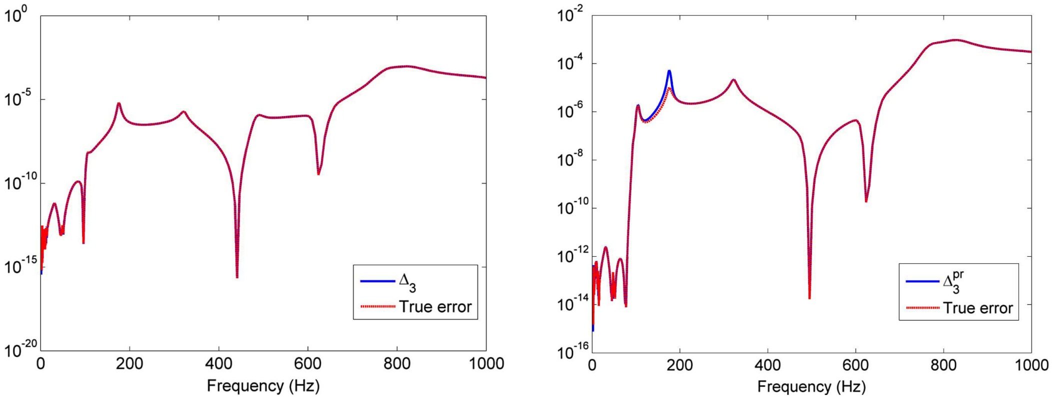

FIGURE 4. CD player: $\Delta_3(s)$ and $\Delta_3^{pr}(s)$ *vs.* the respective true errors at 600 frequency samples.

TABLE 5. RLCtree, $\varepsilon_{\text{tol}} = 10^{-3}$, $q = 3$, $r = 12(\Delta_1)$, $r = 20(\Delta_1^{pr})$.

| Iteration $i$ | $\varepsilon_{\max}(\Delta_1)$ | $\Delta_1(s_i)$ | $\varepsilon_{\max}(\Delta_1^{pr})$ | $\Delta_1^{pr}(s_i)$ |
|---|---|---|---|---|
| 1 | 0.19 | 0.01 | 0.19 | 0.22 |
| 2 | 0.06 | 0.006 | 0.02 | 0.02 |
| 3 | – | – | $2.54 \times 10^{-6}$ | $2.55 \times 10^{-6}$ |

The other estimators behave like tight upper bounds for the true error in this example, especially $\Delta_3(s)$ and $\Delta_3^{pr}(s)$, which actually measure the true error almost exactly at the last two iterations.

The derived ROMs using different error estimators are validated on a validation set $\Xi_{\text{ver}}$ with 900 samples in the interesting frequency range. The effectivity of every error estimator is listed in Table 8. If considering the overall effectivity, then all the estimators underestimate the true error too much except for $\Delta_2^{pr}(s)$. However, if only considering true errors that are bigger than $10^{-11}$, then $\Delta_1^{pr}(s)$, $\Delta_3(s)$ and $\Delta_3^{pr}(s)$ are the best ones, $\Delta_2(s)$ is also good, $\Delta_2^{pr}(s)$ overestimate the true error more than many others. It is clear that $\Delta_1(s)$ is not a good error estimator any more. Figures 5–7 further show the behaviors of the error estimators over the sample set $\Xi_{\text{ver}}$ including 900 samples, which are in agreement with the above analysis for the data in Table 8.

Table 6. RLCtree, $\varepsilon_{\text{tol}} = 10^{-3}$, $q = 3$, $r = 20(\Delta_2)$, $r = 19(\Delta_2^{pr})$.

| Iteration $i$ | $\varepsilon_{\max}(\Delta_2)$ | $\Delta_2(s_i)$ | $\varepsilon_{\max}(\Delta_2^{pr})$ | $\Delta_2^{pr}(s_i)$ |
|---|---|---|---|---|
| 1 | 0.19 | 0.63 | 0.19 | 0.22 |
| 2 | 0.02 | 0.06 | 0.02 | 0.05 |
| 3 | $6.13 \times 10^{-6}$ | $6.45 \times 10^{-6}$ | $2.25 \times 10^{-5}$ | $1.05 \times 10^{-4}$ |

Table 7. RLCtree, $\varepsilon_{\text{tol}} = 10^{-3}$, $q = 3$, $r = 20$.

| Iteration $i$ | $\varepsilon_{\max}(\Delta_3)$ | $\Delta_3(s_i)$ | $\varepsilon_{\max}(\Delta_3^{pr})$ | $\Delta_3^{pr}(s_i)$ |
|---|---|---|---|---|
| 1 | 0.19 | 0.22 | 0.19 | 0.29 |
| 2 | 0.02 | 0.02 | 0.02 | 0.02 |
| 3 | $2.54 \times 10^{-6}$ | $2.55 \times 10^{-6}$ | $2.54 \times 10^{-6}$ | $2.54 \times 10^{-6}$ |

Table 8. RLCtree, effectivity of the error estimators.

| Estimator | For all $\varepsilon(s)$ | | For $\varepsilon(s) \geq 10^{-11}$ | |
|---|---|---|---|---|
| | $\min_{s\in\Xi_{\text{ver}}}$ (eff) | $\max_{s\in\Xi_{\text{ver}}}$ (eff) | $\min_{s\in\Xi_{\text{ver}}}$ (eff) | $\max_{s\in\Xi_{\text{ver}}}$ (eff) |
| $\Delta_1$ | 0.002 | 285 | 0.006 | 132 |
| $\Delta_1^{pr}$ | 0.002 | 253 | 0.9001 | 1.0826 |
| $\Delta_2$ | 0.004 | 244 | 0.37 | 51 |
| $\Delta_2^{pr}$ | 0.56 | 102 | 0.68 | 102 |
| $\Delta_3$ | 0.008 | 258 | 0.9 | 1.2337 |
| $\Delta_3^{pr}$ | 0.008 | 258 | 0.9 | 1.0894 |

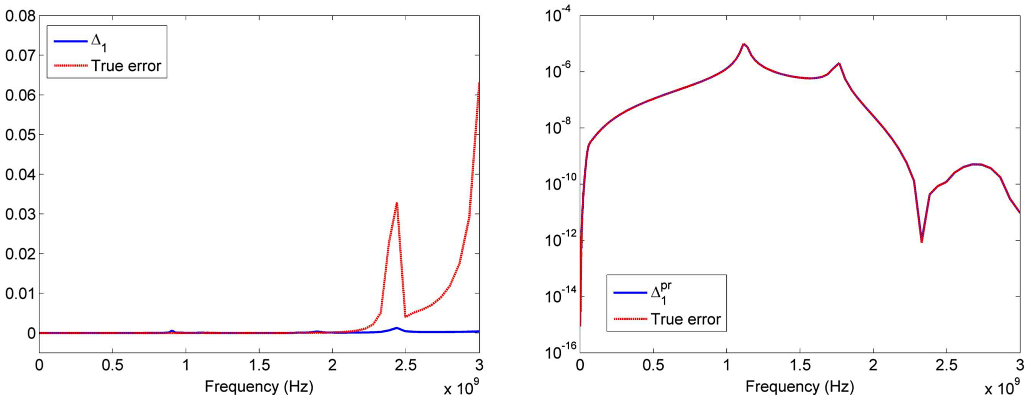

Figure 5. RCLtree: $\Delta_1(s)$ and $\Delta_1^{pr}(s)$ *vs.* the respective true errors at 900 frequency samples.

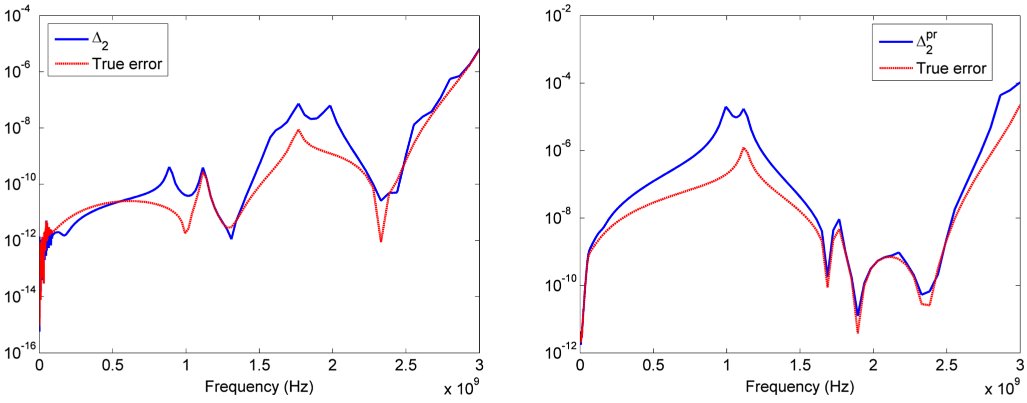

FIGURE 6. RLCtree: $\Delta_2(s)$ and $\Delta_2^{pr}(s)$ *vs.* the respective true errors at 900 frequency samples.

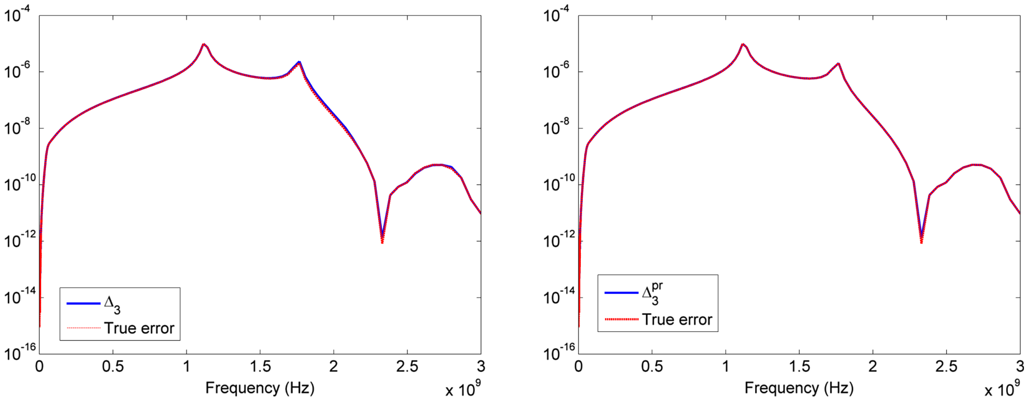

FIGURE 7. RLCtree: $\Delta_3(s)$ and $\Delta_3^{pr}(s)$ *vs.* the respective true errors at 900 frequency samples.

## 7.3. MIMO example

This example has the same frequency range as the second example, therefore we use the same $\Xi$ as for the RLC tree model. The error estimator is the maximal error estimator defined as

$$\Delta(s) = \max_{ij} \Delta_{ij}(s),$$

where $\Delta_{ij}(s)$ estimates the true error $\epsilon_{ij}(s) = |H_{ij}(s) - \hat{H}_{ij}(s)|$. Here $H_{ij}(s)$ and $\hat{H}_{ij}(s)$ are the transfer functions corresponding to the $j$th input port and $i$th output port of the original model and the ROM, respectively. The true error is the maximal true error $\epsilon(s) = \max_{ij} |\epsilon_{ij}(s)|$, and $\epsilon_{\max} = \max_{s\in\Xi} \epsilon(s)$ as defined before.

The results of Algorithm 1 using different error estimators are listed in Tables 9–11. Algorithm 1 stops before the true error $\epsilon_{\max}$ is below the tolerance when using $\Delta_1(s)$, whereas $\Delta_1^{pr}(s)$, $\Delta_3(s)$ and $\Delta_3^{pr}(s)$ exactly estimate the true error at each iteration step. $\Delta_2(s)$ and its primal variation $\Delta_2^{pr}(s)$ produce the same results and make the algorithm converge in 3 iterations. Note that $\Delta_3(s)$ and $\Delta_3^{pr}(s)$ also yield the same results.

The ROMs constructed by Algorithm 1 using the error estimators are further validated over a validation set $\Xi_{\text{ver}}$ with 900 samples, respectively. Table 12 lists the effectivity values of the error estimators. Among them, $\Delta_2(s)$ and its primal variation $\Delta_2^{pr}(s)$ are the best ones and have the same effectivity values. $\Delta_1^{pr}(s)$, $\Delta_3(s)$ and $\Delta_3^{pr}(s)$ have similar results and are still good.

TABLE 9. MIMO example, $\varepsilon_{\text{tol}} = 10^{-3}$, $q = 3$, $r = 20(\Delta_1)$, $r = 52(\Delta_1^{pr})$.

| Iteration $i$ | $\varepsilon_{\max}(\Delta_1)$ | $\Delta_1(s_i)$ | $\varepsilon_{\max}(\Delta_1^{pr})$ | $\Delta_1^{pr}(s_i)$ |
|---|---|---|---|---|
| 1 | 0.28 | $3.16 \times 10^{-5}$ | 0.28 | 0.28 |
| 2 | – | – | $5.91 \times 10^{-5}$ | $5.91 \times 10^{-5}$ |

TABLE 10. MIMO example, $\varepsilon_{\text{tol}} = 10^{-3}$, $q = 3$, $r = 73$.

| Iteration $i$ | $\varepsilon_{\max}(\Delta_2)$ | $\Delta_2(s_i)$ | $\varepsilon_{\max}(\Delta_2^{pr})$ | $\Delta_2^{pr}(s_i)$ |
|---|---|---|---|---|
| 1 | 0.28 | 0.28 | 0.28 | 0.28 |
| 2 | $5.91 \times 10^{-5}$ | $2.3 \times 10^{-3}$ | $5.91 \times 10^{-5}$ | $2.3 \times 10^{-3}$ |
| 3 | $4.72 \times 10^{-8}$ | $1.43 \times 10^{-7}$ | $4.72 \times 10^{-8}$ | $1.43 \times 10^{-7}$ |

TABLE 11. MIMO example, $\varepsilon_{\text{tol}} = 10^{-3}$, $q = 3$, $r = 52$.

| Iteration $i$ | $\varepsilon_{\max}(\Delta_3)$ | $\Delta_3(s_i)$ | $\varepsilon_{\max}(\Delta_3^{pr})$ | $\Delta_3^{pr}(s_i)$ |
|---|---|---|---|---|
| 1 | 0.28 | 0.28 | 0.28 | 0.28 |
| 2 | $5.91 \times 10^{-5}$ | $5.91 \times 10^{-5}$ | $5.91 \times 10^{-5}$ | $5.91 \times 10^{-5}$ |

TABLE 12. MIMO example, effectivity of the error estimators.

| Estimator | For all $\varepsilon(s)$ | | For $\varepsilon(s) \geq 10^{-11}$ | |
|---|---|---|---|---|
| | $\min\limits_{s\in\Xi_{\text{ver}}}$ (eff) | $\max\limits_{s\in\Xi_{\text{ver}}}$ (eff) | $\min\limits_{s\in\Xi_{\text{ver}}}$ (eff) | $\max\limits_{s\in\Xi_{\text{ver}}}$ (eff) |
| $\Delta_1$ | $8.78 \times 10^{-8}$ | 2.53 | $8.78 \times 10^{-8}$ | 1.43 |
| $\Delta_1^{pr}$ | 0.1 | 40 | 0.2 | 26 |
| $\Delta_2$ | 0.1 | 5 | 0.2 | 3.5 |
| $\Delta_2^{pr}$ | 0.1 | 5 | 0.2 | 3.5 |
| $\Delta_3$ | 0.1 | 25 | 0.2 | 21 |
| $\Delta_3^{pr}$ | 0.1 | 28 | 0.2 | 25 |

Figures 8–10 plot the error estimators and the corresponding true errors of the ROMs. The waveforms of the error estimators well reflect the data in Table 12. It is noticed that the maximal true errors over the validation sample set $\Xi_{\text{ver}}$ obtained by $\Delta_1^{pr}(s)$, $\Delta_3(s)$ and $\Delta_3^{pr}(s)$ are still bigger than the error tolerance, though they are exactly reproduced by the error estimators. Since the error estimators accurately measure the maximal true error, the ROMs can be further improved by adding one more expansion point from $\Xi_{\text{ver}}$ (rather than $\Xi$) at which the error estimators are maximal. This will certainly incur more computational costs. Therefore, $\Delta_2(s)$ and $\Delta_2^{pr}(s)$ outperform the other ones for this model.

## 7.4. Parametric example

The micro-gyroscope model is a second-order parametric system with four parameters,

$$\begin{aligned} M(\mu)\ddot{x}(\mu,t) + D(\mu)\dot{x}(\mu,t) + T(\mu)x(\mu,t) &= Bu(t), \\ y(\mu,t) &= Cx(\mu,t). \end{aligned}$$

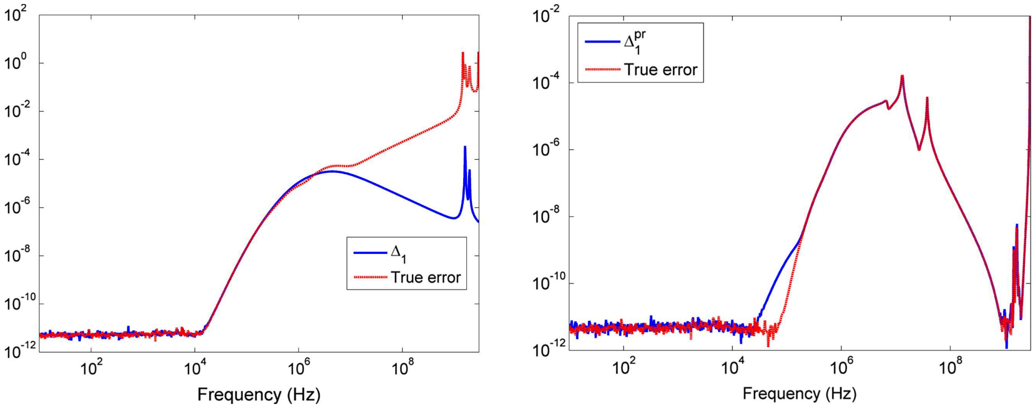

FIGURE 8. MIMO example: $\Delta_1(s)$ and $\Delta_1^{pr}(s)$ *vs.* the respective true errors at 900 frequency samples.

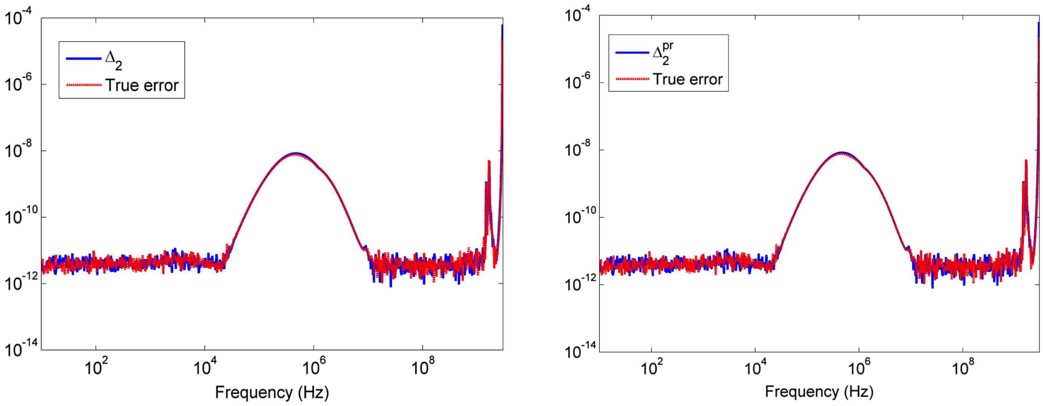

FIGURE 9. MIMO example: $\Delta_2(s)$ and $\Delta_2^{pr}(s)$ *vs.* the respective true errors at 900 frequency samples.

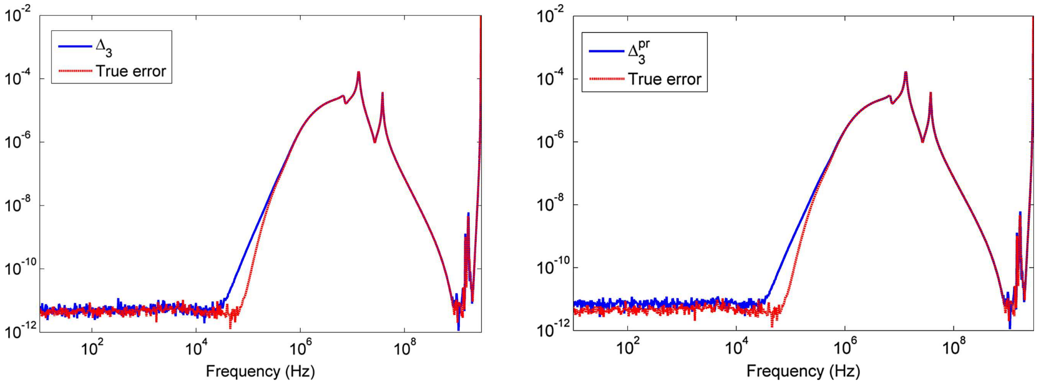

FIGURE 10. MIMO example: $\Delta_3(s)$ and $\Delta_3^{pr}(s)$ *vs.* the respective true errors at 900 frequency samples.

TABLE 13. Gyroscope, $\varepsilon_{\mathrm{tol}} = 10^{-3}$, $q = 1$, $r = 84(\Delta_1)$, $r = 94(\Delta_1^{pr})$.

| Iteration $i$ | $\varepsilon_{\max}(\Delta_1)$ | $\Delta_1(\tilde{\mu}^i)$ | $\varepsilon_{\max}(\Delta_1^{pr})$ | $\Delta_1^{pr}(\tilde{\mu}^i)$ |
|---|---|---|---|---|
| 1 | 0.028 | 0.04 | 0.028 | 0.025 |
| 2 | 0.006 | 0.007 | 0.001 | 0.006 |
| 3 | 0.004 | $3.2 \times 10^{-4}$ | 0.003 | 0.003 |
| 4 | $4 \times 10^{-5}$ | $5.18 \times 10^{-4}$ | $3.85 \times 10^{-4}$ | $3.78 \times 10^{-4}$ |
| 5 | $3.34 \times 10^{-6}$ | $2.99 \times 10^{-5}$ | $1.69 \times 10^{-6}$ | $1.69 \times 10^{-6}$ |
| 6 | $2.95 \times 10^{-7}$ | $3.88 \times 10^{-7}$ | $3.48 \times 10^{-7}$ | $3.47 \times 10^{-7}$ |
| 7 | $7.91 \times 10^{-8}$ | $8.03 \times 10^{-8}$ | $1.39 \times 10^{-7}$ | $1.45 \times 10^{-7}$ |
| 8 | – | – | $8.49 \times 10^{-8}$ | $8.44 \times 10^{-8}$ |

Here, $\mu = (\theta, \alpha, \beta, d)$, $M(\mu) = (M_1 + dM_2)$, $T(\mu) = (T_1 + \frac{1}{d}T_2 + dT_3)$, $D(\mu) = \theta(D_1 + dD_2) + \alpha M(d) + \beta T(d) \in \mathbb{R}^{n\times n}$, $n = 17{,}913$. The parameters are $d, \theta, \alpha, \beta$. $d \in [100\%, 200\%]$, the width of the bearing, taken as the percentage of the base value, and $\theta \in [10^{-7}, 10^{-5}]$MHz, the rotation velocity along the x-axis. $\alpha$, $\beta$ define to the proportional damping [27].

After Laplace transform, the system in frequency domain is

$$\begin{aligned} s^2M(\mu)x(\mu, s) + sD(\mu)x(\mu, s) + T(\mu)x &= Bu_{\mathcal{L}}(s), \\ y(\mu, s) &= Cx(\mu, s). \end{aligned}$$

The above system can be rewritten into the affine form,

$$\begin{aligned} Q(\tilde{\mu})x(\tilde{\mu}) &= Bu_{\mathcal{L}}(\tilde{\mu}), \\ y(\tilde{\mu}) &= Cx(\tilde{\mu}), \end{aligned}$$

where $Q(\tilde{\mu}) = T_1 + \tilde{\mu}_1 M_1 + \tilde{\mu}_2 M_2 + \tilde{\mu}_3 D_1 + \tilde{\mu}_4 D_2 + \tilde{\mu}_5 M_1 + \tilde{\mu}_6 M_2 + \tilde{\mu}_7 T_1 + \tilde{\mu}_8 T_2 + \tilde{\mu}_9 T_3 + \tilde{\mu}_{10} T_2 + \tilde{\mu}_{11} T_3$. Here $\tilde{\mu} = (\tilde{\mu}_1, \ldots, \tilde{\mu}_{11})^T$ includes the newly generated parameters, $\tilde{\mu}_1 = s^2$, $\tilde{\mu}_2 = s^2 d$, $\tilde{\mu}_3 = s\theta$, $\tilde{\mu}_4 = s\theta d$, $\tilde{\mu}_5 = s\alpha$, $\tilde{\mu}_6 = s\alpha d$, $\tilde{\mu}_7 = s\beta$, $\tilde{\mu}_8 = s\beta/d$, $\tilde{\mu}_9 = s\beta d$, $\tilde{\mu}_{10} = 1/d$, $\tilde{\mu}_{11} = d$.

For this example, we use 75 random samples (3 for $\theta$, 5 for $s$, 5 for $d$) to set up the training set $\Xi$ with $\beta = 0$ and $\alpha = 0$. Afterwards, the ROMs are validated at a validation set $\Xi_{\mathrm{ver}}$ including 2500 samples (5 for $\theta$, 10 for $s$, 5 for $d$), with $\beta = 10^{-9}$ and $\alpha = 0.1$ being nonzero.

The results of Algorithm 2 using different error estimators are listed in Tables 13–15. Except for $\Delta_1(\tilde{\mu})$, all the other error estimators tightly estimate the true error at each iteration of the algorithm. The ROMs obtained *via* the error estimators are further validated at samples in $\Xi_{\mathrm{ver}}$, and the effectivity of each is presented in Table 16. Again, $\Delta_1(\tilde{\mu})$ is the worst. The others perform similarly well. We plot the true error of the ROMs and the corresponding error estimators in Figures 11–13. $\Delta_1(\tilde{\mu})$ almost always underestimates the true error, while $\Delta_1^{pr}(\tilde{\mu})$, $\Delta_3(\tilde{\mu})$ and $\Delta_3^{pr}(\tilde{\mu})$ are almost indistinguishable from the true error.

### 7.5. Performances of $\Delta_1(\tilde{\mu})$, $\Delta_2(\tilde{\mu})$ and $\Delta_2^{pr}(\tilde{\mu})$ using Algorithms 3–4

In this subsection, we show the results of Algorithms 3–4 for symmetric systems, where the expansion points for $V_{du}$ are selected differently from those for $V$. The results are listed in Tables 17–19.

Comparing Tables 17–19 with Tables 8, 12, 16, respectively, we see that the performance of $\Delta_1(\tilde{\mu})$ is improved in general, those of $\Delta_2(\tilde{\mu})$, and $\Delta_2^{pr}(\tilde{\mu})$ are only partially improved. The performance of $\Delta_2(\tilde{\mu}))$ is improved, especially for the RLC tree example. However, the performance of $\Delta_2^{pr}(\tilde{\mu})$ does not become uniformly better, especially for the MIMO example. Although $\Delta_1(s)$ behaves better when using Algorithm 3 and 4, it is still worse than its upper bound $\Delta_2(\tilde{\mu})$ or $\Delta_2^{pr}(\tilde{\mu})$.

It is worth pointing out that the order $q = 3$ is used for all the non-parametric examples and all error estimators, while $q = 1$ is used for the parametric system and for all error estimators. Different choices of $q$

Table 14. Gyroscope, $\varepsilon_{\text{tol}} = 10^{-3}$, $q = 1$, $r = 86(\Delta_2)$, $r = 80(\Delta_2^{pr})$.

| Iteration $i$ | $\varepsilon_{\max}(\Delta_2)$ | $\Delta_2(\tilde{\mu}^i)$ | $\varepsilon_{\max}(\Delta_2^{pr})$ | $\Delta_2^{pr}(\tilde{\mu}^i)$ |
|---|---|---|---|---|
| 1 | $4.53 \times 10^{-4}$ | 0.002 | 0.002 | 0.004 |
| 2 | $4.15 \times 10^{-4}$ | $6.16 \times 10^{-4}$ | $4.14 \times 10^{-4}$ | $5.83 \times 10^{-4}$ |
| 3 | $1.71 \times 10^{-5}$ | $8.53 \times 10^{-5}$ | $1.61 \times 10^{-4}$ | $2.69 \times 10^{-4}$ |
| 4 | $8.77 \times 10^{-6}$ | $8.22 \times 10^{-6}$ | $9.7 \times 10^{-5}$ | $1.57 \times 10^{-4}$ |
| 5 | $1.44 \times 10^{-6}$ | $1.07 \times 10^{-6}$ | $9.80 \times 10^{-7}$ | $9.81 \times 10^{-7}$ |
| 6 | $3.09 \times 10^{-8}$ | $3.41 \times 10^{-8}$ | $1.89 \times 10^{-7}$ | $2.06 \times 10^{-7}$ |
| 7 | – | – | $7.21 \times 10^{-8}$ | $8.14 \times 10^{-8}$ |

Table 15. Gyroscope, $\varepsilon_{\text{tol}} = 10^{-3}$, $q = 1$, $r = 73(\Delta_3)$, $r = 83(\Delta_3^{pr})$.

| Iteration $i$ | $\varepsilon_{\max}(\Delta_3)$ | $\Delta_3(\tilde{\mu}^i)$ | $\varepsilon_{\max}(\Delta_3^{pr})$ | $\Delta_3^{pr}(\tilde{\mu}^i)$ |
|---|---|---|---|---|
| 1 | 0.009 | 0.005 | $5.42 \times 10^{-4}$ | 0.002 |
| 2 | 0.009 | 0.005 | $5.60 \times 10^{-4}$ | $5.26 \times 10^{-4}$ |
| 3 | $8.85 \times 10^{-5}$ | $8.85 \times 10^{-5}$ | $9.35 \times 10^{-5}$ | $6.59 \times 10^{-4}$ |
| 4 | $2.20 \times 10^{-4}$ | $2.20 \times 10^{-4}$ | $5.36 \times 10^{-6}$ | $5.36 \times 10^{-6}$ |
| 5 | $1.78 \times 10^{-6}$ | $1.48 \times 10^{-6}$ | $1.31 \times 10^{-6}$ | $1.30 \times 10^{-6}$ |
| 6 | $8.56 \times 10^{-8}$ | $8.51 \times 10^{-8}$ | $5.78 \times 10^{-7}$ | $5.78 \times 10^{-7}$ |
| 7 | – | – | $5.60 \times 10^{-8}$ | $5.59 \times 10^{-8}$ |

Table 16. Gyroscope, effectivity of the error estimators.

| Estimator | For all $\varepsilon(s)$ | | For $\varepsilon(s) \geq 10^{-11}$ | |
|---|---|---|---|---|
| | $\min\limits_{s\in\Xi_{\text{ver}}}$ (eff) | $\max\limits_{s\in\Xi_{\text{ver}}}$ (eff) | $\min\limits_{s\in\Xi_{\text{ver}}}$ (eff) | $\max\limits_{s\in\Xi_{\text{ver}}}$ (eff) |
| $\Delta_1$ | 0.025 | 8.87 | 0.025 | 8.87 |
| $\Delta_1^{pr}$ | 0.2 | 3.65 | 0.2 | 3.65 |
| $\Delta_2$ | 0.38 | 15 | 0.38 | 15 |
| $\Delta_2^{pr}$ | 0.2 | 3.68 | 0.2 | 3.68 |
| $\Delta_3$ | 0.34 | 9.34 | 0.34 | 9.34 |
| $\Delta_3^{pr}$ | 0.5 | 2 | 0.5 | 2 |

Table 17. Algorithm 3: RLCtree, effectivity of the error estimators.

| Estimator | For all $\varepsilon(s)$ | | For $\varepsilon(s) \geq 10^{-11}$ | |
|---|---|---|---|---|
| | $\min\limits_{s\in\Xi_{\text{ver}}}$ (eff) | $\max\limits_{s\in\Xi_{\text{ver}}}$ (eff) | $\min\limits_{s\in\Xi_{\text{ver}}}$ (eff) | $\max\limits_{s\in\Xi_{\text{ver}}}$ (eff) |
| $\Delta_1$ | $3.4488 \times 10^{-4}$ | 38 | 0.05 | 6.5 |
| $\Delta_2$ | 0.01 | 25 | 0.7 | 25 |
| $\Delta_2^{pr}$ | 0.004 | 244 | 1 | 25 |

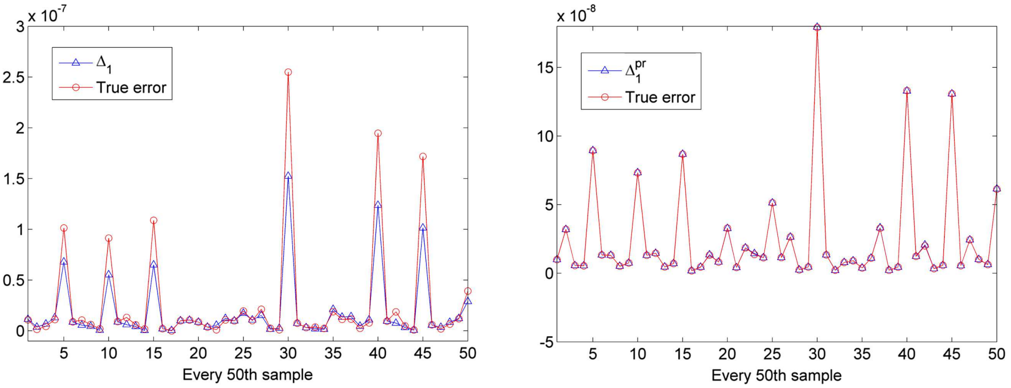

FIGURE 11. Gyroscope: $\Delta_1(\tilde{\mu})$ and $\Delta_1^{pr}(\tilde{\mu})$ *vs.* the respective true errors at 2500 parameter samples.

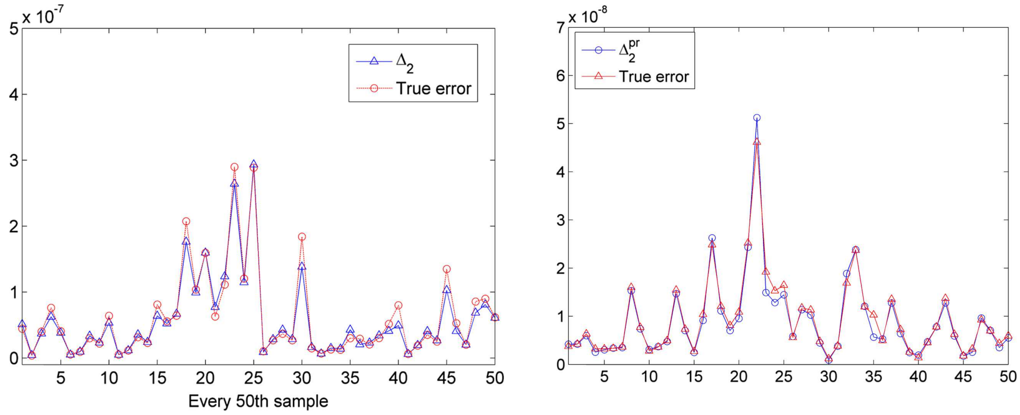

FIGURE 12. Gyroscope: $\Delta_2(\tilde{\mu})$ and $\Delta_2^{pr}(\tilde{\mu})$ *vs.* the respective true errors at 2500 parameter samples.

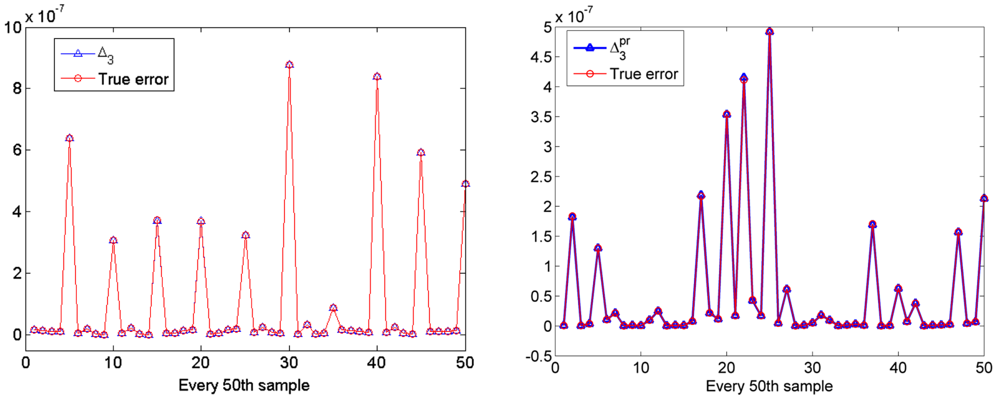

FIGURE 13. Gyroscope: $\Delta_3(\tilde{\mu})$ and $\Delta_3^{pr}(\tilde{\mu})$ *vs.* the respective true errors at 2500 parameter samples.

TABLE 18. Algorithm 3: MIMO example, effectivity of the error estimators.

| Estimator | For all $\varepsilon(s)$ | | For $\varepsilon(s) \geq 10^{-11}$ | |
|---|---|---|---|---|
| | $\min_{s\in\Xi_{\text{ver}}}(\text{eff})$ | $\max_{s\in\Xi_{\text{ver}}}(\text{eff})$ | $\min_{s\in\Xi_{\text{ver}}}(\text{eff})$ | $\max_{s\in\Xi_{\text{ver}}}(\text{eff})$ |
| $\Delta_1$ | 0.14 | 46 | 0.14 | 46 |
| $\Delta_2$ | 0.2 | 15 | 0.1 | 9 |
| $\Delta_2^{pr}$ | 0.32 | 164 | 0.32 | 75 |

TABLE 19. Algorithm 4: Gyroscope, effectivity of the error estimators.

| Estimator | For all $\varepsilon(s)$ | | For $\varepsilon(s) \geq 10^{-11}$ | |
|---|---|---|---|---|
| | $\min_{s\in\Xi_{\text{ver}}}(\text{eff})$ | $\max_{s\in\Xi_{\text{ver}}}(\text{eff})$ | $\min_{s\in\Xi_{\text{ver}}}(\text{eff})$ | $\max_{s\in\Xi_{\text{ver}}}(\text{eff})$ |
| $\Delta_1$ | 0.096 | 28 | 0.096 | 28 |
| $\Delta_2$ | 0.35 | 11 | 0.35 | 11 |
| $\Delta_2^{pr}$ | 0.22 | 3.68 | 0.22 | 3.68 |

may lead to different performance of the greedy algorithms employing different error estimators. In the next subsection, we use the MIMO example and the parametric example to show the performance of Algorithm 1 and Algorithm 2 using the proposed error estimators when the order of moments $q$ varies. In particular, we present the iteration numbers and the wall-clock time of each algorithm for different $q$, corresponding to each error estimator.

### 7.6. Performance of the error estimators when $q$ varies

We list in Tables 20 and 21 the number of iterations and wall-clock time of Algorithm 1 for the MIMO example as well as those of Algorithm 2 for the parametric example, when different error estimators are used and $q$ varies. For the MIMO example, when $q$ increases, the number of iterations may decrease, but the runtime of each error estimator increases in general. For a fixed $q$, except for $\Delta_1(s)$ that uses much less runtime, the other estimators make the algorithm converge without too much difference in time. Not a single estimator is always the best. However, $\Delta_1(s)$ does not lead to a reliable ROM, since it underestimates the true error also for difference values of $q$ and for both examples. For the parametric example, there is a different phenomenon for $q = 0$, where Algorithm 2 actually does not converge when using many of the error estimators. The error estimator as well as the corresponding true errors, do not go below the tolerance after all the 75 training parameter samples have been chosen as expansion points. Therefore, in the second column of Table 21, there is 75 for many error estimators, except for $\Delta_3(\tilde{\mu})$. For $q = 1, 2$, we have similar observations: not a single estimator is always the best. However, it can be concluded that matching too few moments ($q = 0$) or too many moments ($q = 2$) for the parametric model lead to no better behavior of the greedy algorithm than using $q = 1$ in general. Furthermore, using $q = 2$ derives a ROM with more than twice the size of the ROM derived using $q = 1$. Larger ROM will make the online simulation much slower. Since this work does not focus on discussing the efficiency of the greedy algorithm combined with the moment-matching method, which is one of the possible ways of showing the robustness of the proposed estimators, we do not further elaborate on this discussion. Finally, each error estimator has similar effectivity as shown in the previous tables when $q$ changes. The corresponding effectivities are not listed here to avoid repetitions.

TABLE 20. Performance of Algorithm 1 for the MIMO example when $q$ varies.

| Estimator | $q=0$ | | $q=3$ | | $q=5$ | |
|---|---|---|---|---|---|---|
| | Iter | Time(s) | Iter | Time(s) | Iter | Time(s) |
| $\Delta_1$ | 1 | 1.97 | 1 | 4.52 | 1 | 7.84 |
| $\Delta_1^{pr}$ | 4 | 9.48 | 2 | 16.68 | 2 | 31.63 |
| $\Delta_2$ | 5 | 14.73 | 3 | 28.45 | 2 | 26.56 |
| $\Delta_2^{pr}$ | 5 | 14.58 | 3 | 27.93 | 2 | 26.94 |
| $\Delta_3$ | 5 | 14.78 | 2 | 18.31 | 2 | 31.27 |
| $\Delta_3^{pr}$ | 4 | 12.29 | 2 | 27.23 | 2 | 41.56 |

TABLE 21. Performance of Algorithm 2 for the parametric example when $q$ varies.

| Estimator | $q=0$ | | $q=1$ | | $q=2$ | |
|---|---|---|---|---|---|---|
| | Iter | Time | Iter | Time | Iter | Time |
| $\Delta_1$ | 75 | 2075 | 6 | 256.8 | 2 | 423.6 |
| $\Delta_1^{pr}$ | 75 | 2121 | 6 | 349.9 | 2 | 369.4 |
| $\Delta_2$ | 75 | 2454 | 6 | 312.3 | 3 | 994.8 |
| $\Delta_2^{pr}$ | 75 | 2824 | 6 | 385.4 | 3 | 910.7 |
| $\Delta_3$ | 17 | 442.2 | 6 | 396.5 | 3 | 664.3 |
| $\Delta_3^{pr}$ | 75 | 2663 | 6 | 479.7 | 2 | 599.7 |

## 8. CONCLUSIONS

We propose some a posteriori error estimators for the transfer function error of ROMs that are obtained by any (Petrov-)Galerkin-type MOR method. Detailed simulation comparison demonstrates the performance of each. It is clear that neither $\Delta_r(\tilde{\mu})$ nor $\Delta_1(\tilde{\mu})$ are good error estimators and therefore they are not recommended as reliable error estimators. All others perform similarly, especially the primal version of $\Delta_1(\tilde{\mu})$: $\Delta_1^{pr}(\tilde{\mu})$ behaves unexpectedly well and is almost as good as its bounds $\Delta_3(\tilde{\mu})$ and $\Delta_3^{pr}(\tilde{\mu})$ for any of the examples. Among the robust error estimators $\Delta_2(\tilde{\mu})$, $\Delta_2^{pr}(\tilde{\mu})$, $\Delta_1^{pr}(\tilde{\mu})$, $\Delta_3(\tilde{\mu})$ and $\Delta_3^{pr}(\tilde{\mu})$, the estimator $\Delta_1^{pr}$ needs the least computational cost, since only two ROMs (constructed by $V, V_{r_{pr}}$) need to be computed. For nearly symmetric systems, $\Delta_2(\tilde{\mu})$ and its variant $\Delta_2^{pr}(\tilde{\mu})$ are not really improved when choosing different expansion points for $V$ and $V_{du}$, *i.e.*, when using Algorithms 3 and 4. As future work, more theoretical analysis and numerical simulations might be explored to further explain the numerical behaviors of the proposed error estimators.

## REFERENCES

[1] A.C. Antoulas, Approximation of Large-Scale Dynamical Systems. In: Vol. 6 of *Adv. Des. Control.* SIAM Publications, Philadelphia, PA (2005).

[2] A.C. Antoulas and D.C. Sorensen, Approximation of large-scale dynamical systems: an overview. *Int. J. Appl. Math. Comput. Sci.* **11** (2001) 1093–1121.

[3] A.C. Antoulas, D.C. Sorensen and S. Gugercin, A survey of model reduction methods for large-scale systems. *Contemp. Math.* **280** (2001) 193–219.

[4] A.C. Antoulas, C.A. Beattie and S. Gugercin, Interpolatory model reduction of large-scale dynamical systems. In: Efficient Modeling and Control of Large-Scale Systems, edited by J. Mohammadpour and K.M. Grigoriadis. Springer US (2010) 3–58.

[5] A.C. Antoulas, C. Beattie and S. Gugercin, Interpolatory methods for model reduction. In: Vol. 6 of *Computational Science and Engineering 21.* Philadelphia, PA (2020).

[6] U. Baur, C.A. Beattie, P. Benner and S. Gugercin, Interpolatory projection methods for parameterized model reduction. *SIAM J. Sci. Comput.* **33** (2011) 2489–2518.

[7] 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) 331–358.

[8] U. Baur, P. Benner, B. Haasdonk, C. Himpe, I. Martini and M. Ohlberger, Comparison of methods for parametric model order reduction of time-dependent problems. In: Model Reduction and Approximation: Theory and Algorithms, edited by P. Benner, A. Cohen, M. Ohlberger and K. Willcox. SIAM (2017) 377–407.

[9] P. Benner, M. Hinze and E.J.W. ter Maten, editors. Model reduction for circuit simulation. In: Vol. 74 of *Lect. Notes Electr. Eng.* Springer, Dodrecht (2011).

[10] P. Benner, S. Gugercin and K. Willcox, A survey of model reduction methods for parametric systems. *SIAM Rev.* **57** (2015) 483–531.

[11] P. Benner, A. Cohen, M. Ohlberger and K. Willcox, editors. Model Reduction and Approximation: Theory and Algorithms. *Computational Science & Engineering.* SIAM Publications, Philadelphia, PA (2017).

[12] F. Chinesta, R. Keunings and A. Leygue, The Proper Generalized Decomposition for Advanced Numerical Simulations. *SpringerBriefs in Applied Sciences and Technology.* Springer International Publishing (2014).

[13] V. Druskin and V. Simoncini, Adaptive rational Krylov subspaces for large-scale dynamical systems. *Syst. Control Lett.* **60** (2011) 546–560.

[14] V. Druskin, V. Simoncini and M. Zaslavsky, Adaptive tangential interpolation in rational Krylov subspaces for MIMO dynamical systems. *SIAM J. Matrix Anal. Appl.* **35** (2014) 476–498.

[15] L. Feng and P. Benner, A robust algorithm for parametric model order reduction based on implicit moment matching, edited by A. Quarteroni and G. Rozza. In: Vol. 9 of *Reduced Order Methods for Modeling and Computational Reduction, Chap. 6, MS&A Series.* Springer-Verlag, Berlin, Heidelberg, NY (2014) 159–186.

[16] L. Feng and P. Benner, A new error estimator for reduce-order modeling of linear parametric systems. *IEEE Trans. Microw. Theory Techn.* **67** (2019) 4848–4859.

[17] L. Feng, A.C. Antoulas and P. Benner, Some a posteriori error bounds for reduced order modelling of (non-)parametrized linear systems. *ESAIM: M2AN* **51** (2017) 2127–2158.

[18] G. Fotyga, M. Czarniewska, A. Lamecki and M. Mrozowski, Reliable greedy multipoint model-order reduction techniques for finite-element analysis. *IEEE Trans. Antennas Propagat.* **17** (2018) 821–824.

[19] A. Grimm, *Parametric dynamical systems: Transient analysis and data driven modeling.* Ph.D. thesis, Virginia Tech (2018).

[20] S. Hain, M. Ohlberger, M. Radic and K. Urban, A hierarchical a-posteriori error estimator for the reduced basis method. Cornell University. Preprint: arXiv:1802.03298 (2018).

[21] M.W. Hess, S. Grundel and P. Benner, Estimating the inf-sup constant in reduced basis methods for time-harmonic Maxwell's equations. *IEEE Trans. Microw. Theory Techn.* **63** (2015) 3549–3557.

[22] J.S. Hesthaven, G. Rozza and B. Stamm, Certified Reduced Basis Methods for Parametrized Partial Differential Equations. *SpringerBriefs in Mathematics.* Springer, Cham (2016).

[23] M. Hund, P. Mlinarić and J. Saak, An $\mathcal{H}_2 \otimes \mathcal{L}_2$-optimal model order reduction approach for parametric linear time-invariant systems. *Appl. Math. Mech.* **18** (2018) e201800084.

[24] D. Huynh, G. Rozza, S. Sen and A. Patera, A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup lower bounds. *C.R. Acad. Sci. Paris* **345** (2007) 473–478.

[25] A. Monje-Real and V. de la Rubia, Electric field integral equation for fast frequency sweep for scattering of nonpenetrable objects *via* the reduced-basis method. *IEEE Trans. Antennas Propagat.* **68** (2020) 6232–6244.

[26] G. Rozza, D.B.P. Huynh and A.T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Arch. Comput. Methods Eng.* **15** (2008) 229–275.

[27] B. Salimbahrami, R. Eid and B. Lohmann, Model reduction by second order Krylov subspaces: extensions, stability and proportional damping. In: *IEEE Conference on Computer Aided Control Systems Design.* Springer International Publishing (2006) 2997–3002.

[28] A. Schmidt, D. Wittwar and B. Haasdonk, Rigorous and effective a-posteriori error bounds for nonlinear problems-application to RB methods. SimTech, University of Stuttgart (2018).

[29] K. Smetana, O. Zahm and A.T. Patera, Randomized residual-based error estimators for parametrized equations. *SIAM J. Sci. Comput.* **41** (2019) A900–A926.

[30] M. Yano, A space-time Petrov-Galerkin certified reduced basis method: application to the Boussinesq equations. *SIAM J. Sci. Comput.* **36** (2014) 232–266.

[31] Y. Zhang, L. Feng, S. Li and P. Benner, An efficient output error estimation for model order reduction of parametrized evolution equations. *SIAM J. Sci. Comput.* **37** (2015) B910–B936.