Applications

Jens Saak*, Dirk Siebelts and Steffen W. R. Werner

A comparison of second-order model order reduction methods for an artificial fishtail

Ein Vergleich von Modellordnungsreduktionsverfahren zweiter Ordnung für einen künstlichen Fischschwanz

https://doi.org/10.1515/auto-2019-0027
Received February 15, 2019; accepted May 15, 2019

Abstract: In order to apply control theory in small autonomous vehicles, mathematical models with small numbers of states are required for using the limited computational power in embedded programming. In this paper, we consider an artificial fishtail as an example for a complex mechanical system with a second-order large-scale model, which is derived by using the finite element method. To meet the above limitations, the several hundreds of thousands of degrees of freedom need to be reduced to merely a handful of surrogate degrees of freedom.

We seek to achieve this task by various second-order model order reduction methods. All methods are applied on the fishtail's matrices and their results are evaluated and compared in the frequency domain as well as in the time domain.

Keywords: model order reduction, artificial fishtail, balanced truncation, moment matching, modal approximation

Zusammenfassung: Zur Anwendung von Methoden der Regelungs- und Steuerungstheorie in kleinen autonomen Fahrzeugen sind mathematische Modelle mit wenigen Zustandsvariablen erforderlich, um der begrenzten Rechenleistung der eingebetteten Systeme gerecht zu werden. In diesem Beitrag betrachten wir einen künstlichen Fischschwanz als Beispiel für ein komplexes mechanisches System mit einem Originalmodell zweiter Ordnung, welches mit Hilfe der Finite-Elemente-Methode aufgestellt wird. Um die oben genannten Einschränkungen zu erfüllen, müssen die mehreren hunderttausend Freiheitsgrade auf so wenig, wie möglich Ersatzfreiheitsgrade reduziert werden.

Diese Aufgabe wollen wir durch verschiedene Methoden zur Ordnungsreduktion der Systemmodelle zweiter Ordnung erfüllen. Alle Verfahren werden auf die veröffentlichten Matrizen des Fischschwanzmodells angewandt und ihre Ergebnisse sowohl im Frequenz-, als auch im Zeitbereich ausgewertet und verglichen.

Schlagwörter: Modellordnungsreduktion, Strukturerhalt, Frequenzbereich, Zeitbereich, Methodenvergleich, Bionik

1 Introduction

In order to monitor the environment and for mapping or inspection tasks under water [1], autonomous underwater vehicles (AUVs) are indispensable nowadays. The AUVs are mostly driven by thrusters, which can disturb other underwater creatures with acoustic emissions. Supplementary, they have been shown as inefficient [2] and mostly expensive [3]. Since the evolution of fish shows that the shape is nearly optimized and the way of locomotion allows an agile, fast and efficient movement in the underwater environment [2, 4, 5], producing fish-like underwater vehicles, is obvious.

Approaches from [2] and [6] outline the improvements of the fish-like actuation compared to conventional actuation systems with propulsors. Many fish-like robots (robotic fishes) like, e. g., [7], [8] and [9] have been developed using a finite number of joints, connected to servo motors, inside of the fishtail. Different actuation approaches like [10], with a centralized actuation principle, or [11], using shape memory alloys, have been developed over time. This work uses the fish design and mathematical model, introduced in [12], based on the approach from [13], where the whole is a composition of elastic ma-

***Corresponding author: Jens Saak,** Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39106 Magdeburg, Germany, e-mail: saak@mpi-magdeburg.mpg.de
Dirk Siebelts, Chair of Automatic Control, Faculty of Engineering, Kiel University, Kaiserstraße 2, 24143 Kiel, Germany, e-mail: disi@tf.uni-kiel.de
Steffen W. R. Werner, Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39106 Magdeburg, Germany, e-mail: werner@mpi-magdeburg.mpg.de

terials and the fluid actuator is distributed all over the fishtail.

The mathematical model obtained through, e. g., the finite element method (FEM) leads to high system dimensions and, as a consequence, to a high computational effort. In order to enable the model for control and state estimation under real time conditions on the limited hardware of an AUV, model order reduction is unavoidable.

The main purpose of this endeavor is the comparison of available (especially as public code package) model order reduction methods for second-order systems on this type of model. Here, we especially want to compare methods with a global (in time and frequency domains) perspective, such as the several flavors of balanced truncation for second-order systems [14, 15, 16, 17], or (dominant pole based) modal truncation, with more localized methods focusing on the known range of operation frequencies of about 0–10 Hz. The later methods range from simple moment matching (see, e. g., [18, 19, 20]) approaches with expansion frequencies in this exact range, over two-step, or hybrid methods [21, 22] using a moment matching pre-reduction and balancing-based post-processing, to frequency-limited balancing [23, 24].

Our results show that methods, limiting themselves to the frequency range of interest, are superior in approximation quality. Therefore, we exclude a few methods from our comparison. The second-order iterative rational Krylov algorithm, e. g., [25], tackles the $\mathcal{H}_2$-norm optimal approximation problem. While there has been some work on frequency-limited [26, 27] and frequency-weighted [28, 29] $\mathcal{H}_2$-optimal reduction for standard state-space models of first-order, we are not aware of a structure-preserving second-order adaption. Moreover, we are not aware of any stable public implementation. For similar reasons, we do not consider the Hamiltonian-based approach of [30]. We also do not consider any training-based reduction methods, such as the proper orthogonal decomposition (e. g., [19]).

Regarding model reduction software, we focus on two open source projects developed for MATLAB® and GNU Octave. The M-M.E.S.S. toolbox [31] with main focus on the underlying large and sparse matrix equations and the MORLAB toolbox [32], a software package for model order reduction of moderate size, densely populated systems. There are two prominent packages that deliver similar functionality in MATLAB. MOREMBS [33, 34] is a mainly proprietary tool, that for the second-order balancing features uses the same algorithms as M-M.E.S.S.. The hybrid or two-step approaches presented here, are actually adaptions of methods frequently used therein. The second package, that is especially appealing for control engineers, is the sssMOR [35] toolbox, that lifts many features of MATLAB's Control Systems Toolbox™ functionality to large and sparse systems using again M-M.E.S.S. as a back-end for the matrix equation solvers. So one should be able to achieve similar results with those packages. Also for modeling in ANSYS, the MOR4ANSYS tool [36] provides a small selection of the discussed methods.

The structure of our manuscript is as follows. In the following section, we introduce the AUV that serves as the test case for our comparative study. Section 3 then repeats some notation for the model order reduction problem, recapitulates the existing methods and introduces our new contributions and modifications. The actual comparison in terms of the reduction processes and results, evaluated in both frequency and time domains, is subject of the numerical results in Section 4. Section 5 summarizes our findings.

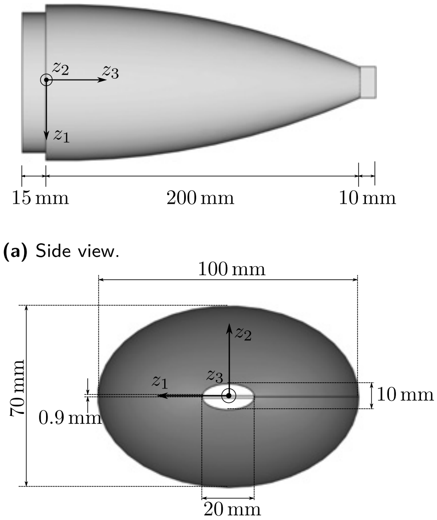
2 The artificial fishtail

The artificial fishtail has been originally introduced and modeled in [12] using the approach by [13]. In order to apply the model order reduction approach on the fishtail, the mathematical model is a crucial factor for the results. Based on [12], the design of the fishtail will be described and a mathematical model will be derived.

2.1 Design

The fishtail is designed using the computer-aided design software FreeCAD [37] and consists of two different materials, a smooth silicone hull generated by three ellipses and a carbon center beam. Fig. 1a shows the fishtail and its dimensions in the z_1z_3-plane. The silicone fish hull is 200 mm long and at its thickest point 100 mm high and 70 mm thick. It narrows to an ellipse with a height of 20 mm and a width of 10 mm at the end. Fig. 1b shows the rear view of the fishtail with the related dimensions. A 15 mm long mounting part is attached to the front end instead of a fishhead. The diameter of the attached silicone is 10 mm smaller compared to the fishtail. The 0.9 mm wide carbon beam is placed in the center of the silicon hull to increase the stability against torsion movements. In longitudinal direction, the center beam starts at the beginning of the mounting part and stops 10 mm after the hull ends, such that the protruding part of the center beam enables the possibility for attaching a fin.

Two separated fluid chamber systems, designed after the idea of [13], are the basis of the actuation. The actu-



(a) Side view.

(b) Rear view.

Figure 1: Sketch of the fishtail with dimensions.

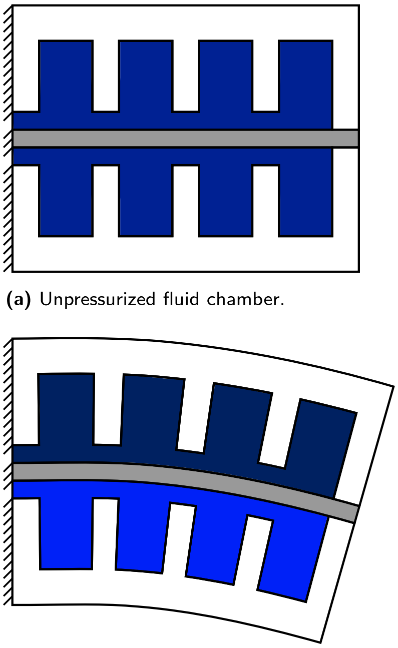
ation principle relies on the expansion of fluid in elastic chambers, where the pressurization (additional to the atmospheric pressure) of a single chamber results in an expansion of the related chamber in the elastic material. The mounted end of the fishtail translates the expansion into a bending movement of the fishtail. Due to the symmetric placement of the chamber systems, an antisymmetric actuation ($u_2(t) = -u_1(t)$) of the second chamber increases the bending effect. Fig. 2 shows the actuation principle for the relaxed state (Fig. 2a) with $u_2(t) = u_1(t) = 0$ and the actuated state (Fig. 2b) for $u_1(t) > 0$ and $u_2(t) < 0$. In both figures, the gray area indicates the center beam, while the blue area represents the fluid in the chamber. The enclosing area is the silicone hull of the fishtail. Unfortunately, the actuation method requires a depressurization below the atmospheric pressure, which requires a vacuum pump and would lead to a large experimental setup. In order to achieve an autonomous fishtail, the chambers systems are pre-pressurized by u_{pre}, such that

$$u_{1,2}(t) = u_{\text{pre}} \pm u(t)$$

holds and only a single pressure supply is required. Applying this actuation principle results in a bounded input

$$0 \leq u(t) \leq u_{\text{pre}}, \qquad \text{for all } t \geq 0.$$

The fluid chambers are placed on both sides (in 2 mm distance) of the carbon center beam and have a defined distance to the outer hull. Fig. 3 shows the fishtail in the $z_1 z_3$-plane with the embedded fluid chamber systems. Each fluid chamber system consists of 24 chambers, which are 4 mm thick.



(a) Unpressurized fluid chamber.

(b) Actuated system.

Figure 2: Fluid chamber in the relaxed and the pressurized state.

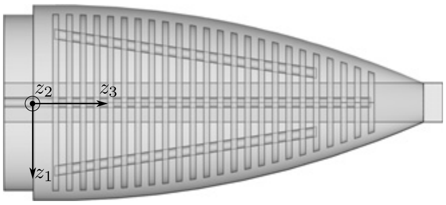


Figure 3: Transparent sketch of the fishtail with embedded fluid chambers.

2.2 The digital realization

Actuating the fishtail results in a smooth bending of the whole structure. Thus a partial differential equation (PDE) is required for describing the behavior in space z and time t. Due to the complex geometrical structure, the FEM is used for the mathematical description. It approximates the structure by simple shapes like tetrahedrons and splits the time and space depending behavior locally into a time dependent trial function and a spatial depending shape function using a weighted residual method. The superposition of the trial functions is the global approximation of the partial differential equation [38].

Assuming that the silicon hull as well as the carbon center beam behave like a linear elastic homogeneous and isotropic material, all equations in this section hold for both areas with different material parameters. The theory of linear elastic materials is based on two constitutive equations for the relations between the displacement $s(t,z)$, the strain $\underline{\varepsilon}(t,z)$ and the stress $\underline{\sigma}(t,z)$ of the structure; see [39]. With the kinematic equation for homogeneous materials

$$\underline{\varepsilon}(t,z) = \frac{1}{2}(\nabla s(t,z) + \nabla^T s(t,z)),$$

a relation between the symmetric second-order strain tensor and the first-order displacement tensor is defined [39]. The relation between the strain and the stress for isotropic materials is given by

$$\underline{\sigma}(t,z) = \lambda \mathrm{trace}\,(\underline{\varepsilon}(t,z))\,\underline{I} + 2\mu\underline{\varepsilon}(t,z),$$

see [39], with the second-order identity tensor $\underline{I}$ and the Lamé parameters

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}, \quad \mu = \frac{E}{(2+2\nu)}.$$

The Lamé parameters are expressed by the elasticity module E and the Poisson ratio ν; see [39].

The second time derivative of the displacement, for $t > 0$, $z \in \Omega$ and initial data $s(0,z) = s_0(z)$, can be described by the divergence of the stress tensor

$$\rho\frac{\partial^2 s(t,z)}{\partial t^2} = \nabla \cdot \underline{\sigma}(t,z), \tag{1}$$

where ρ is the constant density of a material with a homogeneous mass distribution [39]. The volume Ω, for which the equation of motion is defined, can be divided in the volume of the silicone hull Ω_1 and the volume of the carbon center beam Ω_2. Regarding the whole volume, $\Omega = \Omega_1 \cup \Omega_2$ and $\emptyset = \Omega_1 \cap \Omega_2$ have to hold.

Analogously, the boundary surfaces can be partitioned into the boundary of the silicone hull Γ_1 and the boundary of the carbon center beam Γ_2. Assuming a perfect connection between the carbon beam and the hull, no mathematical boundary condition between the two materials exists ($\emptyset = \Gamma_1 \cap \Gamma_2$). Considering that the mounting part of the fishtail is clamped, a displacement of the boundary surface area $\Gamma_{1,1} \subset \Gamma_1 \subset \Gamma$ is not possible, such that there exists a Dirichlet boundary condition

$$s(t,z) = 0, \qquad z \in \Gamma_{1,1}.$$

For a homogeneous pressure distribution inside of a single fluid chamber system, the pressure is equal to the stress in the normal direction (n)

$$\underline{\sigma}(t,z) \cdot n = u_{\mathrm{inp},k}(t), \qquad z \in \Gamma_{1,k+1},\ k \in \{1,2\},$$

where $\Gamma_{1,2}$ and $\Gamma_{1,3}$ are the boundary surface areas of the two fluid chambers. On the remaining boundary surfaces

$$\underline{\sigma}(t,z) \cdot n = 0, \qquad z \in \Gamma_2 \cup (\Gamma_1 \backslash (\Gamma_{1,1} \cap \Gamma_{1,2} \cap \Gamma_{1,3}))$$

has to hold, because no pressure is applied [39].

In order to use the FEM, the weak formulation of equation (1) is required; see [38]. The weak formulation can be obtained by multiplying the equation of motion (in residual form) with a continuous function $v(z)$ and integrating over the whole domain

$$0 = \sum_{i=1}^{2} \int\limits_{\Omega_i} v^T(z)\left(\nabla \cdot \underline{\sigma}(t,z) - \rho_i \frac{\partial^2 s(t,z)}{\partial t^2}\right) \mathrm{d}\Omega, \tag{2}$$

whereby $v(z)$ fulfills the same Dirichlet boundary conditions as $s(t,z)$ [38]. Applying the divergence theorem [40] reformulates equation (2) into

$$\begin{aligned} 0 = &\sum_{i=1}^{2} \int\limits_{\Omega_i} \left(-\nabla v(z) \colon \underline{\sigma}(t,z) - v^T(z)\rho_i \frac{\partial^2 s(t,z)}{\partial t^2}\right) \mathrm{d}\Omega \\ &+ \sum_{i=1}^{2} \int\limits_{\Gamma_i} v^T(z)\,(\underline{\sigma}(t,z) \cdot n)\ \mathrm{d}\Gamma. \end{aligned}$$

Here $\underline{a}\colon\underline{b}$ indicates the double contraction of the two second-order tensors $\underline{a}$ and $\underline{b}$ [40].

Inserting the previously described boundary conditions and taking into account that $v(z)$ fulfills the Dirichlet boundary condition on $\Gamma_{1,1}$ yields

$$\begin{aligned} 0 = &\sum_{i=1}^{2} \int\limits_{\Omega_i} \left(-\nabla v(z) \colon \underline{\sigma}(t,z) - v^T(z)\rho_i \frac{\partial^2 s(t,z)}{\partial t^2}\right) \mathrm{d}\Omega \\ &+ \sum_{i=1}^{2} \int\limits_{\Gamma_{1,i+1}} v^T(z) u_{\mathrm{inp},i}(t)\ \mathrm{d}\Gamma. \end{aligned}$$

Under the assumption that the pressure acts in the negative normal direction of the surface, the actuation boundary condition simplifies to

$$\underline{\sigma}(t,z) \cdot n = -n u_k(t), \qquad z \in \Gamma_{1,k+1},\ k \in \{1,2\}.$$

With $u_{\mathrm{inp},1} = -u_{\mathrm{inp},2}$ the weak formulation can be expressed as

$$\begin{aligned} 0 = &\sum_{i=1}^{2} \int\limits_{\Omega_i} \left(-\nabla v(z) \colon \underline{\sigma}(t,z) - v^T(z)\rho_i \frac{\partial^2 s(t,z)}{\partial t^2}\right) \mathrm{d}\Omega \\ &- \sum_{i=1}^{2} (-1)^{i+1} \int\limits_{\Gamma_{1,i+1}} v^T(z) n u(t)\ \mathrm{d}\Gamma, \end{aligned}$$

where $u(t) \in \mathbb{R}$ is the input of the fishtail.

In order to apply the FEM, the previously introduced CAD model is subdivided into tetrahedral elements using the meshing tool Trelis [41]. The finite element tool Firedrake [42, 43, 44, 45, 46, 47, 48, 49] assembles the matrices by approximating the weak form using a weighted residual method. The time and space dependency of the weak form is separated, such that the time dependency can be factorized [38]. After assembly, FEM results in the second-order ordinary differential equation (ODE)

$$M\ddot{x}(t) + Kx(t) = B_u u(t), \tag{3}$$

consisting of the mass matrix $M \in \mathbb{R}^{n\times n}$, the stiffness matrix $K \in \mathbb{R}^{n\times n}$, the input vector $B_u \in \mathbb{R}^n$ and the state vector $x(t) \in \mathbb{R}^n$ for $t > 0$ with initial state $x(0) = x_0$.

Applying a pulse like input to the resulting ODE causes an infinite oscillation of the state, since no damping has been considered in the mathematical model. In order to obtain a realistic movement of the fishtail, the velocity proportional Rayleigh damping [50] is used

$$E = \alpha_{\mathrm{r}} M + \beta_{\mathrm{r}} K. \tag{4}$$

Inserting the damping into equation (3) results in the final full system representation

$$M\ddot{x}(t) + E\dot{x}(t) + Kx(t) = B_u u(t), \tag{5}$$

for $t > 0$, and initial value $x(0) = x_0$. For the assembly, the parameters from Tab. 1 have been used.

Table 1: Material parameters.

Part	Parameter	Value	Unit
Hull	ρ_1	$1.07 \cdot 10^{-3}$	$\mathrm{kg\,m^{-3}}$
	E_1	$0.025 \cdot 10^{6}$	$\mathrm{kg\,m^{-1}\,s^{-2}}$
	ν_1	0.48	
Beam	ρ_2	$1.4 \cdot 10^{3}$	$\mathrm{kg\,m^{-3}}$
	E_2	$2.96 \cdot 10^{10}$	$\mathrm{kg\,m^{-1}\,s^{-2}}$
	ν_2	0.3	
Rayleigh damping	α_{r}	0.0001	$\mathrm{s^{-1}}$
	β_{r}	0.0002	s

Achieving a fish-like movement of the fishtail requires the control over the end of the carbon center beam (the fin). Thus a virtual output, called point of interest (POI), is introduced

$$y_{\mathrm{poi}}(t) = C_{\mathrm{poi}} x(t).$$

The location of the POI (0 m, 0 m, 0.21 m) is equal to a single mesh point, such that the output vector reduces to

$$C_{\mathrm{poi}} = \begin{bmatrix} \dots & 0 & 1 & 0 & 0 & 0 & \dots \\ \dots & 0 & 0 & 1 & 0 & 0 & \dots \\ \dots & 0 & 0 & 0 & 1 & 0 & \dots \end{bmatrix},$$

where the rows represent the displacement in the different directions (z_1, z_2 and z_3). In order to achieve a good reconstruction of the full state by prolongation of the reduced system state, a group of virtual outputs lying on the boundary surface with the output equation

$$y(t) = C_p x(t) \tag{6}$$

has been defined, whereby the POI is included in this equation. The location of the used mesh-points are given in Tab. 2.

Table 2: Virtual output locations for the model order reduction.

z_1	z_2	z_3
0.05	0.0	0.0
0.0474526	0.0	0.0599584
0.04032111	0.0	0.105274
0.0326229	0.0	0.136726
0.0250675	0.0	0.16107
0.0168069	0.0	0.183588
0.0	0.0	0.21

3 Model order reduction of second-order systems

As a result from (5) and (6) in the modeling above, we investigate second-order systems of the form

$$\begin{aligned} M\ddot{x}(t) + E\dot{x}(t) + Kx(t) &= B_u u(t), \\ y(t) &= C_p x(t), \end{aligned} \tag{7}$$

with $M, E, K \in \mathbb{R}^{n\times n}$ symmetric and positive definite, $B_u \in \mathbb{R}^{n\times m}$ and $C_p \in \mathbb{R}^{p\times n}$, with $m = 1$ and $p = 21$. Using the Laplace transform, their transfer functions are

$$H(s) = C_p(s^2 M + sE + K)^{-1} B_u, \tag{8}$$

for $s \in \mathbb{C}$. With one exception, the methods below all fall into the class of second-order projection methods. The reduced-order models (ROMs) are in those cases represented as

$$\begin{aligned} \hat{M}\ddot{\hat{x}}(t) + \hat{E}\dot{\hat{x}}(t) + \hat{K}\hat{x}(t) &= \hat{B}_u u(t), \\ \hat{y}(t) &= \hat{C}_p \hat{x}(t), \end{aligned}$$

with

$$\begin{aligned} \hat{M} &= W^\top M V, \qquad \hat{E} = W^\top E V, \\ \hat{K} &= W^\top K V, \\ \hat{B}_u &= W^\top B_u, \qquad \hat{C}_p = C_p V, \end{aligned} \tag{9}$$

and $V, W \in \mathbb{R}^{n \times r}$.

The companion form first-order formulation

$$\begin{aligned} \underbrace{\begin{bmatrix} J & 0 \\ 0 & M \end{bmatrix}}_{\mathcal{E}} \dot{z}(t) &= \underbrace{\begin{bmatrix} 0 & J \\ -K & -E \end{bmatrix}}_{\mathcal{A}} z(t) + \underbrace{\begin{bmatrix} 0 \\ B_u \end{bmatrix}}_{\mathcal{B}} u(t), \\ y(t) &= \underbrace{\begin{bmatrix} C_p & 0 \end{bmatrix}}_{\mathcal{C}} z(t), \end{aligned} \tag{10}$$

with an arbitrary invertible matrix $J \in \mathbb{R}^{n \times n}$, is the basis of the derivation of V and W in many methods. Also the exceptional method mentioned above can be interpreted as a structured projection method applied to this formulation of the system. While J is generally arbitrary, the literature mostly uses $J = I_n$ the identity matrix, whereas in our implementations we often use $J = -K$. Then $\mathcal{A}$ is symmetric, which can be exploited to reduce the computational effort in many places.

3.1 Modal approximation

The *modal truncation* method is one of the oldest model reduction approaches relying on a similarity transformation of the system into the so-called modal representation and a truncation of all undesired modes of the system [51]. Therefore, a projection basis for the reduced-order model only consists of the eigenvectors corresponding to the desired eigenvalues. In case of second-order systems like (7), the corresponding quadratic eigenvalue problem has to be considered

$$\left(\lambda^2 M + \lambda E + K\right) x = 0, \tag{11}$$

for $\lambda \in \mathbb{C}$ the eigenvalues and $0 \neq x \in \mathbb{C}^n$ the (right) eigenvectors. Since the system matrices in (11) are symmetric, the left and right eigenvectors are the same, i. e., it is sufficient to consider only the right eigenvectors for the projection basis.

As in [12], the computation of the eigenvalues λ can be simplified by considering the Rayleigh damping (4). Therefore, the quadratic eigenvalue problem (11) then simplifies to solving the generalized eigenvalue problem of the form

$$\omega^2 M x = K x, \tag{12}$$

where the original quadratic eigenvalues λ can be reconstructed using $\omega \in \mathbb{R}_{>0}$. For the reduction of the model, the smallest r eigenvalues of (12) are computed, which correspond to the eigenvalues of (11) with small absolute value. The corresponding eigenvectors $V_{\mathrm{mt}} = [x_1, \ldots, x_r]$ from (12) are then used for the reduction of the system as in (9) by setting $W = V = V_{\mathrm{mt}}$.

For the modal truncation approach, the choice of the eigenvalues that shall remain in the reduced-order model is critical. The drawback of taking only the smallest ones into account is the neglect of input and output matrices, which have a significant influence on the input-output behavior of the system. The extension of the modal truncation method to a more intelligent choice of the eigenvalues is the *dominant pole algorithm* [52]. Here, the eigenvalues with the strongest influence on the system behavior are computed and then chosen for the reduced-order model. Adaptations of the dominant pole algorithm to the case of second-order systems have been done in [53] for single-input single-output systems and in [54] for multi-input multi-output systems.

In this paper, we suggest a variation of the second-order dominant pole algorithm for modally-damped second-order systems. In general, the system (7) is called modally damped if M, E and K are symmetric positive definite and $EM^{-1}K = KM^{-1}E$ holds. Note that second-order systems with Rayleigh damping automatically belong to this class of mechanical systems. As shown in [55], choosing the matrix of eigenvectors $X \in \mathbb{R}^{n \times n}$ from

$$MX\Omega^2 = KX, \tag{13}$$

where $\Omega^2 = \operatorname{diag}(\omega_1^2, \ldots, \omega_n^2)$ and $X = [x_1, \ldots, x_n]$, such that

$$X^\top M X = \Omega^{-1} \quad \text{and} \quad X^\top K X = \Omega, \tag{14}$$

allows to additionally diagonalize the damping term

$$X^\top E X = 2\Xi, \tag{15}$$

with $\Xi = \operatorname{diag}(\xi_1, \ldots, \xi_n)$. Using (13)–(15), the transfer function (8) can be rewritten as

$$\begin{aligned} H(s) &= C_p X (s^2 \Omega^{-1} + 2s\Xi + \Omega)^{-1} X^\top B_u \\ &= \sum_{k=1}^{n} \frac{\omega_k C_p x_k x_k^\top B_u}{(s - \lambda_k^+)(s - \lambda_k^-)}, \end{aligned} \tag{16}$$

where the eigenvalues λ_k^+ and λ_k^- are given as

$$\lambda_k^\pm = -\omega_k \xi_k \pm \omega_k \sqrt{\xi_k^2 - 1}. \tag{17}$$

Note that from (17) it follows that the smallest eigenvalues of (12) correspond to the eigenvalues of (11) closest to the imaginary axis. The pole residue formulation of the trans-

Algorithm 1: Second-Order Modally-Damped Dominant Pole Algorithm (SOMDDPA) [56]

```
Input: System matrices $M = M^\top > 0$, $E = E^\top > 0$, $K = K^\top > 0$ with $EM^{-1}K = KM^{-1}E$, $B_u$, $C_p$, initial value $s_1$,
       tolerance $0 < \tau \ll 1$, number of wanted eigenvalues $k_{\text{wanted}}$.
Output: Eigenvector matrix X and dominant eigenvalue pairs $\lambda^\pm = [\lambda_1^\pm, \dots, \lambda_k^\pm]$.
Initialize $V = X = [\,]$, $\lambda^\pm = [\,]$, $k = 0$, $j = 1$.
Compute the left and right singular vectors $y_0$ and $u_0$ of $\sigma_{\max}(H(s_1))$.
while $k < k_{\text{wanted}}$ do
    Solve the linear systems of equations
        $\left(s_j^2 M + s_j E + K\right) v_j = B_u u_k$ and $(\bar{s}_j^2 M + \bar{s}_j E + K) w_j = C_p^\top y_k$.
    Expand the projection subspace $V = \texttt{orth}([V, \texttt{real}(v_j), \texttt{imag}(v_j), \texttt{real}(w_j), \texttt{imag}(w_j)])$.
    Compute the most dominant eigentriple $(\theta_j^+, \theta_j^-, \hat{x}_j)$ via (13)–(18) of
        $V^\top M V$, $V^\top E V$, $V^\top K V$, $V^\top B_u$, $C_p V$.
    Compute the eigenvector and the residual
        $x_j = V\hat{x}_j$ and $r_j = \left(\left(\theta_j^+\right)^2 M + \theta_j^+ E + K\right) x_j$.
    if $\|r_j\| < \tau$ then
        Set $k = k + 1$ and $X = [X, x_j]$, $\lambda^\pm = [\lambda^\pm, \theta_j^\pm]$.
        Deflate newly found eigentriples.
        Update right and left singular vectors $y_k$ and $u_k$ of $\sigma_{\max}(H(\theta_j^+))$.
    end
    Set $s_{j+1} = \theta_j^+$ and $j = j + 1$.
    Restart if necessary.
end
```

fer function (16) indicates that for a dominant pole algorithm, the pairs of eigenvalues have to be considered. The direct extension of a dominant pole, as given in [54], reads as follows: The pole pair $(\lambda_k^+, \lambda_k^-)$ is called dominant if it satisfies

$$\frac{\|\omega_k C_p x_k x_k^\top B_u\|_2}{Re(\lambda_k^+) Re(\lambda_k^-)} > \frac{\|\omega_j C_p x_j x_j^\top B_u\|_2}{Re(\lambda_j^+) Re(\lambda_j^-)}, \tag{18}$$

for all $j \neq k$. The corresponding dominant pole algorithm then computes the r most dominant pole pairs and the corresponding eigenvectors, such that the reduced-order model is given as

$$H_r(s) = \sum_{k=1}^{r} \frac{\omega_k C_p x_k x_k^\top B_u}{(s - \lambda_k^+)(s - \lambda_k^-)} \approx H(s),$$

where $V_{\text{dpa}} = [x_1, \dots, x_r]$ gives a corresponding projection basis. The resulting method is described in Algorithm 1.

A big advantage of Algorithm 1, compared to the methods in [53] and [54], is the restriction to one-sided projections. This preserves the system and eigenvalue structure in each single step such that the resulting eigenvector basis will be real and no additional unrelated Ritz values are introduced in the reduced-order model, which usually disturb the resulting approximation.

For the whole Algorithm 1, we decided to restrict to the “plus” part of the eigenvalue pairs, since most of the steps produce the same results independent of the chosen part of the eigenvalue pairs. However, in some cases it might be preferred to switch between the parts, e. g., if eigenvalues are accumulated on one side but not on the other. The definition of dominance (18) can easily be exchanged by, e. g., other convenient measures, or used to target certain types of eigenvalues like only real or imaginary ones.

3.2 Moment matching

The *moment matching* (see, e. g., [18, 19, 20]) approach aims to interpolate the transfer function (8) and its derivatives at chosen points in the frequency domain. For second-order systems (7), a system interpolating the transfer function in the point $\theta \in \mathbb{C}$ can be constructed (e. g., [57, 58, 59]) by projection (9) with satisfying either

$$(\theta^2 M + \theta E + K)^{-1} B_u \in \operatorname{range}(V) \text{ or}$$
$$(\theta^2 M + \theta E + K)^{-\mathsf{H}} C_p^\top \in \operatorname{range}(W).$$

The *Padé approximation* can be seen as a special case of moment matching, where $\theta = 0$ is the only expansion point. The projection basis is then build as $V_\ell = [B_u, K^{-1}B_u, \dots, K^{-(\ell-1)}B_u]$.

As observed in Section 4.3.4 and Section 4.4, the plain moment matching approach is sufficient for our model. In the general case, an adaptive method, such as the one discussed in [60], may improve the approximation.

3.3 Balanced truncation

3.3.1 First-order systems

Originally invented for standard state-space systems, *balanced truncation* for first-order ordinary differential equation systems

$$\check{E}\dot{z}(t) = Az(t) + Bu(t), \qquad y = Cz(t), \tag{19}$$

i. e., for invertible $\check{E}$, exploits the energy interpretations of the two system Gramians P_B, $\check{E}^\top P_C \check{E}$ to identify those states with little contribution in the input to output energy transfer in the balanced realization of the system, i. e., $P_B = S = \check{E}^\top P_C \check{E}$ diagonal, via the magnitude of the diagonal entries in S; see [61].

The matrices P_B, P_C are computed solving the two adjoint Lyapunov equations

$$A P_B \check{E}^\top + \check{E} P_B A^\top + BB^\top = 0,$$
$$A^\top P_C \check{E} + \check{E}^\top P_C A + C^\top C = 0.$$

Their lower triangular Cholesky factors L (of P_C), and R (of P_B) are used in the square root method to simultaneously balance and truncate the system for optimal performance. To this end, one first computes the singular value decomposition

$$\begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} S_1 & 0 \\ 0 & S_2 \end{bmatrix} \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}^\top = L^\top \check{E} R,$$

with $S_1 \in \mathbb{R}^{r \times r}$ containing the larges r Hankel singular values of the system and U_1, Q_1 containing the corresponding (dominant) singular vectors. Then the left (T_L) and right (T_R) truncating transformation matrices, corresponding to W and V in the ROM computation analogous to (9), are computed as

$$T_L = L U_1 S_1^{-\frac{1}{2}}, \quad T_R = R Q_1 S_1^{-\frac{1}{2}}.$$

Note that this guarantees $T_L^\top \check{E} T_R = I_r$. Further in case of large and sparse systems, the Cholesky factors L and R are replaced by tall and skinny low-rank factors and the SVD by the economy size SVD. Moreover, asymptotic stability of the original system guarantees existence and uniqueness of the two Gramians, minimality of the realization guarantees their positive definiteness, i. e., the existence of Cholesky factors L and R, and the $\mathcal{H}_\infty$-norm error of the system approximation is bounded by twice the sum of the diagonal entries in S_2, i. e., the truncated Hankel singular values.

3.3.2 Second-order systems

In case of second-order systems, the Gramians of (10) are subdivided in the same 2 × 2 block fashion and diagonal blocks corresponding to the velocity and position portions of the phase space state representation are defined as the velocity and position Gramians of (7) as described in, e. g., [16, 17]. The method in [16] now uses these to form $2n \times 2r$ block-diagonal transformation matrices for projection of (10). To this end, the square root method is run twice, with only position Gramians and only velocity Gramians to determine the corresponding diagonal blocks in the transformation matrices. Subsequently, it establishes a second-order transfer function to identify the reduced second-order matrices. Thus in contrast to (9), here the reduced-order matrices are computed according to

$$\begin{aligned} \hat{M} &= T\left(W_v^\top M V_v\right) T^{-1}, & \hat{E} &= T\left(W_v^\top E V_v\right) T^{-1}, \\ \hat{K} &= T\left(W_v^\top K V_p\right), & & \\ \hat{B}_u &= T\left(W_v^\top B_u\right), & \hat{C}_p &= C_p V_p, \end{aligned} \tag{20}$$

where $T = W_p^\top J V_v$, and W_p, W_v, V_p, V_v are the position and velocity diagonal blocks in the first-order transformation matrices. This is actually the first-order projection method in the discussion above. We refer to it as second-order (so) balanced truncation following [16, 17].

On the other hand, in [17] projection methods are presented, according to (9), running the square root method on combinations of velocity and position second-order Gramians to determine W and V. This possibly requires two singular value decompositions (SVDs). The simplicity of this approach has motivated efficient computation

Table 3: Second-order balancing type methods.

Type	SVDs	Transformations	ROM
v	$USQ^\top = L_v^\top MR_v$	$W = L_vU_1S_1^{-\frac{1}{2}}$ $V = R_vQ_1S_1^{-\frac{1}{2}}$	(9)
fv	$USQ^\top = L_p^\top JR_p$	$W = L_pU_1S_1^{-\frac{1}{2}}$ $V = R_pQ_1S_1^{-\frac{1}{2}}$	(9)
vp	$USQ^\top = L_p^\top JR_v$	$W = L_pU_1S_1^{-\frac{1}{2}}$ $V = R_vQ_1S_1^{-\frac{1}{2}}$	(9)
pv	$USQ^\top = L_v^\top MR_p$	$W = L_vU_1S_1^{-\frac{1}{2}}$ $V = R_pQ_1S_1^{-\frac{1}{2}}$	(9)
vpv	$*SQ^\top = L_p^\top JR_v$ $U ** = L_v^\top MR_p$	$W = L_vU_1S_1^{-\frac{1}{2}}$ $V = R_vQ_1S_1^{-\frac{1}{2}}$	(9)
p	$*SQ^\top = L_p^\top JR_p$ $U ** = L_v^\top MR_v$	$W = L_vU_1S_1^{-\frac{1}{2}}$ $V = R_pQ_1S_1^{-\frac{1}{2}}$	(9)
so	$U_pS_pQ_p^\top = L_p^\top JR_p$ $U_vS_vQ_v = L_v^\top MR_v$	$W_p = L_pU_{p,1}S_{p,1}^{-\frac{1}{2}}$ $V_p = R_pQ_{p,1}S_{p,1}^{-\frac{1}{2}}$ $W_v = L_vU_{v,1}S_{v,1}^{-\frac{1}{2}}$ $V_v = R_vQ_{v,1}S_{v,1}^{-\frac{1}{2}}$	(20)

strategies [62, 63, 64] available in the M-M.E.S.S. [31] and MORLAB [32] toolboxes. We call these methods velocity (v) or free velocity (fv) balancing [14], when only velocity, or only position Gramians are used. Further, we call them position-velocity (pv) or velocity-position (vp) balancing, with the first one referring to the part of the controllability Gramian. These methods all need only a single SVD. Additional variants, using two SVDs are the vpv-balancing, that uses the same right transformation V as in pv-balancing, but uses only velocity information for the left transformation W. While position (p) balancing uses the same information for the left transformation, but both position Gramians for the right transformation. In both cases, the SVD for the right transformation determines the singular values for the truncation.

The transformation formulas and required computations for all seven types have been collected in Tab. 3.

3.4 Frequency-limited balanced truncation

As described in the previous section, the usual balanced truncation approach aims for a global approximation of the system in the frequency- and time-domain. But for the practical application at hand, this is not necessary. Here, it is intended to apply the model only in a certain range of frequencies, namely between 0 and 10 Hz. As first discussed in [65] for this purpose, the *frequency-limited balanced truncation* method can be applied. The difference to the classical balanced truncation method lies in the restriction of the system Gramians to the chosen frequency range $\Omega = [\omega_1, \omega_2] \cup [-\omega_2, -\omega_1]$. Considering the case of first-order systems (19), the limited controllability and observability Gramians P_B^{FL}, $\check{E}^T P_C^{FL} \check{E}$ are given via the solutions of the following two Lyapunov equations

$$A P_B^{FL} \check{E}^\top + \check{E} P_B^{FL} A^\top + B_\Omega B^\top + B B_\Omega^\top = 0,$$
$$A^\top P_C^{FL} \check{E} + \check{E}^\top P_C^{FL} A + C_\Omega^\top C + C^\top C_\Omega = 0,$$

with $B_\Omega = \check{E} F_\Omega B$, $C_\Omega = C F_\Omega \check{E}$ and

$$F_\Omega = \operatorname{Re}\left(\frac{i}{\pi}\ln((A + i\omega_1\check{E})^{-1}(A + i\omega_2\check{E}))\right)\check{E}^{-1},$$

where $\ln(M)$ denotes the principle branch of the matrix logarithm of M.

Even though originally constructed for the first-order case, the method can be extended to second-order systems (7) by considering the general method in Section 3.3. A discussion of this extension for some versions of the second-order balanced truncation method can be found in [66], but has only been applied to small systems so far. The application to the large-scale system case can be done using projection methods [23].

A different concept of the frequency-limited balanced truncation for the large-scale system case is the application of a two-step approach [21, 22]. Therefor, the system is pre-reduced by a moment matching method choosing sample points from the frequency range $\Omega = [\omega_1, \omega_2] \cup [-\omega_2, -\omega_1]$. Afterwards, the frequency-limited balanced truncation is used on the pre-reduced system. An implementation of the frequency-limited balanced truncation suited for such dense medium-scale unstructured systems can be found in the MORLAB toolbox [32].

4 Numerical experiments

In this section, we compare the methods from Section 3 for the model described in Section 2. The numerical realization of the fishtail, we employ in our tests, has a second-order state dimension of $n = 779\,232$. The model is prepressurized, i. e., it features one input ($m = 1$), representing the mass flow between the left and right pressure chambers. Further, we have picked $p = 21$ outputs, representing the displacement into the three spatial directions of the rear tip of the central carbon beam, as well as the six further reference points on the flank of the fishtail, given in Tab. 2. As described in Section 2, the rear tip is the actual point of interest, and its motion is dominated by the displacement in the z_2-direction (compare Fig. 3.). The additional points have been added to improve the state reconstruction in the coordinates of the original model. Earlier experiments with only the tip-displacement as the output lead to rather bad reconstructions, with e. g., the balanced truncation approaches, since they are designed to well approximate the input output behavior, but make no statements about the interior states.

The generation of the full-order model and all time-domain computations have been executed on the compute server described in Tab. 4a at CAU Kiel.

All computations for the generation of the reduced-order models and frequency domain evaluations have been executed on a single node of the high performance compute cluster `mechthild` at the Max Planck Institute in Magdeburg. The hardware and software features can be found in Tab. 4b.

The computation of the Gramian factors, required for the standard balanced truncation variants, uses M-M.E.S.S. and implicitly employs (10) with $J = -K$. On the other hand, the hybrid approach uses MORLAB and employs $J = I_n$. For best comparability of the reduction results, the transfer function sampling and error computation is performed with the second-order square root method from MORLAB for Gramians computed with $J = I_n$ and an equivalent modification reflecting $J = -K$ otherwise.

Table 4: Hardware and Software environments for the experiments.

CPU	2× Intel®Xeon® E5-4110 @ 2.60 GHz (3.4 GHz Turbo)
Cores	2×8
RAM	64 GB DDR4 with ECC
OS	CentOS Linux release 7.5.1804
Platform type	x86_64 (64 Bit)
OpenBLAS	0.3.3 [67]
FIREDRAKE	[68]
Python	3.7.0 [69]
NumPy	1.15.4 [70]
SciPy	1.1.0 [71]

(a) FOM generation and time-domain computations.

CPU	2× Intel®Xeon® Silver 4110 @ 2.10 GHz (3.0 GHz Turbo)
Cores	2×8
RAM	192 GB DDR4 with ECC
OS	CentOS Linux release 7.5.1804
Platform type	x86_64 (64 Bit)
MATLAB	R2108a [72]
MORLAB	4.0 [32]
M-M.E.S.S.	1.01 [31] (with patches from supplemental material)

(b) ROM generation and frequency domain computations.

Prior to any of the computations, we symmetrized the model data to get rid of numerical non-symmetry in the model matrices M, K, and consequently E, exported from FIREDRAKE. This does not only bring the discretized model closer to the theoretical properties, but also practically enables MATLAB to exploit symmetry of the matrices wherever possible.

4.1 Data and code availability

The data matrices defining the model (5), (6) are available as a data package authored by D. Siebelts, A. Kater, and T. Meurer [73]. They are the basis for all experiments reported in the following. The authors of this paper provide the MATLAB and Python codes for execution of the experiments reported below in [74]. Since especially the computation of the Gramian factors for the balancing-based approaches is time consuming and comes with high memory demands, we provide all results and intermediate results as an additional data set [75].

The codes for the experiments in Section 4.3.2 and the dominant pole approaches in Section 4.3.4 are available as separate packages in [24] and [56] authored by P. Benner and S. W. R. Werner.

In case the model should be generated from a different finite element tool, the mesh data and the measurement positions for the problem description can be found in [76].

4.2 Structure of the experiments

In the following, we report first on the evaluation of the reduction results in the frequency domain. Here, we group the methods investigated by the general approach first. These groups represent the subsections of Section 3. We have eight different methods for plain balanced truncation, the same amount of variants replacing the full Gramians by frequency-limited Gramians, as well as frequency-limited Gramians based on a moment matching pre-reduction of the system. Moreover, we investigate different versions of modal approximation, including the approach used in [12] as a reference and introducing second-order dominant pole based reduction. Some versions of Krylov-reduction, including Padé approximation and moment matching for logarithmically distributed expansion points complete the picture. In total we compare 57 ROMs.

In order to keep the presentation limited, among those we identify a number of significant candidates for further comparison in Section 4.3.5. These are then also used in the time domain comparison in Section 4.4.

In all Figures below, the numbers in braces in the legends depict the reduced order of the corresponding model, i. e., its state-space dimension r.

4.3 Frequency domain

In some preliminary tests we identified the frequency range from 10^{-2} to 10^{4} Hz as the most reasonable for further investigation. Considering the model at hand, the most interesting region is up to 10 Hz since one will, likely, not be able to realize higher frequencies in the propulsion motion.

In all our experiments, we sample this range with 400 equally distributed points on the logarithmic scale. The markers in the plots do not directly correspond with the computed data points, but represent only every 80st sample point with a shift of 8 points between subsequent graphs.

Moreover, the dashed vertical lines mark the end of the frequency range of interest. That means, especially for the frequency-limited methods, the error right of there is rather irrelevant for the operation of the system.

4.3.1 Balanced truncation variants

The computation of the low-rank first-order Gramian factors for the companion form (10) of dimension 1 558 464 by 300 (controllability) and 1 890 (observability) took about 41 hours, due to the size and bad conditioning of the system matrices. Here, the controllability factor, adding only a single column to the factor in each iteration step, could not reach the prescribed residual tolerance (10^{-9}) and stopped after the maximum iteration number. On the other hand, the observability factor, adding 21 new columns in every step of the iteration, converged after 90 steps with a final normalized residual of $7.594584 \cdot 10^{-10}$.

In Fig. 4 and Fig. 5 we compare heuristically truncated ROMs and fixed-order ROMs resulting from the square root method applied to these Gramian factors and their appropriate subblocks for the different methods summarized in Tab. 3. The heuristic truncation here drops all singular values (and corresponding singular vectors) starting from the point, where the sum of the following computed singular values is smaller than the prescribed tolerance (10^{-4}) times the largest singular value.

While in Fig. 4 velocity-position (VP) balancing appears to be clearly more accurate than the other methods, one should also observe, that it is of dimension 20, while the second best (free-velocity balancing (FV) and position balancing (P)) achieve comparable accuracy with a factor of four smaller models. This is also clearly reflected in the fixed ROM-order plots in Fig. 5, where VP balancing is the second worst and VPV balancing is completely off.

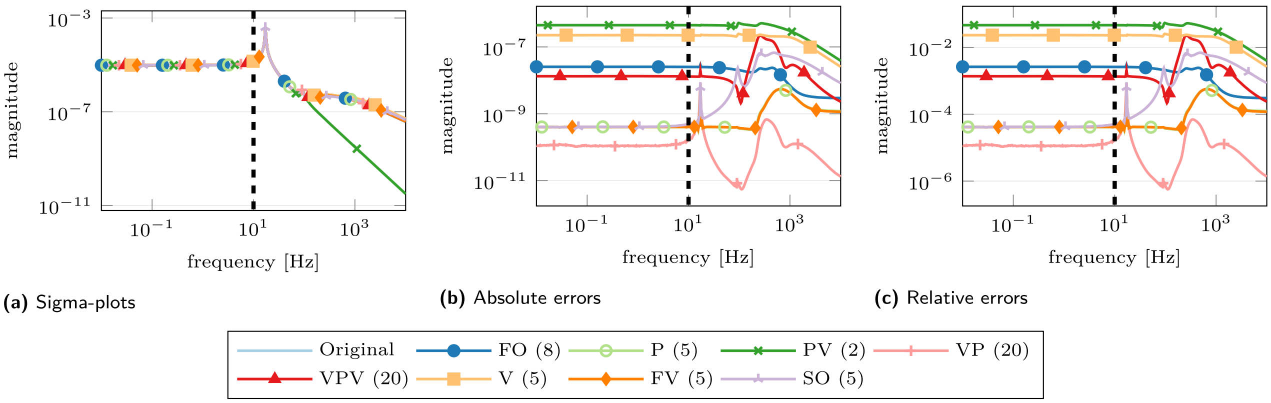
The clear winners here are the free-velocity balancing and position balancing methods together with second-order balancing (SO), that has slight difficulties for the higher frequencies, beyond the actual range of interest. In general, however, except for VPV balancing, all methods do an acceptable job for order 8, which is already smaller than the order 10 of the modal approximation generated in [12].

4.3.2 Frequency-limited balanced truncation variants

As expected, the frequency-limited Gramian factors are slightly thinner than the ones from the previous section. Their column dimensions are 1 407 for the observability and 113 for the controllability. On the one hand, the heuristically truncated models show comparable dimensions, with the small ones getting slightly larger and the larger ones a bit smaller than above. On the other hand, the variance of the results is far smaller but at the same time the overall approximation quality is considerably worse than above. Our interpretation of this observation is that the rather bad conditioning of the matrices, in the elasticity model used here, is affecting the approximation quality of the Krylov subspace based computation of the frequency-limited Gramian factors a lot more than the shifted system solves, where the shifts may very well improve the conditioning, in the computation method for the Gramians above. It is, however, also clearly visible that all methods miss the original model for frequencies above 100 Hz, which should be expected, since we limited the frequency range of interest to 0–10 Hz.

4.3.3 Hybrid frequency-limited balanced truncation variants

In this test, we compute input and output rational (block) Krylov subspaces for shifts between 10^{-4} and 10 Hz. Leading to complex basis matrices of width 2 100 and 100. We then form an orthogonal projection with $W = V$ by concatenating the real and imaginary parts of the complex basis and using a QR decomposition with numerical rank de-

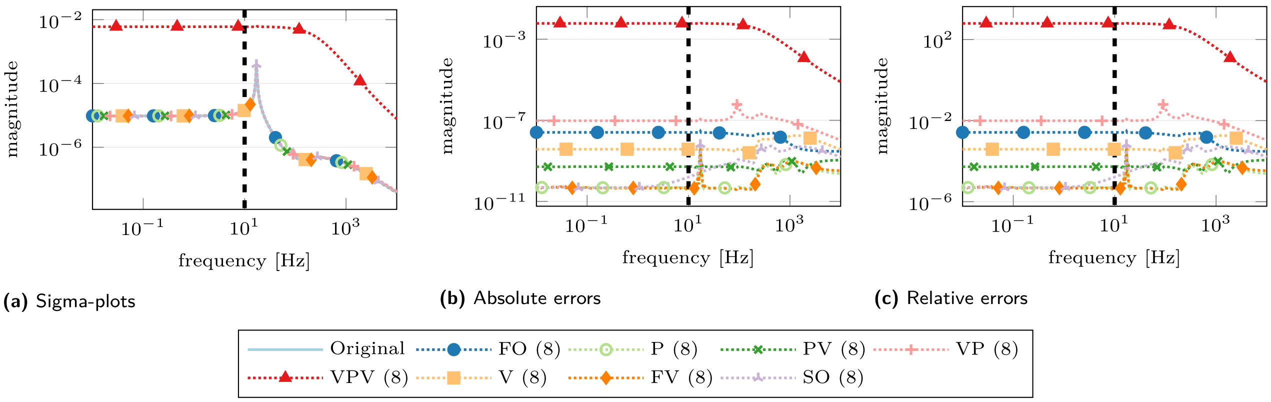


(a) Sigma-plots

(b) Absolute errors

(c) Relative errors

Figure 4: Reduction results for the various balanced truncation with relative sum truncation.

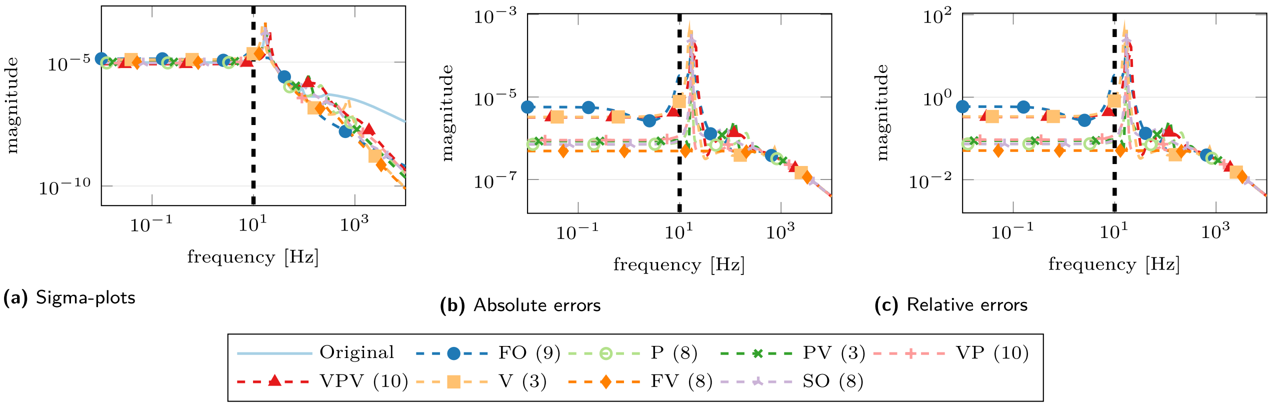


(a) Sigma-plots

(b) Absolute errors

(c) Relative errors

Figure 5: Reduction results for the various balanced truncation with fixed ROM dimension.



(a) Sigma-plots

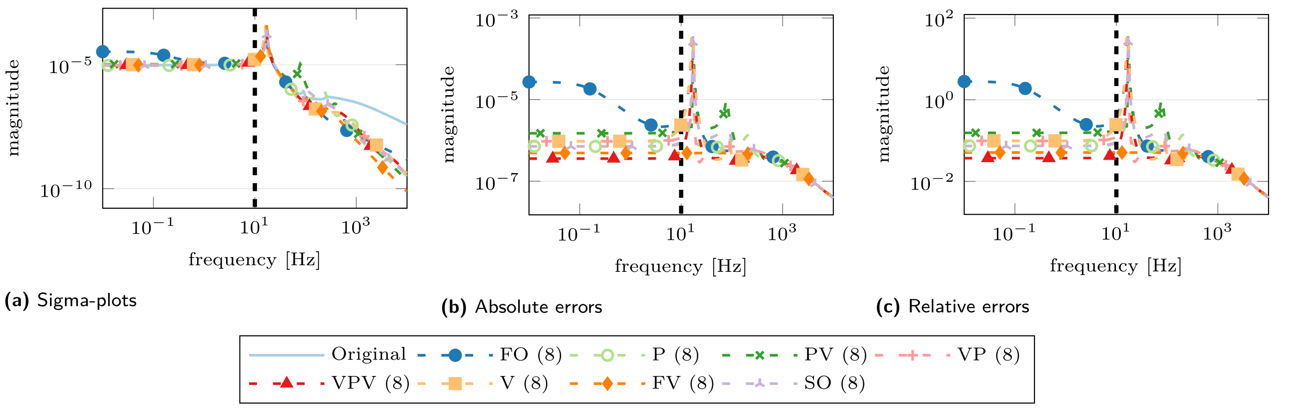
(b) Absolute errors

(c) Relative errors

Figure 6: Reduction results for the various frequency-limited balanced truncation with relative sum truncation.

cision afterwards. This way we generate a Krylov reduced model of dimension 60. We then compute the frequency-limited Gramian factors for its companion form using the sign-function iteration from MORLAB.

Interestingly, although the frequency range is limited in the same way as in Section 4.3.2 above, here (see Fig. 8 and Fig. 9) we observe much better approximations also for higher frequencies. Also, the reduced orders for the heuristic truncation are the smallest in the entire comparison. Moreover, in the frequency range of interest, the fixed-order ROMs of only order 5 can achieve comparable accuracy as the order 8 approximations for the un-

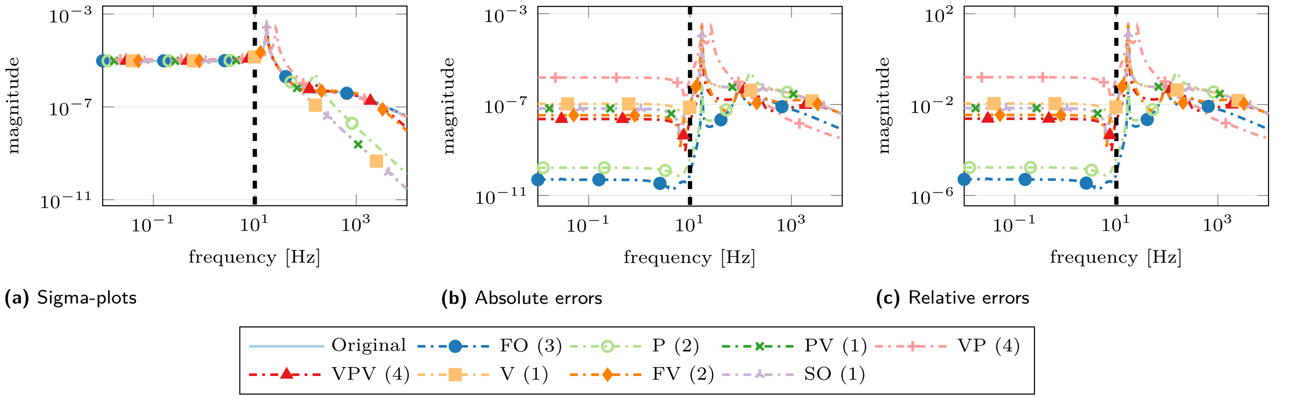


(a) Sigma-plots

(b) Absolute errors

(c) Relative errors

Figure 7: Reduction results for the various frequency-limited balanced truncation with fixed ROM dimension.

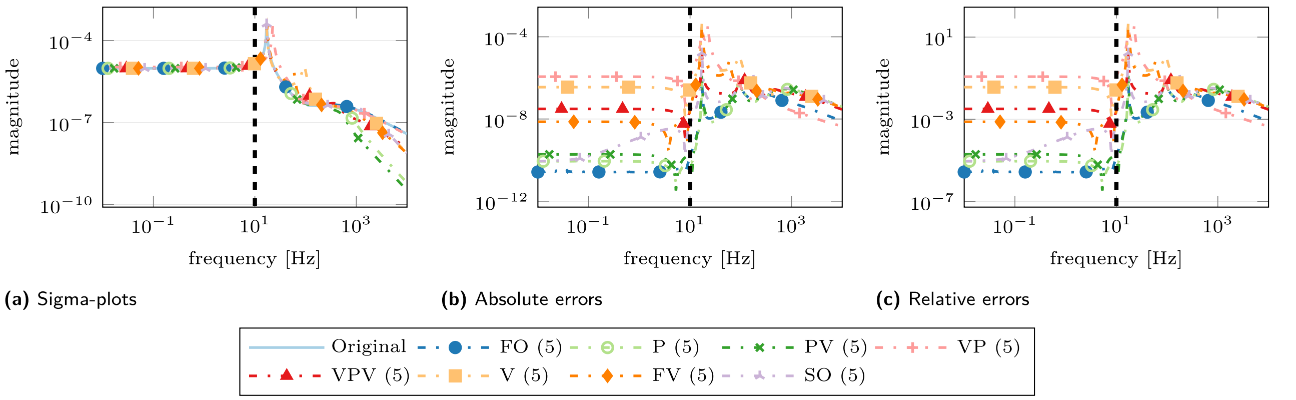


(a) Sigma-plots

(b) Absolute errors

(c) Relative errors

Figure 8: Reduction results for the various hybrid frequency-limited balanced truncation with relative sum truncation.



(a) Sigma-plots

(b) Absolute errors

(c) Relative errors

Figure 9: Reduction results for the various hybrid frequency-limited balanced truncation with fixed ROM dimension.

limited case in Section 4.3.1. Here, again position balancing is very good and only gets beaten by the first-order approximation, which however is not structure preserving, in the sense that we can not retrieve a second-order model from it. Looking at the fixed-order plots in Fig. 9 also position-velocity and second-order balancing are doing a good job, while the others are in the range of and above the finite element error we expect for the original model. They should thus be considered with more care.

4.3.4 Modal approximation and Krylov-based methods

The final group of methods consists of modal truncation type and Krylov subspace based methods. In Fig. 10, we compare directly to the modal truncation (MT) of order 10. The first observation is, that the MT modes are also dominant, such that the dominant pole based truncation methods DPA_LRP (using the dominance measure (18)) and DPA_LRPI (restricting to only conjugate complex poles) can not improve the results here. Also, we can see, that as expected Padé (PD), or higher order (2/4/10) expansion at zero, is good for very low frequencies, but then quickly gets worse. On the other hand, moment matching (MM), with 1/2/5 expansion points logarithmically distributed in the range between 10^{-4} and 10 Hz, is in this comparison giving the best results by far. Note that our expansion points are complex and we always add the real and imaginary parts to the basis in order to keep the transformation matrices real. This leads to generating systems with 2/4/10 vectors and reduced orders 2/3/7 due to linear dependence. Note further that the single expansion point version behaves (see especially Fig. 10b and Fig. 10c) very similar to the Padé ROMs and adding additional expansion points on the other end of the interval of interest is drastically improving the approximation. Thereby, the accumulated computation time for the Krylov subspace approaches is about 35 min.

4.3.5 Comparison

For the comparison, we take the basic modal truncation (MT (10)) as in [12] as the reference. From the last category, i. e., the results shown in Fig. 10, we choose the largest moment matching approximation (MM5 (7)). From the plain balanced truncation category with fixed order ($r = 8$) in Fig. 5 we add position balancing (P (8)). The frequency-limited methods are represented by velocity balancing (FV (8)) from Fig. 6, the position balancing method (P (2)) from the hybrid methods with dynamic truncation (Fig. 8), as well as the first-order (FO (5)) and position balancing (P (5)) from the set of hybrid fixed-order experiments (Fig. 9).

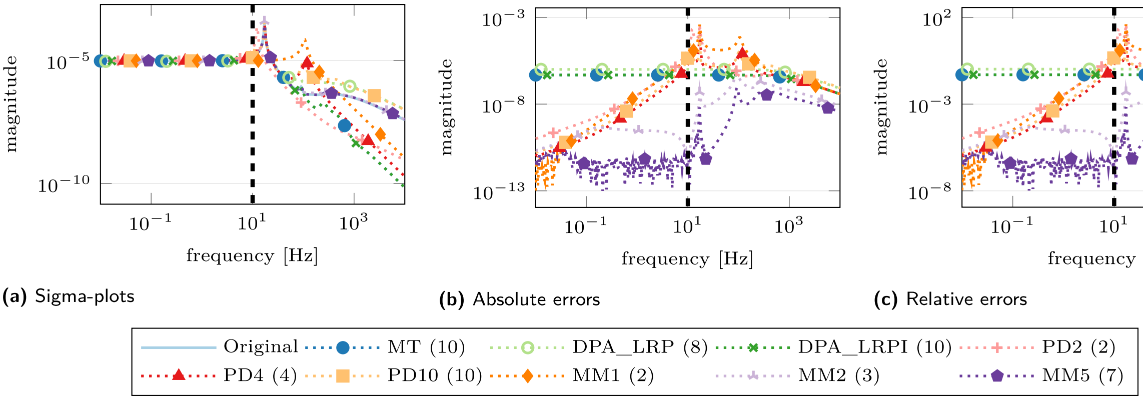
Fig. 11 clearly shows that from a frequency domain perspective, all selected methods present errors multiple orders of magnitude lower than modal truncation even though the reduced orders are also smaller in all cases.

Basically all results in Section 4.3.2 performed worse than the unconstrained frequency results from Section 4.3.1. We think that here the bad conditioning of the system matrices was less harmful to the alternating directions implicit (ADI) iteration for computing the standard Gramian factors than for the Krylov subspace method behind the computation of the frequency-limited Gramian factors. This issue needs closer investigation in the future. Interestingly, this picture looks different in the time domain experiments below.

4.4 Time domain

We repeat the comparison of the same ROMs in time domain. For the time integration of the second-order systems the SS22 algorithm from [38] is used, while the first-order time integration is created with the `lsoda` algorithm provided by `SciPy.integrate.odeint`. We compare two scenarios differing in the actuation of the system by the input $u(t)$. The first scenario is a typical step response for a 50 mbar pressure change

$$u_{\text{step}}(t) = 5000\delta(t - 0.1), \tag{21}$$

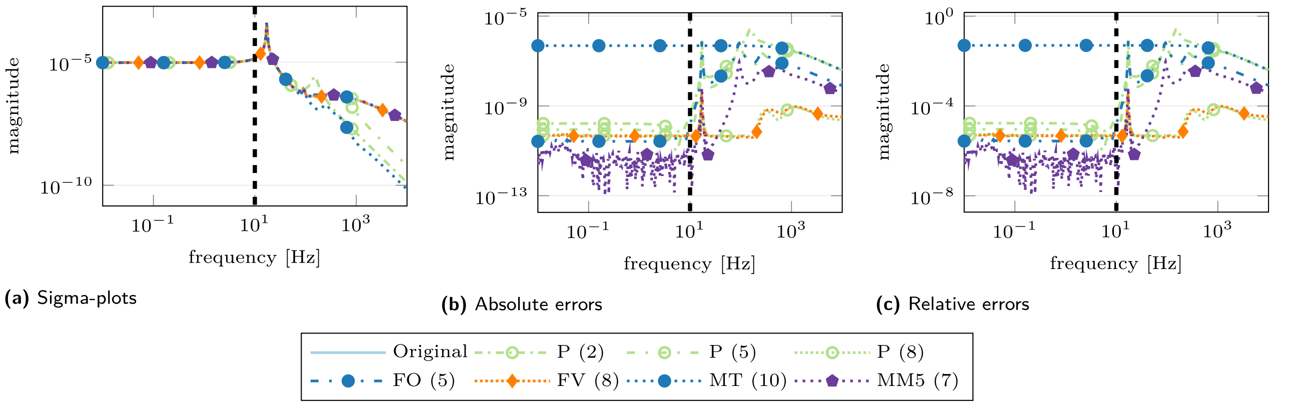


(a) Sigma-plots

(b) Absolute errors

(c) Relative errors

Figure 10: Reduction results for the modal approximation and Krylov-based reduction methods.

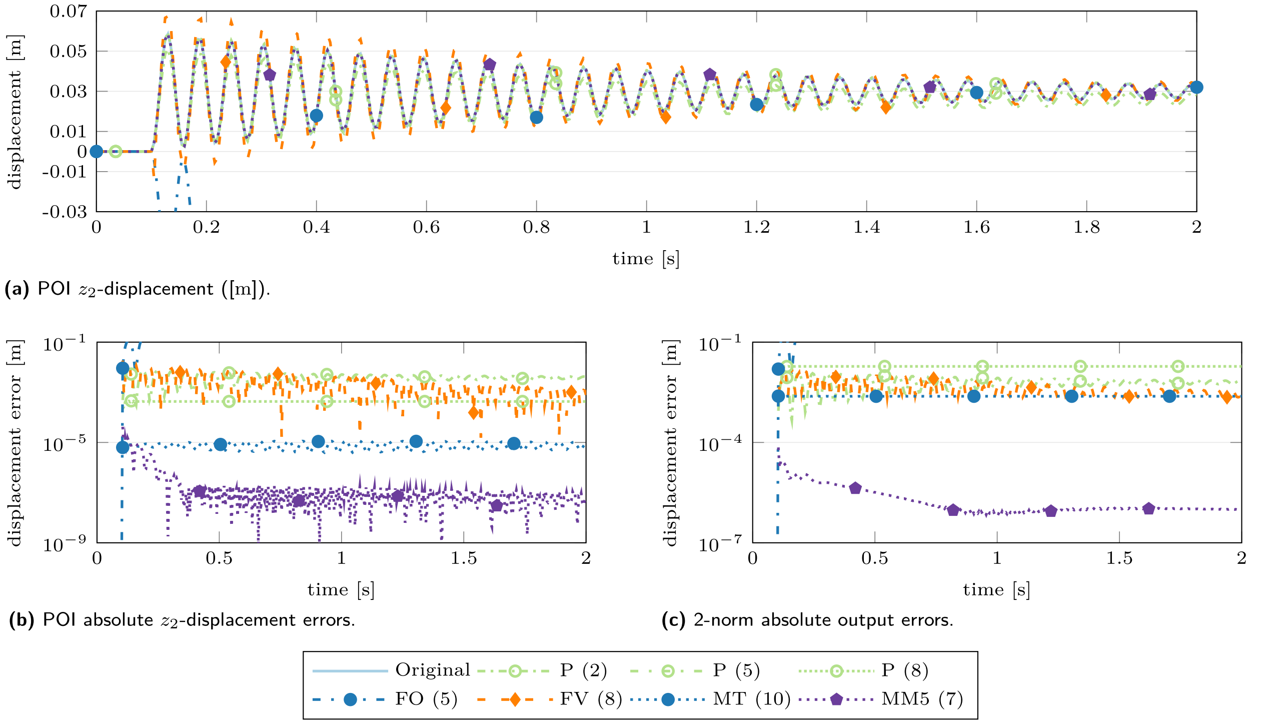


(a) Sigma-plots

(b) Absolute errors

(c) Relative errors

Figure 11: Comparison of reduction results for selected candidates from all reduction methods.



(a) POI z_2-displacement ([m]).

(b) POI absolute z_2-displacement errors.

(c) 2-norm absolute output errors.

Figure 12: Time-domain comparison of reduction results for selected candidates from all reduction methods using 5000 Pa step actuation.

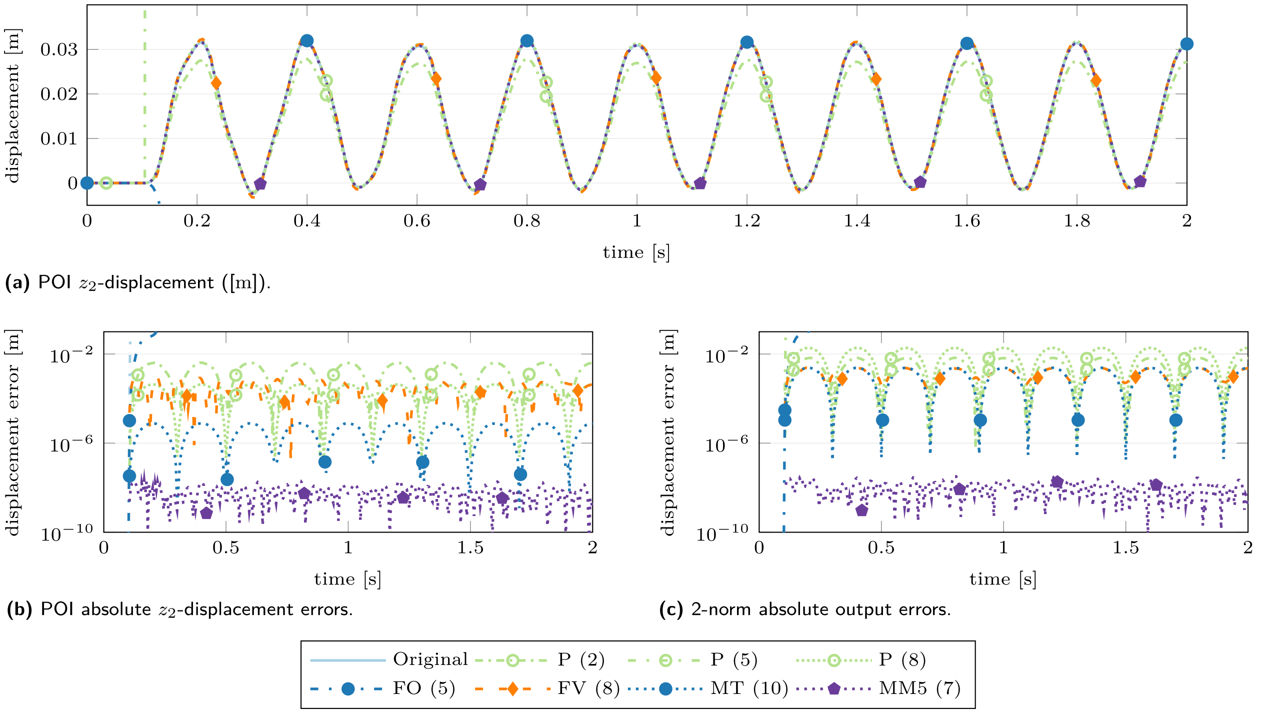
where $\delta(t)$ is the Heaviside function. In the second scenario, we use a $\mathcal{C}^{\infty}([0,2])$ input by the shifted sine

$$u_{\sin}(t) = 2500\,(\sin(10\pi(t-1.75))+1)\,\delta(t-0.1). \tag{22}$$

This removes the non-smooth shock actuation and, at the same time, brings us closer to an actual actuation scenario in fish locomotion.

An important issue, that we did not take into account in the frequency domain, is becoming obvious here. While classical balanced truncation has guarantees on the error and stability preservation, this is not true anymore for the frequency-limited and second-order methods. We have discussed the error bound above already. Here, we observe that the (hybrid) frequency-limited approaches do not preserve stability. Therefore, the P (5) and FO (5) curves are basically missing in Fig. 12 and Fig. 13. They are both leaving the amplitude ranges in all three sub-figures immediately after the control gets active at $t = 0.1$ s.

While it is very obvious that moment matching, with just a few expansion points in the frequency range of interest, is even more clearly the best option for this kind of system, the most impressive result is actually that of the hybrid frequency-limited position balancing (P (2)). With just



(a) POI z_2-displacement ($[\mathrm{m}]$).

(b) POI absolute z_2-displacement errors.

(c) 2-norm absolute output errors.

Figure 13: Time-domain comparison of reduction results for selected candidates from all reduction methods, via smooth sine actuation.

order 2, i. e., a compression factor of 389 616, it achieves a POI output error slightly worse than that of the other methods. Looking at the average output error it is even comparable with the other methods that are 4–5 times larger. We also observe that the frequency is nicely matched and just the amplitude can not be reproduced with only 2 degrees of freedom in both Fig. 12 and Fig. 13.

5 Conclusions

We have performed an in-depth investigation of different model order reduction methods for the artificial fishtail model from [12]. Observing that the transfer function does not feature significant resonances inside the frequency range of interest, it is little surprising that most of the methods perform relatively similar. A slightly unexpected effect was that although from the frequency-domain perspective most of the methods performed much better than the simple modal truncation, in time-domain the only method that performs significantly better than modal truncation is the plain moment matching with expansion points spread across the interval of interest. One feature of this particular model obviously is, that the smallest eigenvalues turn out to be the most dominant, as well. Therefore, the dominant pole-based approaches can not improve over plain modal truncation. For the other techniques it is noteworthy, that even rather global approaches such as balanced truncation, which usually tend to require larger reduced orders, provide a comparably good approximation with lower orders than modal truncation.

By far the biggest surprise is the very good result that can be achieved with the order 2 model from the hybrid frequency-limited approach. Although the final displacement error of a few mm in view of a desired displacement of about 3 cm may seem rather large, the compression factor of almost 400 000 is impressive. Moreover, this model still matches the frequency and phase of the swinging motion very well such that we expect it to perform well as the basis for an embedded controller.

Besides that, the moment matching approaches are the clear winner of the performance comparison. With computation times of less than twice that of the modal approximation, they achieve errors that are orders of magnitude lower, as soon as more than one expansion point is taken.

Overall, we think that our experiments state nicely that the additional effort for the more sophisticated reduction approaches can very well pay off. Due to the length of this manuscript in its current form, we have postponed additional comparisons in terms of, e. g., state reconstruction to follow-up publications.

Funding: The work of the third author was supported by the German Research Foundation (DFG) research training group 2297 “MathCoRe”, Magdeburg, and the German Research Foundation (DFG) priority program 1897: “Calm, Smooth and Smart – Novel Approaches for Influencing Vibrations by Means of Deliberately Introduced Dissipation”.

References

1. E. Kelasidi, P. Liljebäck, K. Y. Pettersen, J. Gravdahl, Innovation in underwater robots: Biologically inspired swimming snake robots, IEEE Robotics & Automation Magazine 23 (2016) 44–62. doi:10.1109/MRA.2015.2506121.
2. F. Fish, Advantages of natural propulsive systems, Marine Technology Society Journal 47 (2013) 37–44. doi:10.4031/MTSJ.47.5.2.
3. X. Tan, Autonomous robotic fish as mobile sensor platforms: Challenges and potential solutions, Marine Technology Society Journal 45 (2011) 31–40. doi:10.4031/MTSJ.45.4.2.
4. M. Sfakiotakis, D. M. Lane, J. Davies, Review of fish swimming modes for aquatic locomotion, IEEE J. Ocean. Eng. 24 (1999) 237–252. doi:10.1109/48.757275.
5. G. Lauder, P. Madden, I. Hunter, J. Tangorra, N. Davidson, L. Proctor, R. Mittal, H. Dong, M. Bozkurttas, Design and performance of a fish fin-like propulsor for AUVs, in: Proceedings of 14th International Symposium on Unmanned Untethered Submersible Technology, Durham, NH. URL https://pdfs.semanticscholar.org/6f56/4e49fc899fe93856b30332b425f37e19a1ac.pdf.
6. M. Triantafyllou, G. Triantafyllou, An efficient swimming machine, Scientific American – SCI AMER 272 (1995) 64–70. doi:10.1038/scientificamerican0395-64.
7. D. S. Barrett, M. Triantafyllou, D. K. P. Yue, M. A. Grosenbaugh, M. J. Wolfgang, Drag reduction in fish-like locomotion, Journal of Fluid Mechanics 392 (1999) 183–212. doi:10.1017/S0022112099005455.
8. I. Yamamoto, Robotic fish development for the next generation underwater vehicle, Advances in Science and Technology 101 (2016) 95–103. doi:10.4028/www.scientific.net/AST.101.95.
9. J. Liu, H. Hu, Mimicry of sharp turning behaviours in a robotic fish, in: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005, pp. 3318–3323. doi:10.1109/ROBOT.2005.1570622.
10. P. Valdivia y Alvarado, K. Youcef-Toumi, Design of machines with compliant bodies for biomimetic locomotion in liquid environments, J. Dyn. Sys., Meas., Control 128 (1) (2006) 3–13. doi:10.1115/1.2168476.
11. C. Rossi, J. Colorado, W. Coral, A. Barrientos, Bending continuous structures with smas: A novel robotic fish design, Bioinspiration & Biomimetics 6 (2011) 045005. doi:10.1088/1748-3182/6/4/045005.
12. D. Siebelts, A. Kater, T. Meurer, Modeling and motion planning for an artificial fishtail, IFAC-PapersOnLine 51 (2) (2018) 319–324. doi:10.1016/j.ifacol.2018.03.055.
13. A. D. Marchese, C. D. Onal, D. Rus, Autonomous soft robotic fish capable of escape maneuvers using fluidic elastomer actuators, Soft Robotics 1 (1) (2014) 75–87. doi:10.1089/soro.2013.0009.
14. D. G. Meyer, S. Srinivasan, Balancing and model reduction for second-order form linear systems, IEEE Trans. Autom. Control 41 (11) (1996) 1632–1644. doi:10.1109/9.544000.
15. V. Chahlaoui, K. A. Gallivan, A. Vandendorpe, P. Van Dooren, Model reduction of second-order system, in: P. Benner, V. Mehrmann, D. Sorensen (Eds.), Dimension Reduction of Large-Scale Systems, Vol. 45 of Lect. Notes Comput. Sci. Eng., Springer-Verlag, Berlin/Heidelberg, Germany, 2005, pp. 149–172. doi:10.1007/3-540-27909-1_6.
16. Y. Chahlaoui, D. Lemonnier, A. Vandendorpe, P. Van Dooren, Second-order balanced truncation, Linear Algebra Appl. 415 (2–3) (2006) 373–384. doi:10.1016/j.laa.2004.03.032.
17. T. Reis, T. Stykel, Balanced truncation model reduction of second-order systems, Math. Comput. Model. Dyn. Syst. 14 (5) (2008) 391–406. doi:10.1080/13873950701844170.
18. E. J. Grimme, Krylov projection methods for model reduction, Ph. D. Thesis, Univ. of Illinois at Urbana-Champaign, USA (1997). URL https://perso.uclouvain.be/paul.vandooren/ThesisGrimme.pdf.
19. A. C. Antoulas, Approximation of Large-Scale Dynamical Systems, Vol. 6 of Adv. Des. Control, SIAM Publications, Philadelphia, PA, 2005. doi:10.1137/1.9780898718713.
20. H. K. F. Panzer, Model order reduction by Krylov subspace methods with global error bounds and automatic choice of parameters, Dissertation, Technische Universität München, Munich, Germany (2014). URL https://mediatum.ub.tum.de/doc/1207822/1207822.pdf.
21. M. Lehner, P. Eberhard, A two-step approach for model reduction in flexible multibody dynamics, Multibody Syst. Dyn. 17 (2–3) (2007) 157–176. doi:10.1007/s11044-007-9039-5.
22. J. Fehr, P. Eberhard, Error-controlled model reduction in flexible multibody dynamics, J. Comput. Nonlinear Dynam. 5 (3) (2010) 031005–1–031005–8. doi:10.1115/1.4001372.
23. P. Benner, P. Kürschner, J. Saak, Frequency-limited balanced truncation with low-rank approximations, SIAM J. Sci. Comput. 38 (1) (2016) A471–A499. doi:10.1137/15M1030911.
24. P. Benner, S. W. R. Werner, Limited balanced truncation for large-scale sparse second-order systems (version 1.0) (Feb. 2019). doi:10.5281/zenodo.2553926.
25. S. Wyatt, Issues in interpolatory model reduction: Inexact solves, second order systems and DAEs, Ph. D. thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, USA (May 2012). URL https://vtechworks.lib.vt.edu/bitstream/handle/10919/27668/Wyatt_SA_D_2012.pdf?sequence=1.
26. P. Vuillemin, Frequency-limited model approximation of large-scale dynamical models, Ph. D. thesis, Université de Toulouse (2014). URL http://www.pierre-vuillemin.fr/publi/vuillemin_thesis_2014.pdf.
27. D. Petterson, A nonlinear optimization approach to $\mathcal{H}_2$-optimal modeling and control, Dissertation, Linköping University (2013). URL http://liu.diva-portal.org/smash/get/diva2:647068/FULLTEXT01.pdf.
28. T. Breiten, C. Beattie, S. Gugercin, Near-optimal frequency-weighted interpolatory model reduction, Syst. Control Lett. 78 (2015) 8–18. doi:10.1016/j.sysconle.2015.01.005.

29. B. Anić, C. Beattie, S. Gugercin, A. C. Antoulas, Interpolatory weighted-$\mathcal{H}_2$ model reduction, Automatica J. IFAC 49 (5) (2013) 1275–1280. doi:10.1016/j.automatica.2013.01.040.
30. C. Hartmann, V.-M. Vulcanov, C. Schütte, Balanced truncation of linear second-order systems: a Hamiltonian approach, Multiscale Model. Simul. 8 (4) (2010) 1348–1367. doi:10.1137/080732717.
31. J. Saak, M. Köhler, P. Benner, M-M.E.S.S.-1.0.1 – the matrix equations sparse solvers library, see also: https://www.mpi-magdeburg.mpg.de/projects/mess (Apr. 2016). doi:10.5281/zenodo.50575.
32. P. Benner, S. W. R. Werner, MORLAB – Model Order Reduction LABoratory (version 4.0), see also: http://www.mpi-magdeburg.mpg.de/projects/morlab (2018). doi:10.5281/zenodo.1574083.
33. MOREMBS (model order reduction of elastic multibody systems). URL https://www.itm.uni-stuttgart.de/en/software/morembs/.
34. J. Fehr, D. Grunert, P. Holzwarth, B. Fröhlich, N. Walker, P. Eberhard, Morembs—a model order reduction package for elastic multibody systems and beyond, in: Reduced-Order Modeling (ROM) for Simulation and Optimization: Powerful Algorithms as Key Enablers for Scientific Computing, Springer International Publishing, Cham, 2018, pp. 141–166. doi:10.1007/978-3-319-75319-5_7.
35. A. Castagnotto, M. Cruz Varona, L. Jeschek, B. Lohmann, sss & sssMOR: Analysis and reduction of large-scale dynamic systems in MATLAB, at-Automatisierungstechnik 65 (2) (2017) 134–150. doi:10.1515/auto-2016-0137.
36. E. B. Rudnyi, J. G. Korvink, Model order reduction for large scale engineering models developed in ANSYS, in: J. Dongarra, K. Madsen, J. Waśniewski (Eds.), Applied Parallel Computing. State of the Art in Scientific Computing, Springer, Berlin, Heidelberg, 2006, pp. 349–356. doi:10.1007/11558958_41.
37. J. Riegel, W. Mayer, Y. van Havre, Freecad, version: 0.16 (2017). URL http://www.Freecadweb.org.
38. O. C. Zienkiewicz, R. L. Taylor, J. Z. Zhu, The Finite Element Method: Its Basis and Fundamentals, 6th Edition, Butterworth-Heinemann, 2005.
39. M. E. Gurtin, An Introduction to Continuum Mechanics, Vol. 158 of Mathematics in Science and Engineering, Academic Press, 1981.
40. G. A. Holzapfel, Nonlinear Solid Mechanics: A Continuum Approach for Engineering, Wiley, 2000.
41. Csimsoft, trelis, version: 16.1.3, (2016). URL http://www.csimsoft.com/trelis.
42. S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, PETSc users manual, Tech. Rep. ANL-95/11 – Revision 3.9, Argonne National Laboratory (2018).
43. S. Balay, W. D. Gropp, L. C. McInnes, B. F. Smith, Efficient management of parallelism in object oriented numerical software libraries, in: E. Arge, A. M. Bruaset, H. P. Langtangen (Eds.), Modern Software Tools in Scientific Computing, Birkhäuser Press, 1997, pp. 163–202. doi:10.1007/978-1-4612-1986-6_8.
44. L. D. Dalcin, R. R. Paz, P. A. Kler, A. Cosimo, Parallel distributed computing using Python, Advances in Water Resources 34 (9) (2011) 1124–1139, new Computational Methods and Software Tools. doi:10.1016/j.advwatres.2011.04.013.
45. V. Hernandez, J. E. Roman, V. Vidal, SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems, ACM Trans. Math. Software 31 (3) (2005) 351–362. doi:10.1145/1089014.1089019.
46. J. E. Roman, C. Campos, E. Romero, A. Tomas, SLEPc users manual, Tech. Rep. DSIC-II/24/02 – Revision 3.10, D. Sistemes Informàtics i Computació, Universitat Politècnica de València (2018).
47. F. Rathgeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, A. T. T. McRae, G.-T. Bercea, G. R. Markall, P. H. J. Kelly, Firedrake: automating the finite element method by composing abstractions, ACM Trans. Math. Softw. 43 (3) (2016) 24:1–24:27. doi:10.1145/2998441.
48. F. Luporini, A. L. Varbanescu, F. Rathgeber, G.-T. Bercea, J. Ramanujam, D. A. Ham, P. H. J. Kelly, Cross-loop optimization of arithmetic intensity for finite element local assembly, ACM Transactions on Architecture and Code Optimization 11 (4) (2015) 57:1–57:25. doi:10.1145/2687415.
49. B. Hendrickson, R. Leland, A multilevel algorithm for partitioning graphs, in: Supercomputing ’95: Proceedings of the 1995 ACM/IEEE Conference on Supercomputing (CDROM), ACM Press, New York, 1995, p. 28. doi:10.1145/224170.224228.
50. L. Meirovitch, Principles and Techniques of Vibrations, Prentice Hall, 1997.
51. E. J. Davison, A method for simplifying linear dynamic systems, IEEE Trans. Autom. Control AC–11 (1966) 93–101. doi:10.1109/TAC.1966.1098264.
52. N. Martins, L. T. G. Lima, H. J. C. P. Pinto, Computing dominant poles of power system transfer functions, IEEE Trans. Power Syst. 11 (1) (1996) 162–170. doi:10.1109/59.486093.
53. J. Rommes, N. Martins, Computing transfer function dominant poles of large-scale second-order dynamical systems, IEEE Trans. Power Syst. 21 (4) (2006) 1471–1483. doi:10.1109/TPWRS.2006.881154.
54. P. Benner, P. Kürschner, Z. Tomljanović, N. Truhar, Semi-active damping optimization of vibrational systems using the parametric dominant pole algorithm, Z. Angew. Math. Mech. 96 (5) (2016) 604–619. doi:10.1002/zamm.201400158.
55. C. A. Beattie, P. Benner, $\mathcal{H}_2$-optimality conditions for structured dynamical systems, Preprint MPIMD/14-18, Max Planck Institute Magdeburg (2014). URL https://www2.mpi-magdeburg.mpg.de/preprints/2014/18/.
56. P. Benner, S. W. R. Werner, SOMDDPA – Second-Order Modally Damped Dominant Pole Algorithm (version 1.0) (Feb. 2019). doi:10.5281/zenodo.2553902.
57. C. A. Beattie, S. Gugercin, Krylov-based model reduction of second-order systems with proportional damping, in: Proceedings of the 44th IEEE Conference on Decision and Control, 2005, pp. 2278–2283. doi:10.1109/CDC.2005.1582501.
58. B. Salimbahrami, Structure preserving order reduction of large scale second order models, Dissertation, Technische Universität München, München (2005). URL https://mediatum.ub.tum.de/doc/601950/00000941.pdf.
59. B. Salimbahrami, B. Lohmann, Order reduction of large scale second-order systems using Krylov subspace methods, Linear Algebra Appl. 415 (2–3) (2006) 385–405. doi:10.1016/j.laa.2004.12.013.

60. T. Bonin, H. Faßbender, A. Soppa, M. Zaeh, A fully adaptive rational global Arnoldi method for the model-order reduction of second-order MIMO systems with proportional damping, Math. Comput. Simulat. 122. doi:10.1016/j.matcom.2015.08.017.
61. B. C. Moore, Principal component analysis in linear systems: controllability, observability, and model reduction, IEEE Trans. Autom. Control AC–26 (1) (1981) 17–32. doi:10.1109/TAC.1981.1102568.
62. P. Benner, J. Saak, Efficient balancing-based MOR for large-scale second-order systems, Math. Comput. Model. Dyn. Syst. 17 (2) (2011) 123–143. doi:10.1080/13873954.2010.540822.
63. P. Benner, P. Kürschner, J. Saak, Improved second-order balanced truncation for symmetric systems, IFAC Proceedings Volumes (7th Vienna International Conference on Mathematical Modelling) 45 (2) (2012) 758–762. doi:10.3182/20120215-3-AT-3016.00134.
64. P. Benner, P. Kürschner, J. Saak, An improved numerical method for balanced truncation for symmetric second order systems, Math. Comput. Model. Dyn. Syst. 19 (6) (2013) 593–615. doi:10.1080/13873954.2013.794363.
65. W. Gawronski, J.-N. Juang, Model reduction in limited time and frequency intervals, Int. J. Syst. Sci. 21 (2) (1990) 349–376. doi:10.1080/00207729008910366.
66. K. Haider, A. Ghafoor, M. Imran, F. M. Malik, Frequency interval Gramians based structure preserving model reduction for second-order systems, Asian J. Control 20 (2) (2018) 790–801. doi:10.1002/asjc.1598.
67. The OpenBLAS Project, Openblas. URL https://www.openblas.net/.
68. Software used in 'Firedrake version for: A Comparison of Second-Order Model Order Reduction Methods for an Artificial Fishtail' (feb 2019). doi:10.5281/zenodo.2564747.
69. Python Software Foundation, Python language reference, version 3.7. URL https://docs.python.org/3.7/.
70. T. Oliphant, NumPy: A guide to NumPy, USA: Trelgol Publishing (2006–). URL http://www.numpy.org/.
71. E. Jones, T. Oliphant, P. Peterson, et al., SciPy: Open source scientific tools for Python (2001–). URL http://www.scipy.org/.
72. The MathWorks, Inc., MATLAB. URL http://www.matlab.com.
73. D. Siebelts, A. Kater, T. Meurer, J. Andrej, Matrices for an artificial fishtail (Feb. 2019). doi:10.5281/zenodo.2558728.
74. J. Saak, D. Siebelts, S. W. R. Werner, Second order reduction comparison for the fishtail model (Feb. 2019). doi:10.5281/zenodo.2563874.
75. J. Saak, D. Siebelts, S. W. R. Werner, Reduction results for an artificial fish tail (Feb. 2019). doi:10.5281/zenodo.2564050.
76. D. Siebelts, A. Kater, T. Meurer, J. Andrej, Mesh for an artificial fishtail (Feb. 2019). doi:10.5281/zenodo.2565793.

Bionotes

Jens Saak
Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39106 Magdeburg, Germany
saak@mpi-magdeburg.mpg.de

Jens Saak studied industrial mathematics at the Zentrum für Technomathematik (ZeTeM) at University of Bremen until 2003. He then moved to the Department of Mathematics of TU Chemnitz, joining the Mathematics in Industry and Technology group (Prof. Dr. Peter Benner), where he received his PhD in 2009. In 2010 Jens joined Prof. Benner moving to the Max Planck Institute for Dynamics of Complex Technical Systems, where he is heading the Teams on Matrix Equations and Scientific Computing in the department for Computational Methods in Systems and Control Theory. His main interest are fast, efficient, accurate and high performance implementations of algebraic and differential matrix equations and their applications in control and model order reduction of large-scale systems, driven by partial differential equations, especially in the context of demanding industrial and (mechanical) engineering environments.

Dirk Siebelts
Chair of Automatic Control, Faculty of Engineering, Kiel University, Kaiserstraße 2, 24143 Kiel, Germany
disi@tf.uni-kiel.de

Dirk Siebelts studied electrical engineering and information technology at Christian-Albrechts-Universität zu Kiel (Kiel University) from 2011 until 2017. After finishing his master studies he joined the Chair of Automatic Control at Kiel University as a doctoral student where he is currently working on control design based on partial differential equations with applications in manufacturing processes.

Steffen W. R. Werner
Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39106 Magdeburg, Germany
werner@mpi-magdeburg.mpg.de

Steffen W. R. Werner finished his master studies in mathematics at the Otto von Guericke University Magdeburg in 2016. Thereafter, he joined the Max Planck Institute for Dynamics of Complex Technical Systems, where he is working on the application of structure-preserving model order reduction methods for mechanical systems.