Pole-Matching Technique for Parametric Model Order Reduction of Second Order Vibroacoustic Systems

Yannik Hüpel[1], Harikrishnan K. Sreekumar[1], Sabine C. Langer[1]
[1] TU Braunschweig, Institut für Akustik, 38106 Braunschweig, Germany, Email: y.huepel@tu-braunschweig.de

Introduction

The steady advancements in computer technology as well as modeling techniques allow simulation models often used in engineering applications, such as design and optimization, to become more elaborated. The increase in complexity also yields a significant computational effort. These so-called large-scale models are needed for an accurate prediction of a system's behavior, however, the entailing computational effort makes an evaluation strenuous.

Large-scale models are often also the result of the modeling process in vibroacoustics, high frequency ranges using discretization methods like the Finite Element Method (FEM). The FEM yields system matrices, which sufficiently describe the model. Furthermore, for most applications it is feasible to compute the input-output behavior of the system, the so-called transfer function. For dynamic second order systems, such as vibroacoustic models, the transfer function can be expressed as follows

$$H(s) = \mathbf{C}(s^2\mathbf{M} + s\mathbf{D} + \mathbf{K})^{-1}\mathbf{B}. \tag{1}$$

Here $\mathbf{B}$ and $\mathbf{C}$ denote the input and output matrix respectively, while $\mathbf{K}$, $\mathbf{D}$, and $\mathbf{M}$ depict the system matrices. The Laplacian frequency parameter is denoted by $s = i\omega$, with ω being the angular frequency.

However as previously mentioned, the computation of the system's behavior is entailed by large computational effort. In order to reduce computational effort, the large-scale system matrices can be reduced by classical Model Order Reduction (MOR) methods. For this contribution we utilized a projection based Krylov method, where moment-matching is used to project the Full Order Model (FOM) onto a subspace of lower order, subsequently creating the frequency Reduced Order Model (fROM) spanned by the Krylov subspace. This leads to fROMs yielding a feasible prediction accuracy in the frequency domain, while also decreasing the computational effort for evaluating the examined model.

However, fast parameter changes express the need to compute new fROMs at the respective parameter values. Therefore, several fROMs accurate in the frequency domain are built, which can be interpolated across the examined parameter space. However, in [1] it is argued that an interpolation of the reduced system matrices does not yield satisfying results. Therefore, it is proposed in [1] to interpolate the so-called Pole-Residue-Depiction (PRD) of the fROMs. The PRD is based on a different expression of Equation 1, since the transfer function of a system can also be computed by the poles (λ_j) and residues (R_j) of a system [1]

$$H(s) = \sum_{j=1}^{n_d} \frac{R_j}{s - \lambda_j}, \tag{2}$$

where n_d denotes the number of poles. Since every differential equation can be transformed into an equivalent system of first order, in [1] the PRD is derived for first order systems. The most important approach is the similarity transformation of the main system matrix $\mathbf{A}$ [1]

$$\mathbf{A} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1}, \tag{3}$$

where $\mathbf{P}$ is a matrix consisting of the eigenvectors. The matrix $\mathbf{\Lambda}$ is diagonal with the eigenvalues as entries. Equation 3 can be used as input for the transfer function depicted in Equation 2 [1]

$$\begin{aligned} \mathbf{H}_{num} &= (\mathbf{C}^I_{j,1}\mathbf{B}^I_{j,1} + \mathbf{C}^I_{j,2}\mathbf{B}^I_{j,2})(s - a_j) \\ &\quad -(\mathbf{C}^I_{j,2}\mathbf{B}^I_{j,1} - \mathbf{C}^I_{j,1}\mathbf{B}^I_{j,2})b_j, \\ \mathbf{H} &= \sum_{j=1}^{n_d} \frac{\mathbf{H}_{num}}{(s - a_j)^2 + b_j^2}, \end{aligned} \tag{4}$$

where a_j and b_j are the real and imaginary part of the respective poles in Λ. Equation 4 allows for the computation of the model's PRD, where each row represents one specific pole and the respective residue thereof [1]

$$\mathrm{ROM} = \begin{bmatrix} a_1 & b_1 & C^{II}_{1,1} & C^{II}_{1,2} \\ \vdots & \vdots & \vdots & \vdots \\ a_m & b_m & C^{II}_{m,1} & C^{II}_{m,2} \end{bmatrix}. \tag{5}$$

Here $C^{II}_{j,i}$ denotes the two respective summands of $\mathbf{H}_{num}$. The PRD depicted in Equation 5 allows for the application of pole-matching techniques. We refer to [1] for a more detailed derivation of the PRD. The new expression of the model allows for an interpolation of the poles and residues of different fROMs expressed in PRD at different points in the examined parameter space, so as to yield the model's behavior for any desired parameter combination. Nonetheless, because of properties of the FEM and MOR algorithms, the poles belonging to the same parametric pole in different PRDs might not always be in the same respective row. However, for an accurate interpolation the respective pole positions should match, which is why a pole-matching technique is employed after the PRDs are built. The pole-matching rearranges the rows of the PRD in such a way for similar poles of various PRDs to be in the same respective row of the PRD. The matched PRDs can be interpolated and yield

accurate results according to [1]. This contribution applies the pole-matching technique to fROMs in PRD of an academic plate model. The Young's modulus is varied and the prediction accuracy of interpolated PRDs is shown. With the application of pole-matching, it becomes possible to efficiently employ simple methods to contain parametric changes in a MOR setting.

Pole-Matching Algorithm

In the previous section, the need for pole-matching was introduced briefly. The most accurate interpolation results are obtained for PRDs where similar poles are in the same respective row of different ROMs. The question arises, how similar poles are identified and how the rows in the PRDs are rearranged. Therefore, the algorithm introduced in [2] is depicted and explained shortly. Algorithm 1 introduces a branch and bound method to solve the optimization problem arising when matching poles. The goal is to compute an index vector v_d which stores the indexes of how the rows of the respective PRD needs to be rearranged, in order to match the poles. The input for the algorithm is two unmatched PRDs, for the output the second PRD's rows were rearranged according to v_d. The optimization problem that enables matching is the minimization of the function $f_d(v_d)$, which can be computed according to [2]

$$f_d(v_d) = ||\mathrm{PRD}_1 W - \mathbb{M}(v_d, \mathrm{PRD}_2) W||_F^2. \qquad (6)$$

Here W denotes weights, with which the PRDs can be multiplied, in order to emphasize either poles or residues when matching. In this contribution, the weights have been set to 1, as to examine the matching property without the influences of weights. $\mathbb{M}$ denotes a mapping function which rearranges the PRD according the index vector v_d, as previously discussed. $||\cdot||_F$ denotes the Frobenius norm. The final determination if poles have matched is done by minimizing Equation 6. Single entries of v_d are swapped and f_d is computed to check if the result minimized. The swaps are stored in the matrix $\mathbf{F}$. Since single entries that are next to each other are swapped, it becomes apparent that Algorithm 1 is most effective for PRDs where the similar poles are in close proximity to each other row wise [2]. Nonetheless, if f_d comes to a minimum, we can conclude that the poles have matched. In order to visualize the pole-matching algorithm, Figure 1 shows the comparison of the respective pole positions of two ROMs. The intersecting lines indicate that an accurate interpolation would not be possible, since the poles do not match at all. However, Figure 2 depicts the same comparison as Figure 1, this time after Algorithm 1 is applied. When poles have been matched, simple interpolation methods can be used to create PRDs at any point in the examined parameter space and evaluate the latter. But before the numerical results are presented, we

Algorithm 1: A branch and bound method to solve the pole-matching optimization problem [2].

```
Input: Two ROMs in PRD
Output: Two ROMs in PRD with poles matched
Initialize v_d = (1, 2, ..., nd), \\With nd ROM row
  size in PRD
the lowest current objective value b_l = f_d(v_d), and
  \\f_d is an error measure of two neighboring
  ROMs
F = J_nd − I_nd \\F is the matrix that indicates
  swaps that have been partaken already
while ||F||_F ≠ 0 do
    \\For every pole in ROM_1
    for i=1,...,nd do
        \\For every pole in ROM_2
        for j=1,...,nd do
            if F(v_{d,i}, v_{d,j}) = 0 then
                continue
            else
                set v_d^* = v_d
                swap the i-th and j-th entry of v_d^*
                if f(v_d^*) ≥ b_l then
                    Update F(v_{d,i}, v_{d,j}) = 0 \\Poles
                      have matched
                else
                    Set v_d = v_d^* and
                      b_l = f_d(v_d^*)\\Update index
                      vector and error
                    break both for loops
                end if
            end if
        end for
    end for
end while
```

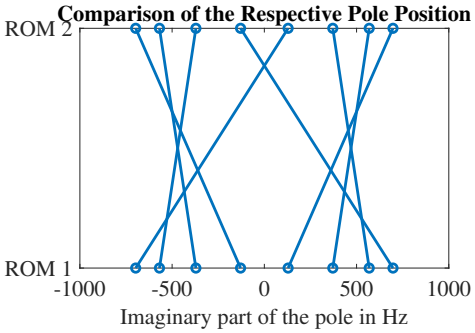

Figure 1: Comparison of the pole positions of two ROMs in PRD before the poles have been matched

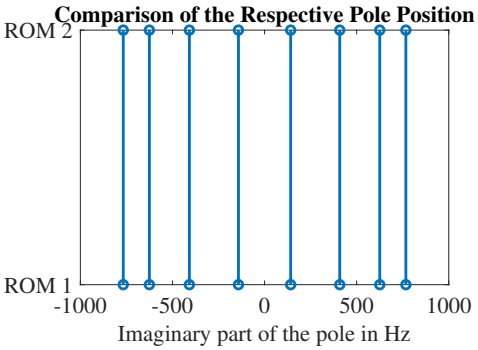

Figure 2: Comparison of the respective pole positions after Algorithm 1 has been applied

introduce the model used for the computations in this contribution.

Plate Model

As previously mentioned, before the final numerical results are presented, the examined mechanical model is introduced here. An academic plate model was used for the basic evaluation for the application of a pole-matching technique. The in-house research code Elementary Parallel Solver (elPaSo) from the Institute for Acoustics at the TU Braunschweig was used to generate the Reissner-Mindlin plate's system matrices. elPaSo utilizes the FEM for the computation of said matrices. Table 1 shows the basic material properties used for the examined plate. The plate possesses the isotropic material properties of

Table 1: Table depicting the model's basic properties

Property	Value
Material	Aluminum
Poisson's ratio	0.3
Thickness	0.003 m
Density	2700 $\frac{kg}{m^3}$
Young's modulus	70-73 GPa
Length	0.8 m
Breadth	0.4 m

aluminium. The purpose of pole-matching is to create parameter space spanning fROMs in PRD which can be interpolated. In order to evaluate the accuracy of those interpolated PRDs, in a first test the examined range is introduced with the Young's modulus. The system matrix size is 20000×20000 entries. The frequency ranges from $0 - 500$ Hz with a step size of 10 Hz. The examined plate is excited with a singular point load and the evaluations are also examined at the excitation, ultimately making it a single input-output system. For simplicity, damping is not considered.

Furthermore, in the introduction of this contribution the computation of the PRD was derived. In a real application it is only feasible to build the PRDs for ROMs, meaning that the actual MOR precedes the computation of PRDs. The regular MOR for the computation of the fROMs is also done with elPaSo. Here, a greedy approach for the choosing of expansion points is implemented, meaning that an error indicator is used to evaluate the accuracy of the fROMs transfer function and if the accuracy is less than a given tolerance, another expansion point is introduced at the point of largest error [3]. Because of deflation and the given accuracy tolerance, the size of the fROMs can differ, but in general we reduced the model to a size of 115×115. The size of the fROM is already a good indicator of the decrease of computational effort. Lastly, the PRDs of the fROMs could be built and interpolated, in order to span the examined parameter space.

Numerical Results

Consequently, in this section we will present the numerical results of the PMOR with pole-matching applied to the presented simple plate model. The main goal is to create parameter space spanning fROMs in PRD that can be interpolated accurately. In order to verify the presented approach, the ROM response is compared to the FOM response. Therefore, the transfer function is computed at several points in the examined parameter space. Since the transfer function spans the frequency domain, we obtain a surface plot which can be seen in Figure 3. The figure allows for a quick and easy evaluation of the

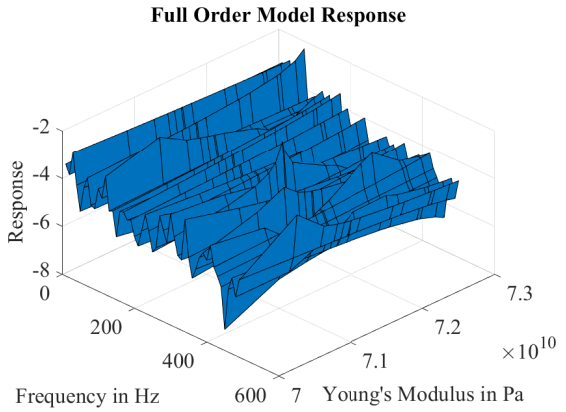

Figure 3: Visualization of the FOM's response behavior

system's behavior across the frequency as well as the parameter domain. Subsequently, we built several different fROMs at different points in the parameter space as to yield accurate interpolation of PRDs. Here, the adaptive approach proposed in [2] is used. PRDs at the beginning and end of the examined parameter space are supplied and an interpolated ROM is built at the furthest distance from both grid points, meaning right in-between said PRDs. If the error measure of transfer functions

$$\epsilon = \sup_{\omega \in \mathbb{R}} ||\mathbf{H}(i\omega) - \mathbf{H}_r(i\omega)||_2 \tag{7}$$

is smaller than a given tolerance, $\tau = 1.00 \times 10^{-3}$, the interpolation is seen as sufficiently accurate. If the tolerance is not met, the fROM built for the computation of the error is inserted as an additional grid point. Therefore, the grid is refined adaptively. Figure 4 depicts the interpolated surface plot of the ROM's response.

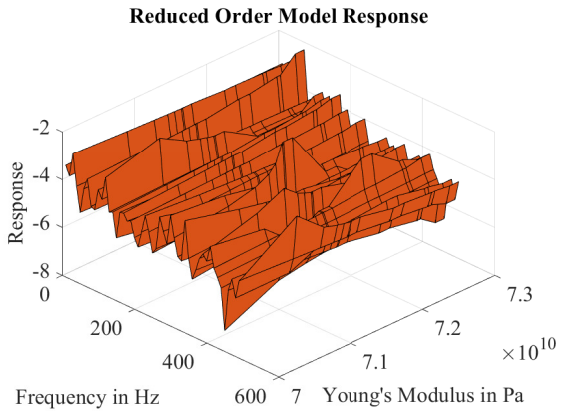

Figure 4: Visualization of the ROM's response behavior

Furthermore, nearest-neighbor interpolation was used

here, since the linear interpolation sometimes introduced large errors when having to interpolate a relatively significant pole. The nearest neighbor interpolation obtained good results and it was possible to compute 9 repository fROMs spanning the parameter space, with which an accurate interpolation becomes possible.
When comparing Figure 3 and 4 qualitatively, it becomes obvious that both surface plots match pretty well. However, some artifacts of the nearest neighbor interpolation introduce a small error when the transfer function is relatively steep. For a better visualization the error as introduced in Equation 7 is also computed for every transfer function. Since the error is valid in the frequency domain, it can be computed for several different parameter points and plotted across the parametric range. This can be seen in Figure 5. Of course, the error at interpolation

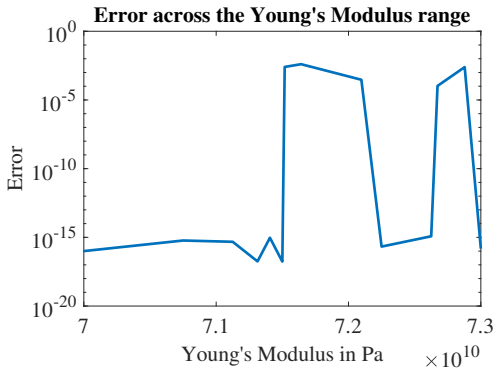

Figure 5: Obtained error across the Young's modulus range

grid points, meaning the repository ROMs, is numerically zero, since here the grid points can be used respectively. Furthermore, it becomes obvious that the first half of the examined parameter space can be depicted quite well by interpolation and in the second half the error increases significantly. However, we have to keep in mind that the ROM's accuracy is about 1.00×10^{-6} and for that the comparison of the FOM and interpolated PRD with an error of 1.00×10^{-3} is still seen as sufficient. Lastly, in order to visualize that the point with the largest error also depicts the FOM's behavior well, the transfer function at that specific point is plotted and compared to the interpolated PRD's transfer function. This is depicted in Figure 6. It can be seen that the prediction accuracy is still really high, except for the response around the pole at frequency 310 Hz. Still, the interpolated PRD's transfer function gives a promising insight into the FOM's behavior, while yielding only a fraction of the computational effort. Therefore, the PRD interpolation in the considered parametric domain is effective with the presented PMOR approaches.

Conclusion

Finally, we conclude the most important findings of this contribution. The pole-matching technique is a promising method to reduce the computational effort for PMOR by utilizing simple interpolation, in order to span the examined parameter space. We showed that the pole-matching algorithm works as intended by visualizing the

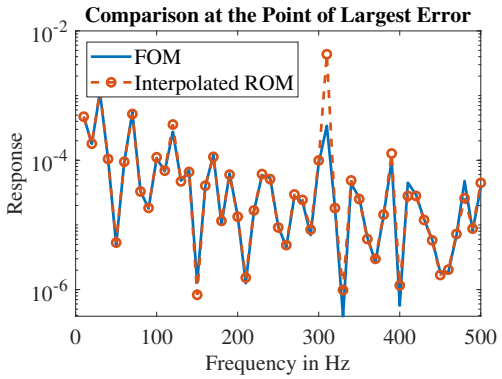

Figure 6: Comparsion of the FOM's and interpolated PRD's transfer function at the point of largest error

pole positions in different ROMs before and after the algorithm has been applied. Furthermore, it became obvious that the nearest neighbor interpolation is sufficient to span the examined parameter space in the Young's modulus range. Therefore, we can conclude that pole-matching and interpolating PRDs is suited to accurately depict a simple model's behavior and is worth pursuing for PMOR. 9 ROMs, in our case reducing the original system size by a factor of 100, were enough to accurately span the examined parameter subspace, meaning the computational effort is also decreased significantly. Further details on the PMOR application for other parameters mentioned in Table 1 is not considered in the scope of this work. A detailed study is a point for future research.
The next steps include examining a more complex coupled model with fluid-structure interaction, which also introduces more parameters as well as the coupling terms, which can also introduce additional problems. Furthermore, a different way of computing PRDs is also a topic of future works, where the computational effort can be reduced further, by not needing a transformation to a first order system.

References

[1] Yue, Yao; Feng, Lihong; Benner, Peter (2019): Reduced-order modelling of parametric systems via interpolation of heterogeneous surrogates. In Adv. Model. and Simul. in Eng. Sci. 6 (1). DOI: 10.1186/s40323-019-0134-y.

[2] Yue, Yao; Feng, Lihong; Benner, Peter (2019): An Adaptive Method for Interpolating Reduced-Order Models Based on Matching and Continuation of Poles. 2019 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO)

[3] Sreekumar, Harikrishnan K.; Ullmann, Rupert; Sicklinger, Stefan; Langer, Sabine C.: Efficient Krylov Subspace Techniques for Model Order Reduction of Automotive Structures in Vibroacoustic Applications, in Model Reduction of Complex Dynamical Systems (2021), pages 259-282, Springer International Publishing, DOI: 10.1007/978 − 3 − 030 − 72983 − 712