Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 55-20 (2022) 163–168

Structured vector fitting framework for mechanical systems ⋆

Steffen W. R. Werner * **Ion Victor Gosea** ** **Serkan Gugercin** ***

* *Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA. (e-mail: steffen.werner@nyu.edu)*
** *Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany. (e-mail: gosea@mpi-magdeburg.mpg.de)*
*** *Department of Mathematics and Division of Computational Modeling and Data Analytics, Academy of Data Science, Virginia Tech, Blacksburg, VA 24061, USA. (e-mail: gugercin@vt.edu)*

Abstract: We develop a structure-preserving formulation of the data-driven vector fitting algorithm for modally damped mechanical systems. Using the structured pole-residue form of the transfer function of modally damped second-order systems, we propose two potential, structured extensions of the barycentric formula for system transfer functions. Integrating these new forms within the classical vector fitting algorithm leads to the formulation of two new algorithms. These allow the computation of modally damped mechanical systems from data in a least-squares fashion, where the learned model is guaranteed to have the desired structure. We test the proposed algorithms on two benchmark models.

Keywords: data-driven modeling, mechanical systems, reduced-order modeling, vector fitting, least-squares fit, barycentric forms

1. INTRODUCTION

Data-driven reduced-order modeling (DD-ROM) is essential in constructing high-fidelity compact models to approximate the underlying physical phenomena when an explicit model, a state-space formulation with access to internal variables, is not available, yet abundant input/output data are. Thus, DD-ROM circumvents the need to access an exact description of the original model and is applicable when traditional intrusive projection-based model reduction is not. As in the latter case, it is important that the learned model inherits the physical meaning and structures of the system that has generated the data. This is the setup we are interested in here. Our goal is to develop a data-driven structure-preserving modeling framework for mechanical systems described by second-order dynamics.

Data in our setting corresponds to transfer function (frequency domain) samples of the underlying mechanical system. Let $H(s)$ denote this transfer function and let ξ_i denote the sampling frequencies (points). Thus, we assume access to the data (measurements) $h_i = H(\xi_i)$, for $i = 1, 2, \ldots, \ell$. The goal of DD-ROM in this setting is to construct a reduced transfer function (a low-order rational function) $\widehat{H}(s)$ such that $\widehat{H}(\xi_i) \approx h_i = H(\xi_i)$ in an appropriate measure. We will call this unstructured (or first-order fitting) in this paper since the only requirement in this case is that $\widehat{H}(s)$ is a rational function and thus corresponds to a transfer function with a first-order state-space form. In this setting, the barycentric rational form of the approximant plays a crucial role; see Berrut and Trefethen (2004). The Loewner framework from Antoulas and Anderson (1986); Mayo and Antoulas (2007) that enforces interpolation of the data, the Vector Fitting (VF) algorithm from Gustavsen and Semlyen (1999) that minimizes a least-squares distance, and the AAA algorithm from Nakatsukasa et al. (2018) that combines interpolation and least squares are just three of the many techniques for rational data fitting. We refer the reader to (Rodriguez, 2020, Sec. 2.1) for further references.

Second-order systems are an important class of structured dynamical systems used to describe, for example, the dynamics of mechanical systems, and in particular, their vibrational response. Since the underlying second-order structure corresponds to important physical properties, retaining this structure is vital so that the learned model is physically meaningful. Therefore, given the frequency response samples of such systems, our goal is to construct a *structure-preserving DD-ROM*, in the sense that the learned model can be interpreted as the transfer function of a second-order (mechanical) system. Note that not every rational function can be written as the transfer function of a second-order system (although the reverse is true). There have been some recent works on constructing data-driven second-order models in the interpolatory Loewner framework; see Schulze et al. (2018); Benner et al. (2020).

⋆ Gosea and Werner, while he was at Max Planck Institute Magdeburg, have been supported in parts by the German Research Foundation (DFG) Research Training Group 2297 "MathCoRe", Magdeburg. Gugercin was supported in parts by the National Science Foundation under Grant No. DMS-1923221.

10.1016/j.ifacol.2022.09.089

There is however a larger variety of projection-based structure-preserving model reduction methods for second-order systems. We refer the reader to Saak et al. (2019); Werner (2021) and the references therein for details on the projection-based approaches that are not explicitly considered here.

In this paper, we focus on structure-preserving second-order DD-ROM using the least-squares measure. More specifically, we enforce modal-damping structure in the learned model. We achieve this goal by extending the VF algorithm to the structured setting. Up to now, VF has been developed to produce unstructured rational approximants. We revise the barycentric formula behind the VF approximant such that upon convergence the learned model has the desired second-order structure. This new formulation of the barycentric form leads to a sequence of linear least-squares problems whose structure also inherit the underlying second-order dynamics.

The rest of the paper is organized as follows: After providing an overview of the classical VF approach and modally damped second-order systems in Section 2, we develop the modified barycentric forms and the resulting structure-preserving VF approaches together with the corresponding proposed numerical algorithms in Section 3. The proposed methods are then tested in two benchmark examples in Section 4, followed by the conclusions and future research directions in Section 5.

2. BACKGROUND

In this section, we provide a brief overview of the classical vector fitting algorithm and summarize the key structural features of the special class of mechanical systems under consideration.

2.1 Classical vector fitting approach

Assume that one has access to the samples of the transfer function of an underlying single-input/single-output (SISO) dynamical system to be modeled, $H(s)$, at the sampling points (frequencies) $\xi_1, \xi_2, \ldots, \xi_\ell \in \mathbb{C}$. Given the data $\{H(\xi_i)\}_{i=1}^{\ell}$, the goal is to construct (learn) a degree-r scalar rational function $\widehat{H}(s)$ to solve the nonlinear rational least-squares (LS) problem

$$\min_{\widehat{H}} \sum_{i=1}^{\ell} |\widehat{H}(\xi_i) - H(\xi_i)|^2. \tag{1}$$

Let $\widehat{H}(s) = \frac{n(s)}{d(s)}$ where $n(s)$ and $d(s)$ are, respectively, degree-$(r-1)$ and degree-r polynomials in s. In other words, $H(s)$ is parametrized by its denominator and numerator coefficients. Inserting this form of $\widehat{H}(s)$ into (1), one can rewrite the nonlinear LS error to minimize as

$$\sum_{i=1}^{\ell} |\widehat{H}(\xi_i) - H(\xi_i)|^2 = \sum_{i=1}^{\ell} \frac{1}{|d(\xi_i)|^2} |n(\xi_i) - d(\xi_i) H(\xi_i)|^2.$$

The nonlinearity results from the dependence of the error on $d(s)$. To solve this nonlinear LS problem, starting with an initial guess of $\widehat{H}^{(0)}(s) = \frac{d^{(0)}(s)}{n^{(0)}(s)}$, Sanathanan and Koerner (1963) proposed an iterative scheme where in the k-th step the error term (1) is replaced by

$$\begin{aligned} &\sum_{i=1}^{\ell} |\widehat{H}^{(k)}(\xi_i) - H(\xi_i)|^2 = \\ &\qquad \sum_{i=1}^{\ell} \frac{1}{|d^{(k-1)}(\xi_i)|^2} |n^{(k)}(\xi_i) - d^{(k)}(\xi_i) H(\xi_i)|^2. \end{aligned} \tag{2}$$

Note that the new error term (2) is now linear in the variables $n^{(k)}(s)$ and $d^{(k)}(s)$. Therefore, the SK iteration in Sanathanan and Koerner (1963) converts the original nonlinear LS problem (1) into solving a sequence of weighted linear LS problems (2).

There are various equivalent forms to represent the rational function $\widehat{H}(s)$. One can work with numerator and denominator coefficients as unknowns, or the poles and residues, for example. A numerically efficient formulation is the so-called barycentric representation; see Berrut and Trefethen (2004). Let $\widehat{H}^{(k)}(s)$ denote the iterate in the k-th step of the SK iteration as above. Also let $\lambda_1^{(k)}, \ldots, \lambda_r^{(k)}$ be mutually distinct points. Then, $\widehat{H}^{(k)}(s)$ can be written in the barycentric form as

$$\widehat{H}^{(k)}(s) = \frac{n^{(k)}(s)}{d^{(k)}(s)} = \frac{\sum_{j=1}^{r} \frac{\phi_j^{(k)}}{s - \lambda_j^{(k)}}}{1 + \sum_{j=1}^{r} \frac{\varphi_j^{(k)}}{s - \lambda_j^{(k)}}}, \tag{3}$$

where $\{\phi_j^{(k)}\}_{j=1}^{r}$ and $\{\varphi_j^{(k)}\}_{j=1}^{r}$ are the barycentric weights. Note that the $\lambda_j^{(k)}$'s are not the poles of $\widehat{H}(s)$. We refer the reader to, e.g., Drmač et al. (2015b) to switch between the pole-residue form and the barycentric form.

Now inserting $n^{(k)}(s)$ and $d^{(k)}(s)$ from (3) into (2), in the k-th step of the SK iteration, one needs to solve the weighted linear LS problem

$$\min_{x^{(k)}} \|\Delta^{(k)}(A^{(k)} x^{(k)} - h)\|_2^2, \tag{4}$$

where

$$\Delta^{(k)} = \operatorname{diag}\left(\frac{1}{|d^{(k)}(\xi_1)|}, \ldots, \frac{1}{|d^{(k)}(\xi_\ell)|} \right), \quad h = \begin{bmatrix} H(\xi_1) \\ \vdots \\ H(\xi_\ell) \end{bmatrix}, \tag{5}$$

$$A^{(k)} = \begin{bmatrix} \frac{1}{\xi_1 - \lambda_1^{(k)}} & \cdots & \frac{1}{\xi_1 - \lambda_r^{(k)}} & \frac{-H(\xi_1)}{\xi_1 - \lambda_1^{(k)}} & \cdots & \frac{-H(\xi_1)}{\xi_1 - \lambda_r^{(k)}} \\ \vdots & & \vdots & \vdots & & \vdots \\ \frac{1}{\xi_\ell - \lambda_1^{(k)}} & \cdots & \frac{1}{\xi_\ell - \lambda_r^{(k)}} & \frac{-H(\xi_\ell)}{\xi_\ell - \lambda_1^{(k)}} & \cdots & \frac{-H(\xi_\ell)}{\xi_\ell - \lambda_r^{(k)}} \end{bmatrix}, \tag{6}$$

for the solution vector

$$x^{(k)} = \begin{bmatrix} \phi_1^{(k)} & \cdots & \phi_r^{(k)} & \varphi_1^{(k)} & \cdots & \varphi_r^{(k)} \end{bmatrix}^{\mathsf{T}},$$

which forms $\widehat{H}^{(k)}(s)$ at the k-th step. In addition to incorporating the barycentric form into the SK iteration, Gustavsen and Semlyen (1999) have also observed that the only restrictions on $\{\lambda_j^{(k)}\}_{j=1}^{r}$ are to be distinct and they can be updated at every step. This is precisely what Gustavsen and Semlyen (1999) have proposed, leading to the *Vector Fitting (VF)* algorithm. VF updates $\{\lambda_j^{(k)}\}_{j=1}^{r}$ as the roots of denominator $d^{(k)}(s)$. Making again use of the barycentric representation (3), these roots are actually the eigenvalues of $\widehat{A}^{(k)} - \widehat{G}^{(k)}\widehat{C}^{(k)}$, where

Algorithm 1 (Unstructured) Vector Fitting (VF)

Input: Vector h of data samples (5), initial guess for $\{\lambda_j^{(1)}\}_{j=1}^r$.
Output: Learned ROM $\widehat{H}(s) = \widehat{C}(sI_r - \widehat{A})^{-1}\widehat{B}$.

1: Initialize $\Delta^{(1)} = I_\ell$ and $k = 1$.
2: **while** not converged **do**
3: Construct the coefficient matrix $A^{(k)}$ in (6).
4: Solve the weighted linear least-squares problem (4).
5: Update the expansion points $\{\lambda_j^{(k+1)}\}_{j=1}^r$ to be the eigenvalues of $\widehat{A}^{(k)} - \widehat{G}^{(k)}\widehat{C}^{(k)}$ using (7) and (8).
6: Update the weighting matrix $\Delta^{(k+1)}$ by (5).
7: Increment $k \leftarrow k + 1$.
8: **end while**
9: Set the final ROM matrices to be $\widehat{A} = \operatorname{diag}(\lambda_1^{(k)}, \ldots, \lambda_r^{(k)})$, $\widehat{B} = \begin{bmatrix}\phi_1^{(k)} & \ldots & \phi_r^{(k)}\end{bmatrix}^\mathsf{T}$ and $\widehat{C} = \begin{bmatrix}1 & \ldots & 1\end{bmatrix}^\mathsf{T}$.

$$\widehat{A}^{(k)} = \operatorname{diag}(\lambda_1^{(k)}, \ldots, \lambda_r^{(k)}), \tag{7}$$

$$\widehat{G}^{(k)} = \begin{bmatrix}\varphi_1^{(k)} & \ldots & \varphi_r^{(k)}\end{bmatrix}^\mathsf{T} \quad \text{and} \quad \widehat{C}^{(k)} = [1 \ \ldots \ 1]^\mathsf{T}. \tag{8}$$

If the algorithm converges, due to the $\{\lambda_j^{(k)}\}_{j=1}^r$ updating strategy, $d^{(k)}(s) \to 1$ and thus the final approximation is obtained in the pole-residue form with the denominator in (3) being 1. The resulting method is summarized in Algorithm 1, and we refer the reader to Gustavsen and Semlyen (1999), (Grivet-Talocia and Gustavsen, 2015, Chap. 7) and Drmač et al. (2015a) for further details.

2.2 Modally damped second-order systems

Next, we take a look at the pole-residue formulation of the structured system class considered here, namely the modally damped second-order systems. As in the previous section, for simplicity we restrict the analysis to the SISO case. Assume we have a second-order system of the form

$$\begin{aligned} M\ddot{q}(t) + E\dot{q}(t) + Kq(t) &= B_\mathrm{u}u(t), \\ y(t) &= C_\mathrm{p}q(t), \end{aligned}$$

with $M, E, K \in \mathbb{R}^{n\times n}$, $B_\mathrm{u} \in \mathbb{R}^n$, $C_\mathrm{p}^\mathsf{T} \in \mathbb{R}^n$, and modal damping $EM^{-1}K = KM^{-1}E$ as in Beattie and Benner (2014). Note here that for the mechanical system case one additionally has $M = M^\mathsf{T} > 0$, $E = E^\mathsf{T} \geq 0$, and $K = K^\mathsf{T} > 0$. This assumption is not necessary in general and instead we only assume that the pencil $\lambda M - K$ is diagonalizable, since with modal damping all three system matrices are simultaneously diagonalizable. First, we consider the generalized eigenvalue problems

$$KX = MX\Omega^2, \quad K^\mathsf{T}Y = M^\mathsf{T}Y\Omega^2,$$

where the eigenvector matrices X and Y are scaled such that

$$Y^\mathsf{T}MX = \Omega^{-1}, \quad Y^\mathsf{T}KX = \Omega,$$

with $\Omega = \operatorname{diag}(\omega_1, \ldots, \omega_n)$. Due to modal damping, the damping matrix can also be diagonalized such that

$$Y^\mathsf{T}EX = 2\Psi,$$

where $\Psi = \operatorname{diag}(\psi_1, \ldots, \psi_n)$ are the damping ratios of the system. Then, the transfer function $H(s)$ satisfies

$$\begin{aligned} H(s) &= C_\mathrm{p}(s^2M + sE + K)^{-1}B_\mathrm{u} \\ &= C_\mathrm{p}X(s^2\Omega^{-1} + 2s\Psi + \Omega)^{-1}Y^\mathsf{T}B_\mathrm{u} \\ &= \sum_{j=1}^n \frac{\omega_j\phi_j^\pm}{s^2 + 2\psi_j\omega_j s + \omega_j^2} \\ &= \sum_{j=1}^n \frac{\omega_j\phi_j^\pm}{(s-\lambda_j^+)(s-\lambda_j^-)}, \end{aligned} \tag{9}$$

where the pairwise poles of the system are given by

$$\lambda_j^\pm = -\omega_j\psi_j \pm \omega_j\sqrt{\psi_j^2 - 1}. \tag{10}$$

Since every second-order system can also be written in its first-order form, we can write $H(s)$ in the generic pole-residue formulation as

$$H(s) = \sum_{j=1}^{2n} \frac{\phi_j}{s-\lambda_j} = \sum_{j=1}^n \frac{\phi_j^+}{s-\lambda_j^+} + \sum_{j=1}^n \frac{\phi_j^-}{s-\lambda_j^-}. \tag{11}$$

Note that modal damping and the second-order structure enforce additional properties in the generic pole-residue form, which means that only for the underlying second-order systems those two formulations, i.e., (9) and (11), are equivalent. An important advantage of (9) is the enforcement of the underlying system structure.

3. SECOND-ORDER VECTOR FITTING ALGORITHMS

The classical VF algorithm as outlined in Section 2.1 produces an unstructured rational LS fit. In this section, we will develop a structured version of VF to model second-order modally damped system. We will achieve this goal by employing the special pole-residue formulation (9) in VF and by modifying the corresponding barycentric form appearing in VF. We will propose two formulations for the revised barycentric form and analyze both forms. At the end of the newly developed structured VF iteration, the learned model will be guaranteed to have the modally damped form.

3.1 Partially structured barycentric form

In our first approach, we develop a second-order VF formulation for modally damped systems using a partially structured barycentric formulation. The method computes a second-order system by enforcing $\widehat{H}^{(k)}$, the reduced-order model at iteration step k, to have the transfer function

$$\widehat{H}^{(k)}(s) = \frac{\displaystyle\sum_{j=1}^r \frac{\omega_j^{(k)}\phi_j^{\pm,(k)}}{(s-\lambda_j^{+,(k)})(s-\lambda_j^{-,(k)})}}{1 + \displaystyle\sum_{j=1}^r \frac{\varphi_j^{+,(k)}}{s-\lambda_j^{+,(k)}} + \sum_{j=1}^r \frac{\varphi_j^{-,(k)}}{s-\lambda_j^{-,(k)}}}, \tag{12}$$

In other words, the form (12) replaces (3) in VF. The motivation for the revised form (12) stems from the desired modally damped structure. Recall that as classical VF converges, the denominator converges to 1 and the numerator yields the transfer function of the final reduced model. In the structured form (12), we keep the denominator as before in the classical pole-residue form (11). However, the numerator is replaced by the structured pole-residue form (9). Therefore, upon convergence, the final reduced

$$A_{\mathrm{so1}}^{(k)} = \begin{bmatrix} \frac{\omega_1^{(k)}}{(\xi_1-\lambda_1^{+,(k)})(\xi_1-\lambda_1^{-,(k)})} & \cdots & \frac{\omega_r^{(k)}}{(\xi_1-\lambda_r^{+,(k)})(\xi_1-\lambda_r^{-,(k)})} & \frac{-H(\xi_1)}{(\xi_1-\lambda_1^{+,(k)})} & \frac{-H(\xi_1)}{(\xi_1-\lambda_1^{-,(k)})} & \cdots & \frac{-H(\xi_1)}{(\xi_1-\lambda_r^{+,(k)})} & \frac{-H(\xi_1)}{(\xi_1-\lambda_r^{-,(k)})} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots \\ \frac{\omega_1^{(k)}}{(\xi_\ell-\lambda_1^{+,(k)})(\xi_\ell-\lambda_1^{-,(k)})} & \cdots & \frac{\omega_r^{(k)}}{(\xi_\ell-\lambda_r^{+,(k)})(\xi_\ell-\lambda_r^{-,(k)})} & \frac{-H(\xi_\ell)}{(\xi_\ell-\lambda_1^{+,(k)})} & \frac{-H(\xi_\ell)}{(\xi_\ell-\lambda_1^{-,(k)})} & \cdots & \frac{-H(\xi_\ell)}{(\xi_\ell-\lambda_r^{+,(k)})} & \frac{-H(\xi_\ell)}{(\xi_\ell-\lambda_r^{-,(k)})} \end{bmatrix} \tag{13}$$

model, given by the numerator in (12), is guaranteed to have the desired form.

We now discuss the structure of the resulting second-order VF algorithm. As in Section 2.1, using the relaxation step (2) we solve a sequence of weighted linear LS problems of the form

$$\min_{\tilde{x}^{(k)}} \|\Delta^{(k)}(A_{\mathrm{so1}}^{(k)}\tilde{x}^{(k)} - h)\|_2^2 \tag{14}$$

for the solution vector

$$\tilde{x}^{(k)} = \begin{bmatrix} \phi_1^{\pm,(k)} & \cdots & \phi_r^{\pm,(k)} & \varphi_1^{+,(k)} & \varphi_1^{-,(k)} & \cdots & \varphi_r^{+,(k)} & \varphi_r^{-,(k)} \end{bmatrix}^{\mathsf{T}},$$

which determines $\widehat{H}^{(k)}(s)$, where h and $\Delta^{(k)}$ are as in (5), and the new coefficient matrix $A_{\mathrm{so1}}^{(k)}$ as in (13). The new coefficient matrix $A_{\mathrm{so1}}^{(k)}$ encodes the underlying second-order structure. Consequently, we replace Steps 3 and 4 in Algorithm 1 with (13) and (14) in the proposed second-order VF iteration. Using (10), the stiffness and damping coefficients of the pole pairs are given by

$$\begin{aligned} \omega_j^{(k)} &= \sqrt{\lambda_j^{+,(k)}\lambda_j^{-,(k)}}, \\ \psi_j^{(k)} &= -\frac{1}{2\omega_j^{(k)}}(\lambda_j^{+,(k)} + \lambda_j^{-,(k)}), \end{aligned}$$

for $j = 1, \ldots, r$. This formulation is needed in constructing $A_{\mathrm{so1}}^{(k)}$ in (13), as well as to set up the final data-driven second-order model $\widehat{H}(s) = \widehat{C}_{\mathrm{p}}(s^2\widehat{M} + s\widehat{E} + \widehat{K})^{-1}\widehat{B}_{\mathrm{u}}$ where

$$\begin{aligned} \widehat{M} &= \operatorname{diag}\left(\frac{1}{\omega_1^{(k)}}, \ldots, \frac{1}{\omega_r^{(k)}}\right), \\ \widehat{E} &= \operatorname{diag}(2\psi_1^{(k)}, \ldots, 2\psi_r^{(k)}), \\ \widehat{K} &= \operatorname{diag}(\omega_1^{(k)}, \ldots, \omega_r^{(k)}), \\ \widehat{B}_{\mathrm{u}} &= \begin{bmatrix} \phi_1^{\pm,(k)} & \cdots & \phi_r^{\pm,(k)} \end{bmatrix}^{\mathsf{T}} \quad \text{and} \quad \widehat{C}_{\mathrm{p}} = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}^{\mathsf{T}}. \end{aligned} \tag{15}$$

A brief sketch of the resulting second-order VF algorithm is given in Algorithm 2.

Remark 1. (Splitting of expansion points).
Another major difference to the classical VF is the splitting of the expansion points into two groups $\{\lambda_j^{+,(k)}\}_{j=1}^r$ and $\{\lambda_j^{-,(k)}\}_{j=1}^r$, related to each other by (10). For mechanical systems with real realizations, the splitting of complex points in conjugate pairs with positive imaginary parts ($\lambda_j^{+,(k)}$) and negative imaginary parts ($\lambda_j^{-,(k)}$) comes naturally. In the case of real expansion points, a physics-inspired splitting is with respect to bifurcation, i.e., with respect to a centered point on the real axis at which the real points would collide and split into complex conjugate pairs. For simplicity, we assume the real expansion points lie in the left open half-plane. Then, we sort the points such that those with largest magnitude ($\lambda_j^{-,(k)}$) are paired with those with smallest magnitude ($\lambda_j^{+,(k)}$).

Algorithm 2 Structured Vector Fitting – Version 1

Input: Vector h of data samples (5), initial guess for $\{\lambda_j^{+,(1)}\}_{j=1}^r$ and $\{\lambda_j^{-,(1)}\}_{j=1}^r$.
Output: Learned ROM $\widehat{H}(s) = \widehat{C}_{\mathrm{p}}(s^2\widehat{M} + s\widehat{E} + \widehat{K})^{-1}\widehat{B}_{\mathrm{u}}$.
1: Initialize $\Delta^{(1)} = I_\ell$ and $k = 1$.
2: **while** not converged **do**
3: Construct the coefficient matrix $A_{\mathrm{so1}}^{(k)}$ in (13).
4: Solve the weighted linear least-squares problem (14).
5: Update the expansion points $\{\lambda_j^{\pm,(k+1)}\}_{j=1}^r$ to be the eigenvalues of $\widehat{A}^{(k)} - \widehat{G}^{(k)}\widehat{C}^{(k)}$ using (7) and (8).
6: Update the weighting matrix $\Delta^{(k+1)}$ by (5).
7: Increment $k \leftarrow k + 1$.
8: **end while**
9: Set the final ROM matrices using (15).

3.2 Fully structured barycentric form

A second revised barycentric form for $\widehat{H}^{(k)}$ is obtained by replacing both the numerator and denominator by second-order-type pole-residue forms (9), i.e., we write $\widehat{H}^{(k)}$ as

$$\widehat{H}^{(k)}(s) = \frac{\sum_{j=1}^{r} \frac{\omega_j^{(k)}\phi_j^{\pm,(k)}}{(s-\lambda_j^{+,(k)})(s-\lambda_j^{-,(k)})}}{1 + \sum_{j=1}^{r} \frac{\omega_j^{(k)}\varphi_j^{\pm,(k)}}{(s-\lambda_j^{+,(k)})(s-\lambda_j^{-,(k)})}}. \tag{16}$$

As in Section 3.1, this new barycentric form changes the form of the weighted linear LS problem in the resulting structured VF algorithm. Using (16), we obtain

$$\min_{\tilde{x}^{(k)}} \|\Delta^{(k)}(A_{\mathrm{so2}}^{(k)}\tilde{x}^{(k)} - h)\|_2^2, \tag{17}$$

where the least-squares matrix $A_{\mathrm{so2}}^{(k)}$ is given in (20), and the weighting matrix and data samples are as in (5). Then the solution vector

$$\tilde{x}^{(k)} = \begin{bmatrix} \phi_1^{\pm,(k)} & \cdots & \phi_r^{\pm,(k)} & \varphi_1^{\pm,(k)} & \cdots & \varphi_r^{\pm,(k)} \end{bmatrix}^{\mathsf{T}}$$

yields the resulting second-order system as in (15). The splitting of the expansion points also works as in Remark 1. However, the updating step of the expansion points (Algorithm 2 Step 5) changes. The denominator of (16) corresponds to a second-order system rather then a first-order system. While it would be possible to also rewrite this second-order system in first-order form, the zeros of the denominator are actually given by the eigenvalues of the quadratic matrix pencil

$$\lambda^2\widehat{M}^{(k)} + \lambda\widehat{E}^{(k)} + \left(\widehat{K}^{(k)} + \widehat{G}_{\mathrm{u}}^{(k)}\widehat{C}_{\mathrm{p}}^{(k)}\right), \tag{18}$$

where $\widehat{M}^{(k)}$, $\widehat{E}^{(k)}$, $\widehat{K}^{(k)}$ and $\widehat{C}_{\mathrm{p}}^{(k)}$ are constructed as their final learned counterparts in (15), and

$$\widehat{G}^{(k)} = \begin{bmatrix} \varphi_1^{\pm,(k)} & \cdots & \varphi_r^{\pm,(k)} \end{bmatrix}^{\mathsf{T}}. \tag{19}$$

A brief sketch of the resulting method is given in Algorithm 3.

$$A_{\mathrm{so2}}^{(k)} = \begin{bmatrix} \frac{\omega_1^{(k)}}{(\xi_1-\lambda_1^{+,(k)})(\xi_1-\lambda_1^{-,(k)})} & \cdots & \frac{\omega_r^{(k)}}{(\xi_1-\lambda_r^{+,(k)})(\xi_1-\lambda_r^{-,(k)})} & \frac{-\omega_1^{(k)}H(\xi_1)}{(\xi_1-\lambda_1^{+,(k)})(\xi_1-\lambda_1^{-,(k)})} & \cdots & \frac{-\omega_r^{(k)}H(\xi_1)}{(\xi_1-\lambda_r^{+,(k)})(\xi_1-\lambda_r^{-,(k)})} \\ \vdots & & \vdots & \vdots & & \vdots \\ \frac{\omega_1^{(k)}}{(\xi_\ell-\lambda_1^{+,(k)})(\xi_\ell-\lambda_1^{-,(k)})} & \cdots & \frac{\omega_r^{(k)}}{(\xi_\ell-\lambda_r^{+,(k)})(\xi_\ell-\lambda_r^{-,(k)})} & \frac{-\omega_1^{(k)}H(\xi_\ell)}{(\xi_\ell-\lambda_1^{+,(k)})(\xi_\ell-\lambda_1^{-,(k)})} & \cdots & \frac{-\omega_r^{(k)}H(\xi_1)}{(\xi_\ell-\lambda_r^{+,(k)})(\xi_\ell-\lambda_r^{-,(k)})} \end{bmatrix} \tag{20}$$

Algorithm 3 Structured Vector Fitting – Version 2

Input: Vector h of data samples (5), initial guess for $\{\lambda_j^{+,(1)}\}_{j=1}^r$ and $\{\lambda_j^{-,(1)}\}_{j=1}^r$.

Output: Learned ROM $\widehat{H}(s) = \widehat{C}_{\mathrm{p}}(s^2\widehat{M} + s\widehat{E} + \widehat{K})^{-1}\widehat{B}_{\mathrm{u}}$.

1: Initialize $\Delta^{(1)} = I_\ell$ and $k = 1$.
2: **while** not converged **do**
3: Construct the coefficient matrix $A_{\mathrm{sol2}}^{(k)}$ in (20).
4: Solve the weighted linear least-squares problem (17).
5: Update the expansion points $\{\lambda_j^{\pm,(k+1)}\}_{j=1}^r$ to be the eigenvalues of (18) using (15) and (19).
6: Update the weighting matrix $\Delta^{(k+1)}$ by (5).
7: Increment $k \leftarrow k+1$.
8: **end while**
9: Set the final ROM matrices using (15).

We note that realness of the resulting state-space realization can be preserved similar to the classical VF; see Gustavsen and Semlyen (1999). Indeed, this task becomes simpler in case of Algorithm 3 due to the natural pairing of complex conjugate expansion points; cf. Remark 1.

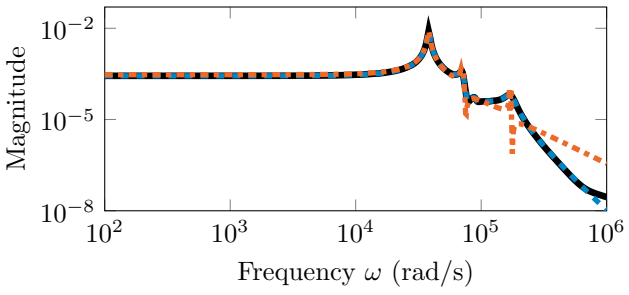
4. NUMERICAL EXAMPLES

We test the two proposed approaches on two benchmark problems:

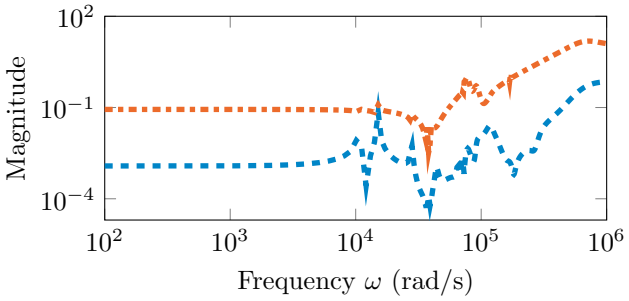
(i) the butterfly gyroscope example from the Oberwolfach Benchmark Collection (2004), and
(ii) the artificial fishtail model from Siebelts et al. (2019).

For simplicity, we consider here only single-input/single-output versions of these examples. While the outputs of the first model are summed together, only the second output entry of the second model is used. In both examples, we consider data sets with 1000 linearly equidistant sampling points on the positive imaginary axis. For the butterfly gyroscope example, the points lie in $[10^2, 10^6]$ rad/s, while for the artificial fishtail model, the points are in $[0, 1\,000]$ rad/s. Since the matrices of the original mechanical systems are real-valued, the data samples are closed under conjugation. This is done by additionally including the complex conjugate counterparts of both the evaluations and the sampling points into the data sets. Both proposed structured VF algorithms from Section 3 are applied to these two models. Thereby, we denote the approach from Algorithm 2 using the partially structured barycentric form by SOVF1 and the method in Algorithm 3 based on the fully structured barycentric form by SOVF2.

The experiments reported here have been executed on a machine equipped with an AMD Ryzen 5 5500U processor running at 2.10 GHz and with 16 GB total main memory. The computer runs on Windows 10 Home version 20H2 (build 19042.1237) and for the experiments we use MATLAB 9.9.0.1592791 (R2020b).



(a) Transfer functions.



(b) Pointwise relative errors.

Fig. 1. Results for the butterfly gyroscope data.

Code and data availability
The source code, authored by Steffen W. R. Werner, of the implementations used to compute the presented results, the used data and the computed results are available at

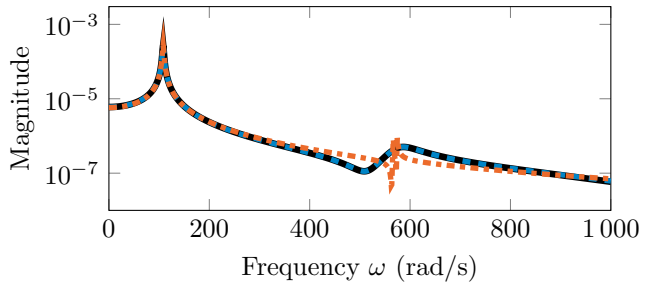
`doi:10.5281/zenodo.5539944`

under the BSD-2-Clause license.

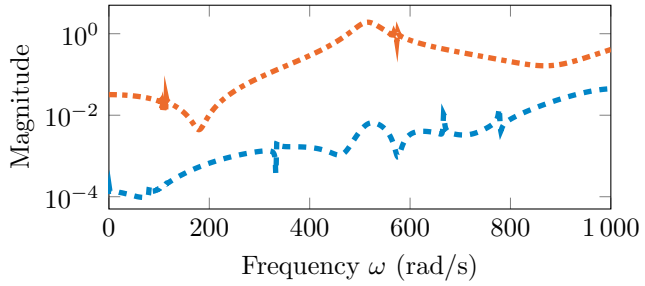
4.1 Butterfly gyroscope example

First, we present the results for the butterfly gyroscope model as shown in Figure 1. For the given data, we have used the two proposed approaches to learn structure-preserving models of order $r = 8$. While SOVF2 converges up to numerical accuracy, this is not the case for SOVF1. However, the denominator in SOVF1 converges reasonably close to the value one. Consequently, we have simply considered the mechanical system associated with the rational function in the numerator of (12) and ignored the denominator entry altogether.

As it can be easily observed in Figure 1a, SOVF1 accurately approximates the given data over the full frequency range. A minor exception is given by the right limit of the frequency interval, where the transfer function of SOVF1 slightly deviates from the data. On the other hand, SOVF2 lacks this good approximation behavior as it is illustrated



(a) Transfer functions.



(b) Pointwise relative errors.

Fig. 2. Results for the artificial fishtail data.

in Figure 1b. The accuracy of this approach is at least two orders of magnitude worse than that of SOVF1. Still, SOVF2 yields a reasonable approximation for the low frequency range. We have observed that SOVF2 typically introduces poles close to the imaginary axis, and has better approximation quality in the large magnitude range.

4.2 Artificial fishtail example

We now present results for the artificial fishtail example, as shown in Figure 2. We have used both approaches and constructed structure-preserving learned models of order $r = 10$. As in the first example presented in Section 4.1, SOVF1 provides a high-fidelity approximation over the full frequency interval and outperforms SOVF2. As shown in Figure 2b, the SOVF2 approximation has a large mismatch around 580 rad/s, where the relative approximation error is the highest. As in the previous example, SOVF2 provides an accurate approximation of the main dominant peak, i.e., the one located around 100 rad/s. Further details on the numerical results and convergence of the applied methods can be found in the accompanying code package.

5. CONCLUSIONS

We have proposed two new approaches for data-driven modeling of modally damped mechanical systems by developing structure-preserving vector fitting formulations. We have revised the barycentric formula to represent structured transfer functions, and have shown that the structure of the original model is automatically preserved in the reduced one. The two approaches have been applied to two benchmark models and the preliminary results are promising. The method corresponding to the partially structured transfer function formulation has been proven especially accurate and reliable in both test cases. A more thorough investigation is needed to explain the discrepancies in accuracy encountered in the SOVF2 formulation. Extending the analysis and numerical algorithms to MIMO problems and employing the proposed structured barycentric forms to develop a AAA-like framework are natural next steps.

REFERENCES

Antoulas, A.C. and Anderson, B.D.O. (1986). On the scalar rational interpolation problem. *IMA J. Math. Control Inf.*, 3(2–3), 61–8. doi:10.1093/imamci/3.2-3.61.

Beattie, C. and Benner, P. (2014). $\mathcal{H}_2$-optimality conditions for structured dynamical systems. Preprint MPIMD/14-18, Max Planck Institute for Dynamics of Complex Technical Systems Magdeburg.

Benner, P., Goyal, P., and Pontes Duff, I. (2020). Data-driven identification of Rayleigh-damped second-order systems. e-print 1910.00838, arXiv. doi:10.48550/arXiv.1910.00838. Optimization and Control (math.OC).

Berrut, J.P. and Trefethen, L.N. (2004). Barycentric Lagrange interpolation. *SIAM Rev.*, 46(3), 501–517. doi:10.1137/S0036144502417715.

Drmač, Z., Gugercin, S., and Beattie, C. (2015a). Quadrature-based vector fitting for discretized $\mathcal{H}_2$ approximation. *SIAM J. Sci. Comput.*, 37(2), A625–A652. doi:10.1137/140961511.

Drmač, Z., Gugercin, S., and Beattie, C. (2015b). Vector fitting for matrix-valued rational approximation. *SIAM J. Sci. Comput.*, 37(5), A2346–A2379. doi:10.1137/15M1010774.

Grivet-Talocia, S. and Gustavsen, B. (2015). *Passive Macromodeling: Theory and Applications*. Wiley Series in Microwave and Optical Engineering. John Wiley & Sons, Hoboken, NJ. doi:10.1002/9781119140931.

Gustavsen, B. and Semlyen, A. (1999). Rational approximation of frequency domain responses by vector fitting. *IEEE Trans. Power Del.*, 14(3), 1052–1061. doi:10.1109/61.772353.

Mayo, A.J. and Antoulas, A.C. (2007). A framework for the solution of the generalized realization problem. *Linear Algebra Appl.*, 425(2–3), 634–662. doi:10.1016/j.laa.2007.03.008. Special issue in honor of P. A. Fuhrmann, Edited by A. C. Antoulas, U. Helmke, J. Rosenthal, V. Vinnikov, and E. Zerz.

Nakatsukasa, Y., Sète, O., and Trefethen, L.N. (2018). The AAA algorithm for rational approximation. *SIAM J. Sci. Comput.*, 40(3), A1494–A1522. doi:10.1137/16M1106122.

Oberwolfach Benchmark Collection (2004). Butterfly gyroscope. hosted at MORwiki – Model Order Reduction Wiki. URL `http://modelreduction.org/index.php/Butterfly_Gyroscope`.

Rodriguez, A.C. (2020). *Approximation of Parametric Dynamical Systems*. Ph.D. thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, USA. URL `http://hdl.handle.net/10919/99895`.

Saak, J., Siebelts, D., and Werner, S.W.R. (2019). A comparison of second-order model order reduction methods for an artificial fishtail. *at-Automatisierungstechnik*, 67(8), 648–667. doi:10.1515/auto-2019-0027.

Sanathanan, C. and Koerner, J. (1963). Transfer function synthesis as a ratio of two complex polynomials. *IEEE Trans. Autom. Control*, 8(1), 56–58. doi:10.1109/TAC.1963.1105517.

Schulze, P., Unger, B., Beattie, C., and Gugercin, S. (2018). Data-driven structured realization. *Linear Algebra Appl.*, 537, 250–286. doi:10.1016/j.laa.2017.09.030.

Siebelts, D., Kater, A., Meurer, T., and Andrej, J. (2019). Matrices for an artificial fishtail. hosted at MORwiki – Model Order Reduction Wiki. doi:10.5281/zenodo.2558728.

Werner, S.W.R. (2021). *Structure-Preserving Model Reduction for Mechanical Systems*. Dissertation, Otto-von-Guericke-Universität, Magdeburg, Germany. doi:10.25673/38617.