Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 59-1 (2025) 139–144

Towards Automated Model Order Reduction and Feedback Control for Nonlinear Finite Element Models ⋆

Arwed Schütz * **Michael Olbrich** ** **Aliakbar Taghdiri** *
Christoph Ament ** **Tamara Bechtold** *

* *Jade University of Applied Sciences, 26389 Wilhelmshaven, Germany, (e-mail: {arwed.schuetz, tamara.bechtold}[at]jade-hs.de).*
** *Faculty of Applied Computer Science, University of Augsburg, 86159 Augsburg, Germany (e-mail: {michael.olbrich, christoph.ament}[at]uni-a.de)*

Abstract: Spanning applications from microsystems to passenger jets, finite element (FE) models play a crucial role in the design of a wide range of technical products. However, such models are not suited for applications at system-level or in control due to their large scale. While several schemes to obtain models and controllers based on FE models exist, they either require high-level access to the FE software, create black-box models without physical interpretation, or require specialist knowledge. This paper outlines a workflow to generate highly efficient models and appropriate controllers for nonlinear FE models at the push of a button. In a first step, model order reduction via proper orthogonal decomposition creates an accurate surrogate model of drastically smaller dimension. Nonlinearities are handled via the trajectory piecewise linear approximation (TPWL), maximizing compatibility with commercial FE software by exclusively relying on data produced by regular solutions. Complementing TPWL, gain-scheduling is deployed to establish a precise controller. The proposed workflow is demonstrated for a tunable prism, showcasing its efficacy.

Keywords: finite element method, nonlinear model order reduction, proper orthogonal decomposition, trajectory piecewise linear approximation, modeling, second-order systems

1. INTRODUCTION

Numerical models are a cornerstone of modern engineering, facilitating design optimization, reducing development costs, and minimizing material usage. One of the most prominent methods for such models is finite element (FE) analysis. FE models enable accurate simulations of various physical phenomena for any geometry and scale, ranging from microchips to cruise ships. Accurate predictions of a device's behavior render FE analysis indispensable, particularly when models are well-calibrated to experimental data. However, this level of accuracy comes at high computational costs as resulting models are commonly of high dimension. This major drawback prevents applications in, e.g., system-level simulation or feedback control.

An elegant method to bridge this gap is found in projection-based model order reduction (MOR) as comprehensively described by Antoulas (2005). Based on the original system, MOR constructs a low-dimensional yet highly accurate surrogate model, also referred to as a reduced order model (ROM). After its offline construction, the ROM provides significant speed-up during online deployment. MOR synergizes well with FE models, can be interpreted in physical terms, and maintains the same mathematical structure. However, as described by Baur et al. (2014), this methodology is well-established for linear systems, but requires additional methods to handle nonlinearities.

Such methods are referred to as hyperreduction. The state of the art is defined by the discrete empirical interpolation method by Chaturantabut and Sorensen (2010) or the energy conserving mesh sampling and weighting method by Farhat et al. (2014). These methods evaluate a small subset of the nonlinear terms for their approximation. Therefore, these methods require the specific nonlinear expressions. This poses a significant restriction for non-academic software as its solvers act as black boxes that can only be sampled. An alternative method exclusively built from easily accessible data is the trajectory piecewise linear approximation (TPWL) by Rewieński (2003). It approximates nonlinear terms with a weighted sum of their linearizations sampled around several states. TPWL and FE form a natural combination as nonlinear solvers are based on iterative linearizations. Therefore, all required data such as Jacobians and residuals are readily available as demonstrated in our previous work in Schütz et al. (2023). Note that TPWL's interpolation-based nature limits its approximation quality to trajectories in the vicinity of sampled data.

⋆ This research is part of the project Adaptive Optics for THz (ADOPT) within the priority program Cooperative Multistage Multistable Microactuator Systems (KOMMMA) and was funded by Deutsche Forschungsgemeinschaft (German Research Foundation) grant number 424616052.

10.1016/j.ifacol.2025.03.025

A control scheme well-suited for both automatic controller design and TPWL is found in gain-scheduling. As described by Leith and Leithead (2000), the key idea is to linearize a nonlinear model at several operating points and to design a linear controller for each of these models. These controllers are then interpolated during operation based on so-called scheduling variables such as outputs or states. Hence, the concepts of TPWL and gain-scheduling are strongly related and use the same data. Tonkens et al. (2021) have successfully demonstrated this combined methodology based on an FE model to control a soft robot. This paper introduces another numerical case study in form of a bi-axially actuated tunable prism for microscopy using an industry-standard FE tool.

The remainder of this paper is structured as follows: Section 2 introduces the tunable prism, which serves as a numerical case study for this work. Section 3 guides through the modeling process, including the FE model, nonlinear MOR via TPWL, and an evaluation of the ROM's performance. Section 4 describes the design and evaluation of a corresponding controller using gain-scheduling. Finally, Section 5 summarizes the work and suggests future research directions.

2. NUMERICAL CASE STUDY: PRISM ACTUATOR

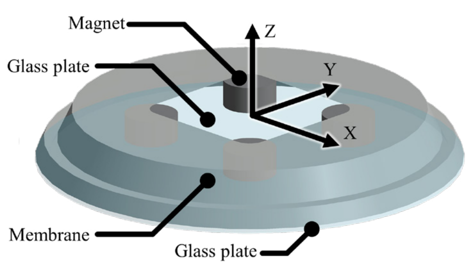

Fig. 1. The prism as proposed by Weber et al. (2021b). The relevant degrees of freedom (DOFs) and outputs of later models are the upper glass plate's vertical position and its rotation along the y-axis and x-axis.

Confocal microscopes offer excellent optical resolution. The overall image is assembled point by point, as only a small part of the sample is in focus. Therefore, the process deploys scanning patterns to capture the three-dimensional specimen. The required motion is conventionally achieved by translational positioning stages, movable lenses, mirrors, or some combination thereof. In addition, further elements might be necessary to correct imaging errors. Consequently, corresponding microscopes are bulky, susceptible to environmental influences, and limited by inertia, potentially introducing motion artifacts.

A promising alternative are bi-axial tunable prisms as proposed by Lemke et al. (2019), leading to higher resolution, faster scans, more compact designs, and increased robustness due to less moving components. Recently, Weber et al. (2021a,b) presented a novel design relying on magnetic actuation instead of brittle piezoelectric elements, further improving robustness. In addition, the low-cost design features inherent self-sensing capabilities as studied by Weber et al. (2023). Fig. 1 shows the prism's composition, which is available in more detail in the original work by Weber et al. (2021b). A rubber membrane mounted on a circular glass substrate is sealed with a glass window on top. The paraffin oil inside controls optical properties. Four magnets attached to the glass window interact with coils below (not shown) for vertical motion and tilting.

3. MATHEMATICAL MODELING

The modeling process consists of three stages and an evaluation: in Subsection 3.1, an FE model establishes the reference solution and creates data for subsequent steps. Subsection 3.2 introduces projection-based MOR. Finally, Subsection 3.3 adds TPWL to achieve efficient hyperreduction, resulting in the final ROM. Subsection 3.4 evaluates the ROM's performance in a representative load case with respect to the original FE model.

All FE analyses are conducted using Ansys® Academic Research Mechanical, Release 2022 R2. MOR and TPWL use Model Reduction inside Ansys and Python, particularly NumPy and SciPy.

3.1 Finite Element Model

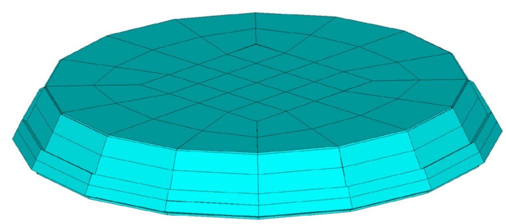

Fig. 2. FE mesh of the simplified prism. The geometric discretization error is lower than implied as the quadratic elements are displayed as flat quadrilaterals.

The design in Fig. 1 is slightly simplified to enable an efficient FE representation as shown in Fig. 2. A total of 160 quadratic shell elements represents the basic structure of membrane and glass. Four point masses capture the magnets' inertias. Designated fluid elements incorporate the effects of the incompressible parrafin oil inside. This efficient representation results in an FE model of only 1543 DOFs. The low order proves particularly useful for developing numerical workflows, as small file sizes and low-dimensional calculations provide almost instant feedback. Material properties are simplified to be linear, but large deformations and the incompressible fluid are considered, rendering the analysis nonlinear. The electromagnetic actuation is decoupled from this model. Instead, the relation between force, magnet position, and coil current is characterized by additional FE analyses not presented here.

Mathematically, the FE model corresponds to a large-scale system of $n = 1543$ nonlinear second-order ordinary differential equations

$$\Sigma = \begin{cases} \boldsymbol{M}(\boldsymbol{x})\,\ddot{\boldsymbol{x}} + \boldsymbol{E}(\boldsymbol{x})\,\dot{\boldsymbol{x}} + \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{B}\,\boldsymbol{u} \\ \boldsymbol{y} = \boldsymbol{C}\,\boldsymbol{x} \end{cases}, \quad (1)$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector and $\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^n$ are the nonlinear restoring forces. The state-dependent matrices for inertia and damping are denoted as $\boldsymbol{M}(\boldsymbol{x})$ and $\boldsymbol{E}(\boldsymbol{x}) \in \mathbb{R}^{n\times n}$, respectively. The output matrix $\boldsymbol{C} \in \mathbb{R}^{q\times n}$ assembles the user-defined outputs $\boldsymbol{y} \in \mathbb{R}^q$.

To capture the prism's behavior, $q = 3$ outputs are chosen, comprising the glass plate's vertical position and its rotations along the two horizontal axes. Note that these quantities can be converted to polar angle and azimuth angle or the four magnets' vertical positions, e.g., for a seamless connection with magnetic force characteristics. The input matrix $\boldsymbol{B} \in \mathbb{R}^{n\times p}$ distributes the p inputs in $\boldsymbol{u}$. These inputs are determined by currents of the four coils, but are implemented as a vertical force and tilting moments acting on the glass plate. Conversions similar to the outputs are possible.

3.2 Model Order Reduction

Projection-based MOR aims to produce a low-dimensional model that retains a high degree of accuracy. The key idea is to identify a subspace of significantly smaller dimension that captures most of the original state's dynamics. Using a projection matrix $\boldsymbol{V} \in \mathbb{R}^{n\times r}$ which columns span this subspace, the original state $\boldsymbol{x}$ can be approximated as

$$\boldsymbol{x} \approx \boldsymbol{V}\,\boldsymbol{x}_r\,, \tag{2}$$

where $\boldsymbol{x}_r \in \mathbb{R}^r$ is the reduced state vector. The most prominent method to find an appropriate subspace for nonlinear models is the proper orthogonal decomposition (POD). First, the original system is sampled, i.e., simulated for anticipated trajectories. Corresponding solutions for the state vector are referred to as snapshots and are collected as columns of a snapshot matrix. A singular value decomposition then reveals an appropriate subspace in form of left singular vectors. Reducing the original model's order additionally requires projecting the system onto the same subspace. These two steps lead to a ROM given by

$$\Sigma_r = \begin{cases} \boldsymbol{V}^\top \boldsymbol{M}\left(\boldsymbol{V}\,\boldsymbol{x}_r\right)\ddot{\boldsymbol{x}}_r + \boldsymbol{V}^\top \boldsymbol{E}\left(\boldsymbol{V}\,\boldsymbol{x}_r\right)\dot{\boldsymbol{x}}_r \\ \qquad\qquad + \boldsymbol{V}^\top \boldsymbol{f}\left(\boldsymbol{V}\,\boldsymbol{x}_r\right) = \boldsymbol{B}_r\,\boldsymbol{u}\,. \\ \boldsymbol{y} = \boldsymbol{C}_r\,\boldsymbol{x}_r \end{cases} \tag{3}$$

All matrices with subscript r indicate reduced versions of their high-dimensional counterparts and summarize products of projection matrices and original system matrices. Exceptions are input and output, which remain unchanged. However, the nonlinearities gained complexity: evaluating nonlinear terms at reduced level requires expanding the reduced state $\boldsymbol{x}_r$ back to its original dimension, evaluating the nonlinearity, and finally, projecting it back to the reduced space. Methods to evaluate such terms more efficiently are referred to as hyperreduction, which is achieved by TPWL in the next subsection.

3.3 Trajectory Piecewise Linear Approximation

The basic idea is to approximate nonlinear terms by weighted sums of their respective linearizations. Hence, the nonlinearity is linearized around several states, typically along representative trajectories. These linearizations can be sampled on the fly, as suggested by Rewieński (2003), or by pruning an extensive set of sampled data, as described by Tiwary and Rutenbar (2005). A weighted summation of these linearized quantities approximates the global nonlinearity. Finally, this approximation is incorporated into the dynamical system. The weights are typically based on the (reduced) state to schedule linearizations. Mohseni et al. (2016) demonstrate alternatives such as technically meaningful outputs. Further note that TPWL is commonly combined with MOR, but may also be deployed on its own. Advantages of TPWL include its robustness and its non-intrusive nature, i.e., the fact that only easily obtainable data rather than detailed analytic descriptions are required. Furthermore, the initial set of linearizations can be incrementally updated with new data to some extent without the need to repeat the whole workflow. However, its approximation quality tends to degrade when trajectories deviate significantly from sampled ones. Depending on the choice of interpolation scheme and sampling density, rapid switching between different linearized models might cause chattering model selection during simulation.

In mathematical terms, TPWL first linearizes nonlinearities around N states $\boldsymbol{x}_i$ from the system's trajectory as

$$\begin{aligned} \boldsymbol{f}\left(\boldsymbol{x}\right)\big|_{\boldsymbol{x}_i} &\approx \underbrace{\boldsymbol{f}\left(\boldsymbol{x}\right)\big|_{\boldsymbol{x}_i}}_{\boldsymbol{f}_i} + \underbrace{\frac{\partial \boldsymbol{f}\left(\boldsymbol{x}\right)}{\partial \boldsymbol{x}}\bigg|_{\boldsymbol{x}_i}}_{\boldsymbol{K}_i}\left(\boldsymbol{x} - \boldsymbol{x}_i\right) \\ &= \underbrace{\boldsymbol{f}_i - \boldsymbol{K}_i\,\boldsymbol{x}_i}_{\hat{\boldsymbol{f}}_i} + \boldsymbol{K}_i\,\boldsymbol{x}\,, \end{aligned} \tag{4}$$

where $\boldsymbol{f}_i \in \mathbb{R}^n$ is the nonlinear function evaluated at $\boldsymbol{x}_i$ and $\boldsymbol{K}_i \in \mathbb{R}^{n\times n}$ its Jacobian, i.e., a constant force vector and a stiffness matrix. A weighted sum of these linearizations forms the global approximation

$$\boldsymbol{f}\left(\boldsymbol{x}\right) \approx \sum_{i=1}^{N} w_i(\boldsymbol{x})\;\hat{\boldsymbol{f}}_i + \sum_{i=1}^{N} w_i(\boldsymbol{x})\;\boldsymbol{K}_i\;\boldsymbol{x}\,. \tag{5}$$

The state-dependent matrices $\boldsymbol{M}\left(\boldsymbol{x}\right)$ and $\boldsymbol{E}\left(\boldsymbol{x}\right)$ are approximated similarly but without derivative information, i.e., as a weighted sum of sampled values. The weights $w_i(\boldsymbol{x})$ are computed according to Algorithm 1 in Appendix A. As the state-dependent weights are dynamically evaluated during simulation, the TPWL-approximated model remains nonlinear. Denoting the weights as w_i for readability and substituting (5) into (1) results in

$$\begin{aligned} &\sum_{i=1}^{N}\Big(w_i\,\boldsymbol{M}_i\Big)\,\ddot{\boldsymbol{x}} + \sum_{i=1}^{N}\Big(w_i\,\boldsymbol{E}_i\Big)\,\dot{\boldsymbol{x}} \\ &\qquad + \sum_{i=1}^{N}\Big(w_i\,\boldsymbol{K}_i\Big)\,\boldsymbol{x} + \sum_{i=1}^{N}\Big(w_i\,\hat{\boldsymbol{f}}_i\Big) = \boldsymbol{B}\,\boldsymbol{u}\,. \end{aligned} \tag{6}$$

All components are compatible to projection-based MOR and can be reduced using the same global projection for all systems leading to the ROM

$$\begin{aligned} &\sum_{i=1}^{N}\Big(w_i\,\boldsymbol{M}_{i,r}\Big)\,\ddot{\boldsymbol{x}}_r + \sum_{i=1}^{N}\Big(w_i\,\boldsymbol{E}_{i,r}\Big)\,\dot{\boldsymbol{x}}_r \\ &\qquad + \sum_{i=1}^{N}\Big(w_i\,\boldsymbol{K}_{i,r}\Big)\,\boldsymbol{x}_r + \sum_{i=1}^{N}\Big(w_i\,\hat{\boldsymbol{f}}_{i,r}\Big) = \boldsymbol{B}_r\,\boldsymbol{u}, \end{aligned} \tag{7}$$

which is deployed in this form. The weights w_i for a ROM are commonly based on the reduced state $\boldsymbol{x}_r$, but this work relies on the output $\boldsymbol{y}$ instead. All these steps to generate the ROM and their associated computational costs are offline, i.e., before online deployment in an application. Please note that the linearized systems might be reduced with individual projections instead of a global one as proposed by Lohmann and Eid (2009) to decrease the overall order.

3.4 Reduced Order Model Evaluation

The original FE model is simulated to generate data for MOR and TPWL. Static analyses suffice for sampling as all nonlinearities only depend on the state and not on its derivatives. Compared to transient analyses, static analyses are associated with less computational costs and avoid building up numerical error. Inspired by the original work by Weber et al. (2021b), samples are chosen to capture typical system states, i.e., to cover the glass plate tilted in several angles to all sides. This grid of polar angle and azimuthal angle is sampled for $1\,^\circ$ - $5\,^\circ$ in steps of $1\,^\circ$ and a full revolution in steps of $45\,^\circ$, respectively. The forces and torques required for these orientations are determined by a set of additional displacement-driven FE analyses. Together with the prism's resting position, a total of 41 static loadcases is analyzed, resulting in the same number of sampled linearized systems for TPWL. Each loadcase comprises 10 substeps with ramped loads, creating 410 samples for the state vector and constitute the snapshot matrix for POD. A singular value decomposition identifies dominant patterns ranked by their respective singular value. ROMs of dimensions $5 - 25$ have been created for later studies in Subsection 4.2.

The ROM of dimension 20 is compared against the FE reference model with a transient analysis covering the sampled range. A spiral motion with a duration of 0.5 s is analyzed in time steps of 0.5 ms with the input

$$\boldsymbol{u}(t) = \begin{pmatrix} F_z(t) \\ M_x(t) \\ M_y(t) \end{pmatrix} = \begin{pmatrix} 0 \\ \hat{M} \cdot \sin(\omega \cdot t) \cdot t \\ \hat{M} \cdot \cos(\omega \cdot t) \cdot t \end{pmatrix}, \tag{8}$$

where the torque has an amplitude of $\hat{M} = 15\,\mathrm{N} \cdot \mathrm{mm}$ and an angular frequency of $\omega = 2\pi/0.05\,\mathrm{s}^{-1}$. Fig. 3 presents the results for the two angular outputs computed by both models. The plot additionally provides the absolute error, which is well-suited to assess approximation quality despite numerous zeros. In general, the ROM provides accurate results as its error does not surpass $0.075\,^\circ$. However, the error accumulates over time. In terms of computational effort, the ROM solves in 3.77 s compared to the original model's 549.48 s, achieving a speed-up of more than two orders of magnitude.

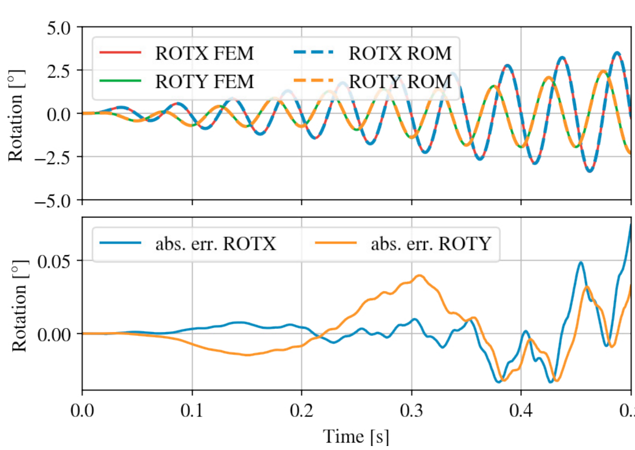

Fig. 3. Result comparison and absolute error between the reference FE model and the ROM using TPWL. Rotation axes are indicated in Fig. 1.

4. TPWL-BASED CONTROL

The composition of the ROM from linear subsystems offers the advantage of a feedback controller design using linear control theory. In the following, we describe a control strategy that can largely be automated. Its suitability is then shown in a simulative study.

4.1 Control Design

Similar to the approximation of the nonlinear dynamics by a set of linearized subsystems, we can design controllers for each of these individual models. The system input then results from the interpolation of the respective controller outputs according to Algorithm 1, i.e., in the same way as used for the simulation. This approach is known as gain-scheduling.

For this purpose, we extend the second-order dynamics of each linearized system in (7) to a system of first order differential equations with the extended state vector $\boldsymbol{\xi} = [\boldsymbol{x}_\mathrm{r}^\top, \dot{\boldsymbol{x}}_\mathrm{r}^\top]^\top$. The i-th state space equation is then given by

$$\dot{\boldsymbol{\xi}} = \boldsymbol{F}_i\,\boldsymbol{\xi} + \boldsymbol{G}_i\,\boldsymbol{u} + \boldsymbol{h}_i \tag{9}$$

with the matrices

$$\boldsymbol{F}_i = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{M}_{i,\mathrm{r}}^{-1}\boldsymbol{K}_{i,\mathrm{r}} & -\boldsymbol{M}_{i,\mathrm{r}}^{-1}\boldsymbol{E}_{i,\mathrm{r}} \end{bmatrix}, \quad \boldsymbol{G}_i = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{M}_{i,\mathrm{r}}^{-1}\boldsymbol{B}_\mathrm{r} \end{bmatrix},$$
$$\boldsymbol{h}_i = \begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{M}_{i,\mathrm{r}}^{-1}\boldsymbol{f}_{i,\mathrm{r}} \end{bmatrix}. \tag{10}$$

A suitable control strategy for systems with multiple inputs and outputs is given by a state feedback controller with control law

$$\boldsymbol{u}_i = \boldsymbol{K}_i\,(\boldsymbol{\xi}_\mathrm{ref} - \boldsymbol{\xi}), \tag{11}$$

with controller gain matrix $\boldsymbol{K}_i$ and reference state $\boldsymbol{\xi}_\mathrm{ref}$. For each of the linearized systems of the TPWL-approximated model, the gain matrix can be obtained offline by the linear quadratic regulator (LQR) approach, which minimizes the cost function

$$J = \int_0^\infty (\boldsymbol{\xi}_\mathrm{ref} - \boldsymbol{\xi})^\top \boldsymbol{Q}\,(\boldsymbol{\xi}_\mathrm{ref} - \boldsymbol{\xi}) + \boldsymbol{u}_i^\top \boldsymbol{R}\,\boldsymbol{u}_i\;\mathrm{d}t\,. \tag{12}$$

The weighting matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ penalize state deviations and the input effort, respectively, and can be chosen depending on the requirements. In this case, the control goal is defined by the reference state instead of the output reference $\boldsymbol{y}_\mathrm{ref}$. This can pose a challenge, since in general the full state leading to the desired output is not known exactly. Tonkens et al. (2021) solve this issue by optimal trajectory planning, in which the input and state trajectories are computed by minimizing a cost function over a predefined prediction horizon. While the obtained state trajectory is used as reference $\boldsymbol{\xi}_\mathrm{ref}$, the optimal input is additionally applied in a feedforward manner.

In this work, we propose an alternative method, which only requires the desired output trajectory. The control law is chosen as

$$\boldsymbol{u}_i = \boldsymbol{S}_i\,\boldsymbol{y}_\mathrm{ref} - \boldsymbol{K}_i\,\boldsymbol{\xi} + \boldsymbol{g}_i\,, \tag{13}$$

with the additional static prefilter matrix $\boldsymbol{S}_i$ and a compensation term $\boldsymbol{g}_i$ for the affine part $\boldsymbol{h}_i$ within the dynamics (9). These are designed to improve steady-state accuracy, i.e., to ensure the condition

$$\lim_{t\to\infty} \boldsymbol{y}(t) = \boldsymbol{y}_\mathrm{ref}\,. \tag{14}$$

To obtain the prefilter and compensation terms, we insert control law (13) into the model equations (9), resulting in

$$\dot{\boldsymbol{\xi}} = (\boldsymbol{F}_i - \boldsymbol{G}_i\boldsymbol{K}_i)\boldsymbol{\xi} + \boldsymbol{G}_i\boldsymbol{S}_i\boldsymbol{y}_{\mathrm{ref}} + \boldsymbol{G}_i\boldsymbol{g}_i + \boldsymbol{h}_i\,. \tag{15}$$

Since we are interested in the steady state, we set (15) to zero, solve for $\boldsymbol{\xi}$, and calculate the steady-state output, which is then given by

$$\begin{aligned}\lim_{t\to\infty}\boldsymbol{y} &= \boldsymbol{C}\boldsymbol{\xi}\big|_{\dot{\boldsymbol{\xi}}\to 0} \qquad (16)\\ &= -\boldsymbol{C}(\boldsymbol{F}_i - \boldsymbol{G}_i\boldsymbol{K}_i)^{-1}\left(\boldsymbol{G}_i\boldsymbol{S}_i\boldsymbol{y}_{\mathrm{ref}} + \boldsymbol{G}_i\boldsymbol{g}_i + \boldsymbol{h}_i\right)\,.\end{aligned}$$

To fulfill the steady-state condition (14), we can choose $\boldsymbol{S}_i$ and $\boldsymbol{g}_i$ as:

$$\boldsymbol{S}_i = -\left(\boldsymbol{C}(\boldsymbol{F}_i - \boldsymbol{G}_i\boldsymbol{K}_i)^{-1}\boldsymbol{G}_i\right)^{-1} \tag{17}$$

$$\boldsymbol{g}_i = -\boldsymbol{G}_i^{+}\,\boldsymbol{h}_i\,. \tag{18}$$

Here, the superscript $(\cdot)^{+}$ denotes the pseudo inverse. The matrices $\boldsymbol{S}_i$, $\boldsymbol{K}_i$ and the vector $\boldsymbol{g}_i$ are computed offline based on the linearized models. The overall control output can then be computed online as the weighted sum of all individual outputs

$$\boldsymbol{u} = \sum_{i=1}^{N} w_i\,\boldsymbol{u}_i\,. \tag{19}$$

Note that this corresponds to an interpolation of the individual controller outputs and its smoothness depends on how the weighting coefficients w_i are calculated. The proposed strategy corresponds to a state-feedback controller for which the reduced system states need to be available. In general, however, these cannot be measured. Instead, the states need to be estimated online from the output trajectory. For this, we use the state estimator given in Tonkens et al. (2021), which corresponds to weighted linear Luenberger observers for each submodel.

4.2 Controller Evaluation

The order and accuracy of the ROM are critical for successful feedback control. On the one hand, the ROM must be sufficiently small to allow its evaluation within the sampling time. On the other hand, its accuracy influences the control quality, and a system that is controlled based on an inaccurate model may even become unstable. In this section, we therefore evaluate the controller performance in simulation depending on the dimension of the ROM.

As control goal, we define a reference trajectory $\boldsymbol{y}_{\mathrm{ref}}$ that covers a large part of the working area. Here, we concentrate on the rotational degrees of freedom, for which a spiral-shaped motion is chosen, while the vertical displacement should remain at its initial zero-position.

The controller performance is evaluated in terms of the tracking control error regarding the rotation angles. The root mean squared error (RMSE) of the deviation between the controlled motion and the reference as well as the average time needed for a single evaluation of the controller depending on the ROM dimension are illustrated in Fig. 4. For a better interpretation of the results, the time sequences of the marked dimensions are visualized in Fig. 5. Note that for the simulations we used the ROM of dimension 25 as approximation of the original system to be controlled, and the lower dimensional ROMs were used for the controller design.

In this case, the ROM requires at least dimension 10 to achieve stable feedback control. Moreover, sufficiently small control errors require dimension 13 and higher. It should be mentioned, however, that the suitability of the ROM's dimension also depends on the chosen control approach. Lower dimensions may thus still be appropriate to control the prism actuator in case of alternative control methods such as robust control.

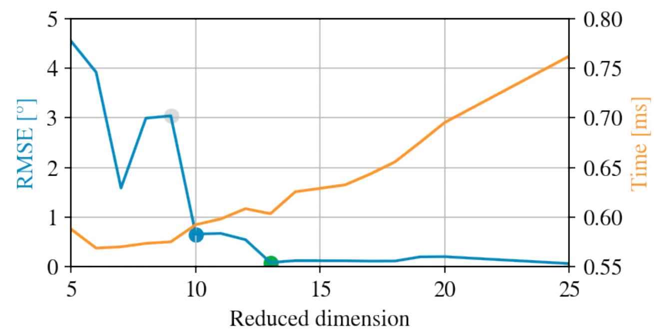

Fig. 4. Tracking control RMSE and average time needed for a single controller evaluation, both vs. dimension of the ROM the controller was designed with.

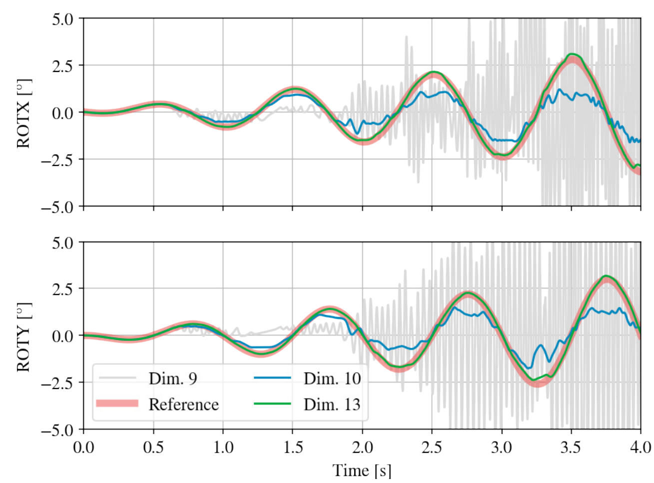

Fig. 5. Reference trajectory and results for the ROMs marked in Fig. 4.

5. CONCLUSION

This paper has presented a convenient workflow to derive low-dimensional yet accurate surrogate models from nonlinear FE models and to design appropriate controllers. All required data are easily obtainable from the FE solution process, even with commercial FE software. Creating a model does not require specialist knowledge or high-level access to the FE code. Designing a corresponding controller based on gain-scheduling preserves the ease of use and the potential for automatization. A numerical study has demonstrated the quality of controllers obtained from ROMs applied to a higher-dimensional model.

Aspects for future work are refining TPWL in terms of more efficient sampling, pruning sampled data, indicating range of validity, and investigating weighting schemes. To ensure precise control, approaches focused on robustness yield mentionable potential. In addition, experimental work to deploy the control scheme on hardware is planned.

DATA AVAILABILITY

Part of the code allowing to reproduce Fig. 3 is provided on GitLab at `https://gitlab.gwdg.de/jade-hochschule/fms/2025-mathmod`. Corresponding files include the ANSYS APDL script to create the FE model and to simulate the load case in (8). Furthermore, all files for TPWL-approximated ROMs of dimension 5, 10, and 20 as well as the python script for their simulation are supplied. Further data and code is provided upon reasonable request.

ACKNOWLEDGEMENTS

We thank our project partners Prof. Dr.-Ing. Ulrike Wallrabe and Pascal Weber from the Department of Microsystems Engineering (IMTEK) at the University of Freiburg for their collaboration, providing the case study of this work.

REFERENCES

Antoulas, A.C. (2005). *Approximation of large-scale dynamical systems*. Advances in design and control. Society for Industrial and Applied Mathematics, Philadelphia.

Baur, U., Benner, P., and Feng, L. (2014). Model Order Reduction for Linear and Nonlinear Systems: A System-Theoretic Perspective. *Archives of Computational Methods in Engineering*, 21(4), 331–358.

Chaturantabut, S. and Sorensen, D.C. (2010). Nonlinear Model Reduction via Discrete Empirical Interpolation. *SIAM Journal on Scientific Computing*, 32(5), 2737–2764.

Farhat, C., Avery, P., Chapman, T., and Cortial, J. (2014). Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency. *International Journal for Numerical Methods in Engineering*, 98(9), 625–662.

Leith, D.J. and Leithead, W.E. (2000). Survey of gain-scheduling analysis and design. *International Journal of Control*, 73(11), 1001–1025.

Lemke, F., Weber, P.M., Wallrabe, U., and Wapler, M.C. (2019). Piezo-Actuated Adaptive Prisms for Optical Scanning. In A. Martínez-García, I. Bhattacharya, Y. Otani, and R. Tutsch (eds.), *Progress in Optomechatronic Technologies*, volume 233 of *Springer Proceedings in Physics*, 85–91. Springer Singapore, Singapore.

Lohmann, B. and Eid, R. (2009). Efficient order reduction of parametric and nonlinear models by superposition of locally reduced models. In *Methoden und Anwendungen der Regelungstechnik*, Berichte aus der Steuerungs- und Regelungstechnik. Shaker, Aachen.

Mohseni, S.S., Yazdanpanah, M.J., and Ranjbar N., A. (2016). New strategies in model order reduction of trajectory piecewise–linear models. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 29(4), 707–725.

Rewieński, M. (2003). *A Trajectory Piecewise-Linear Approach to Model Order Reduction of Nonlinear Dynamical Systems*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA.

Schütz, A., Maeter, S., and Bechtold, T. (2023). Nonlinear Model Order Reduction of a MEMS Actuator by a Trajectory Piecewise-Linear Approximation. In *MikroSystemTechnik Kongress*, 551–554. VDE Verlag GmbH, Dresden.

Tiwary, S.K. and Rutenbar, R.A. (2005). Scalable trajectory methods for on-demand analog macromodel extraction. In W.H. Joyner, G.E. Martin, and A.B. Kahng (eds.), *Proceedings of the 42nd annual conference on Design automation - DAC '05*, 403. ACM Press.

Tonkens, S., Lorenzetti, J., and Pavone, M. (2021). Soft Robot Optimal Control Via Reduced Order Finite Element Models. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 12010–12016. IEEE.

Weber, P.M., Gowda, H.G., Wallrabe, U., and Wapler, M.C. (2021a). All in one: Tunable prism and lens in a single element. In *Proceedings of the 2021 International Symposium on Optomechatronic Technology (ISOT)*.

Weber, P.M., Wallrabe, U., and Wapler, M.C. (2023). Self-Sensing of a Magnetically Actuated Prism. *Sensors (Basel, Switzerland)*, 23(12).

Weber, P.M., Wapler, M.C., Lemke, F., and Wallrabe, U. (2021b). Bi-Axial Magnetically Actuated Tunable Prism. In *2021 21st International Conference on Solid-State Sensors, Actuators and Microsystems (Transducers)*, 431–434. IEEE.

Appendix A. TPWL WEIGHTING SCHEME

The weighting procedure for TPWL is given in Algorithm 1. It determines the weighting coefficients w_i for a linear combination of sampled quantities. Depending on whether TPWL is applied to a ROM or an unreduced model, the weight computation uses the reduced state $\boldsymbol{x}_r$ or the original state $\boldsymbol{x}$, respectively. This work deploys an alternative and relies on weights based on the output $\boldsymbol{y}$. The weights take values between zero and one and thus, samples are interpolated to obtain the quantity of interest. The weighting scheme basically corresponds to nearest-neighbor interpolation with continuous but sharp transitions. This approximately piecewise linear procedure is the source of the method's name. The parameter β defines the sharpness of transition and is set to $\beta = 25$ as suggested in the original work by Rewieński (2003). A vanishingly small numerical offset ε avoids division by zero.

Algorithm 1 Weighting scheme for TPWL.

Input: $\boldsymbol{x}_r$
Output: $w_{1,\dots,N}$
  for $i = 1, \dots, N$ **do**
    $d_i = \|\boldsymbol{x}_r - \boldsymbol{x}_{r,i}\|$
  $m = \min_{i=1,..,N} d_i + \varepsilon$
  for $i = 1, \dots, N$ **do**
    $\hat{w}_i = \exp(-\beta\, d_i/m)$
  $S = \sum_i^N \hat{w}_i$
  for $i = 1, \dots, N$ **do**
    $w_i = \hat{w}_i/S$