MACHINE TOOL SIMULATION BASED ON REDUCED ORDER FE MODELS

Heike Faßbender, TU Braunschweig, Germany

Andreas Soppa, TU Braunschweig, Germany

Corresponding author: Andreas Soppa
Institut *Computational Mathematics*
TU Braunschweig – Pockelsstr. 14, 38106 Braunschweig – Germany
Email: `andreas.soppa@tu-bs.de`

Abstract.

Numerical simulations of the behavior of machine tools are usually based on a finite element (FE) discretization of their mechanical structure. In order to capture all necessary details FE models are in general very large and sparse. Hence the computation of the simulation takes an unacceptable long time and requires much memory space. To calculate the results in reasonable time typically modal reduction is used to obtain a model of lower order. This method has the disadvantage that the reduced system only contains information of the modes chosen to generate the reduced system. Moreover, the choice of the essential modes is usually based on a heuristic and cannot be fully automated. Therefore, there is a need for alternate reduction methods which can be fully automated.

In the last years new methods to reduce large and sparse dynamical systems were presented. The two most important ones are balanced truncation approximations (BTA) and Krylov subspace methods. Here the main focus is on the reduction of the special FE systems with Krylov subspace methods. The main goal of this work is to discuss whether these methods are suitable for this type of application, not on the derivation of a new reduction technique. Several Krylov subspace methods for first or second order systems based on the Interpolation-Rational-Krylov-Algorithm (IRKA) were implemented. The reduced systems are compared with reduced systems obtained by modal reduction and BTA.

1 Introduction

The simulation of a machine tool is based on two major parts: a structural model of the machine tool representing its behavior and reaction on certain control inputs and the control loop generating those inputs. Figure 1 shows the entire set up of the simulation for the example of a control loop to control the torque of an axis motor.
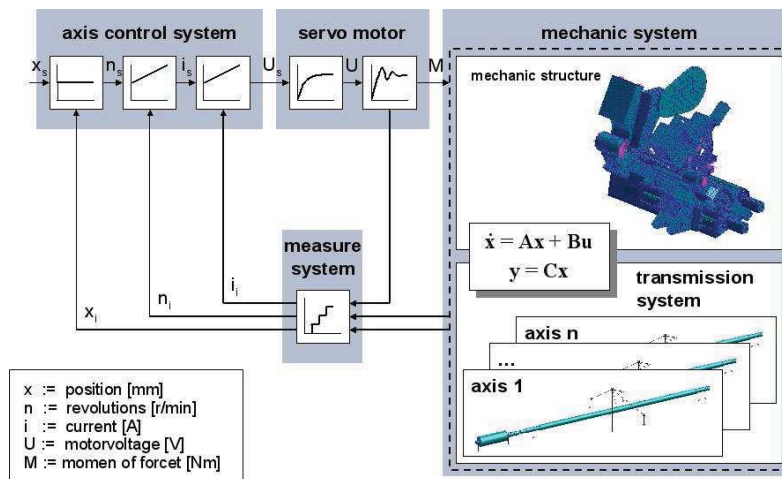

Figure 1: Control loop and embedded mechanical structure.

To describe the behavior and the reaction of the mechanical structure on control inputs a system of FE semi-discritized partial differential equations (PDE) is used.

After linearization one obtains a system of ordinary differential equations (ODE) of second order:

$$
\begin{aligned}
M\ddot{x}(t) + D\dot{x}(t) + Kx(t) &= Bu(t) \\
y(t) &= C_v^T \dot{x}(t) + C_p^T x(t)
\end{aligned}
\tag{1}
$$

where $M, D, K \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times p}$, $C_v^T, C_p^T \in \mathbb{R}^{m \times n}$, $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^p$, $y(t) \in \mathbb{R}^{m \times p}$. In the special case discussed here the damping matrix D is chosen as

$$
D = \alpha \cdot M + \beta \cdot K,
$$

that is, D is proportional to the mass matrix M and the stiffness matrix K, where α and β are real parameters which are chosen by the experience of the design engineer and lie between 0 and 0.1. This kind of proportional damping

is called Rayleigh damping. The system matrices are sparse, of high order and non-symmetric. Further the system may be unstable. All of this accounts for unacceptable computational and resource demands in simulation and control of these models. In order to reduce these demands to acceptable computational times, usually model order reduction techniques are employed which generate a reduced order model that captures the essential dynamics of the system and preserves its important properties.

Model order methods are methods to find a second order system of reduced dimension $k \ll n$

$$
\begin{aligned}
\hat{M}\ddot{\hat{x}}(t) + \hat{D}\dot{\hat{x}}(t) + \hat{K}\hat{x}(t) &= \hat{B}u(t) \\
\hat{y}(t) &= \hat{C}_v^T \dot{\hat{x}}(t) + \hat{C}_p^T \hat{x}(t),
\end{aligned}
\tag{2}
$$

where $\hat{M}, \hat{D}, \hat{K} \in \mathbb{R}^{k \times k}$, $\hat{B} \in \mathbb{R}^{k \times p}$, $\hat{C}_v^T, \hat{C}_p^T \in \mathbb{R}^{m \times k}$, $\hat{x}(t) \in \mathbb{R}^k$, $u(t) \in \mathbb{R}^p$, $\hat{y}(t) \in \mathbb{R}^{m \times p}$, which approximates the original system in some sense.

Moreover, as any second order model can be transformed into a first order system called state space form

$$
\underbrace{\begin{bmatrix} I & 0 \\ 0 & M \end{bmatrix}}_{E} \underbrace{\begin{bmatrix} \dot{x}(t) \\ \ddot{x}(t) \end{bmatrix}}_{\dot{z}(t)} = \underbrace{\begin{bmatrix} 0 & I \\ -K & -D \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix}}_{z(t)} + \underbrace{\begin{bmatrix} 0 \\ B \end{bmatrix}}_{G} u(t),
$$

$$
y(t) = \underbrace{\begin{bmatrix} C_p^T & C_v^T \end{bmatrix}}_{C^T} \underbrace{\begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix}}_{z(t)},
\tag{3}
$$

where $E, A \in \mathbb{R}^{2n \times 2n}$, $G \in \mathbb{R}^{2n \times p}$, $C^T \in \mathbb{R}^{m \times 2n}$, $z(t) \in \mathbb{R}^{2n}$, $u(t) \in \mathbb{R}^p$, $y(t) \in \mathbb{R}^{m \times p}$, and $I \in \mathbb{R}^{n \times n}$ is the identity matrix. By the transformation process the dimension of the system doubles. The corresponding reduced order system is of the form

$$
\begin{aligned}
\hat{E}\dot{\hat{z}}(t) &= \hat{A}\hat{z}(t) + \hat{G}u(t) \\
\hat{y}(t) &= \hat{C}^T \hat{z}(t),
\end{aligned}
\tag{4}
$$

where $\hat{E}, \hat{A} \in \mathbb{R}^{k \times k}$, $\hat{G} \in \mathbb{R}^{k \times p}$, $\hat{C}^T \in \mathbb{R}^{m \times k}$, $\hat{z}(t) \in \mathbb{R}^k$, $u(t) \in \mathbb{R}^p$, $\hat{y}(t) \in \mathbb{R}^{m \times p}$.

In engineering the modal reduction [2] is most common. In the last years new reduction methods to reduce large and sparse dynamical systems were presented. The two most famous methods are balanced truncation approximations (BTA) and Krylov subspace methods. The efficient application of BTA to the specific problem considered here is discussed by Benner and Saak in another contribution.

Here we consider the reduction of large structural mechanical FE models by Krylov subspace methods.

2 Krylov subspace methods

A Krylov subspace is defined by

$$
\mathcal{K}_r(P, q) = \text{span}\{q, Pq, P^2 q, \cdots, P^{r-1} q\}
\tag{5}
$$

where $P \in \mathbb{R}^{n \times n}$ and $q \in \mathbb{R}^n$. q is called the starting vector.

the reduced first order systems. For our purposes, a Krylov subspace method generates a Petrov-Galerkin projection Π which can be used to project the system (1) (resp., (3)) onto a system (2) (resp., (4)) of dimension $k \ll n$. For that Krylov subspace methods generate matrices $V \in \mathbb{R}^{2n \times k}$ and $W \in \mathbb{R}^{2n \times k}$ with $W^T V = I_k$ such that the reduced first order system (4) is constructed by applying the Petrov-Galerkin projection $\Pi = VW^T$ to (3) such that

$$
\hat{E} = W^T E V, \quad \hat{A} = W^T A V, \quad \hat{G} = W^T G, \quad \text{and} \quad \hat{C}^T = C^T V.
\tag{6}
$$

Similarly, the reduced second order system (2) is constructed by applying a Petrov-Galerkin projection to (1) such that

$$
\hat{M} = W^T M V, \quad \hat{D} = W^T D V, \quad \hat{K} = W^T K V, \quad \hat{B} = W^T B, \quad \hat{C}_p^T = C_p^T V, \quad \text{and} \quad \hat{C}_v^T = C_v^T V,
\tag{7}
$$

where $V \in \mathbb{R}^{n \times k}$ and $W \in \mathbb{R}^{n \times k}$ with $W^T V = I_k$.

In both cases, the columns of V and W span certain Krylov subspaces. As the vectors in the Krylov sequence $q, Pq, P^2 q, \ldots$ tend to become almost numerically linearly dependent even for a moderate dimension of P, the columns of V and W are chosen such that they represent a (numerically) better basis of the corresponding Krylov subspace (e.g., they result from orthogonalizing a Krylov sequence or from bi-orthogonalizing two Krylov sequences). Usually, this is achieved by employing the Lanczos- or the Arnoldi algorithm, see Section 3 for more details.

A block Krylov subspace is defined by

$$
\mathcal{K}_r(P, Q) = \text{span}\{Q, PQ, P^2 Q, \cdots, P^{r-1} Q\} = \text{span}\{v_0, v_1, \ldots, v_{r-1}\},
\tag{8}
$$

where $P \in \mathbb{R}^{n \times n}$ and the columns of $Q \in \mathbb{R}^{n \times \ell}$ are the linearly independent. Such a block Krylov subspace with ℓ starting vectors (assembled in Q) can be considered as a union of ℓ Krylov subspaces defined for each starting vector. For more details on Krylov subspace methods see, e.g., [12].

2.1   First order systems

The transfer function of a first order system (3) is the linear mapping of the Laplace transformation of the input to the output

$$H(s) = C^T(sE - A)^{-1}G.$$

After expansion in a Laurent expansion series around an expansion point s_0 one obtains the moments $h_j(s_0)$, $j = 0,..,\infty$ of the transfer function

$$H(s) \quad = \quad \sum_{j=0}^{\infty} h_j(s_0)(s - s_0)^j$$

$$\text{where} \quad h_j(s_0) \quad = \quad C^T[(A - s_0E)^{-1}E]^j(A - s_0E)^{-1}G.$$

Consider the block Krylov subspace (8) for

$$P = (A - s_0E)^{-1}E \text{ and } Q = (A - s_0E)^{-1}G. \tag{9}$$

Assume that a (block) orthogonal basis for this block Krylov subspace is generated using a suitable method. Choose the column vectors of $V = W$ as the first k vectors of that basis and apply the congruence transformation (6) (with $V, W = V$ and $V^T V = I$). Then the transfer function of the resulting reduced system matches at least the first $\lfloor k/p \rfloor$ moments of the transfer function of the original system [1]. That is, at least the first $\lfloor k/p \rfloor$ moments $\hat{h}_j(s_0)$, of the transfer function $\hat{H}(s)$ of the reduced system (3) equal the first moments $h_j(s_0)$, of the transfer function $H(s)$ of the original system (4) at the expansion point s_0

$$h_j(s_0) = \hat{h}_j(s_0), \quad j = 0, 1, \cdots, \lfloor k/p \rfloor - 1.$$

When V is determined from the block Krylov subspace generated by

$$P = (A - s_0E)^{-1}E \text{ and } Q = (A - s_0E)^{-1}G,$$

while W is determined from the block Krylov subspace generated by

$$P = (A - s_0E)^{-T}E^T \text{ and } Q = (A - s_0E)^{-T}C,$$

such that $W^T V = I$, then the transfer function of the system obtained by applying the congruence transformation (6) with V and W matches at least the first $\lfloor k/p + k/m \rfloor$ moments of the transfer function of the original system [1].

An alternative is to use more than one expansion point, these methods are called Multi-Point-Rational-Interpolation methods [5]. Assume that $\hat{i}$ expansion points s_i, $i = 1, 2, \cdots, \hat{i}$ are to be considered. As before, at least $\lfloor k_i/p \rfloor$ moments are to be matched per expansion point s_i. Let $k_i \in \mathbb{N}$ such that $\sum_{i=1}^{\hat{i}} k_i = k$. Now, these methods require that at least the first $\lfloor k_i/p \rfloor$ moments $\hat{h}_j(s_i)$ of the transfer function $\hat{H}(s)$ of the reduced system equal the first moments $h_j(s_i)$ of the transfer function of the original system $H(s)$ at the expansion points s_i, $i = 1, 2, \cdots, \hat{i}$:

$$h_j(s_i) = \hat{h}_j(s_i), \quad j = 0, 1, \cdots, \lfloor k_i/p \rfloor - 1; \quad i = 1, 2, \cdots, \hat{i}$$

The column vectors of the matrix V are determined from the $\hat{i}$ block Krylov subspaces generated by

$$P = (A - s_iE)^{-1}E \text{ and } Q = (A - s_iE)^{-1}G, \quad i = 1, 2, \ldots, \hat{i}.$$

From each of these subspaces, k_i column vectors are used to generate V.
Similarly, the column vectors of matrix W are determined from the $\hat{i}$ block Krylov subspaces generated by

$$P = (A - s_iE)^{-T}E^T \text{ and } Q = (A - s_iE)^{-T}C.$$

In [6] the choice of expansion points s_i, $i = 1, \ldots, \hat{i}$ is discussed. Starting from an initial set of expansion points s_i a reduced order system is determined. Then a new set of expansion points are assigned by $s_i = -\lambda_i$, $i = 1, \ldots, \hat{i}$ where λ_i are the eigenvalues of the matrix pencil $\hat{E} - \lambda\hat{A}$ with $\hat{E}, \hat{A}$ as in (4) ordered such that $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n|$. With this new set of expansion points, typically a reduction with better approximation abilities is obtained.

2.2 Second order systems

The bottleneck of the computation of reduced systems by Krylov subspace methods is to solve linar systems of the form (9). By the transformation of a second order system into a first order system the dimension of the system doubles. Therefore the reduction of the second order system may be more effective if the dimension of the original model is very large. The transfer function of a second order system is given by the Laplace transformation of (1):

$$H(s) = (C_p + sC_v)^T (s^2 M + sD + K)^{-1} B.$$

After expansion in a Laurent expansion series around an expansion point s_0 one obtains the moments $h_j(s_0)$, $j = 0, .., \infty$ of the transfer function

$$H(s) = \sum_{j=0}^{\infty} h_j(s_0)(s - s_0)^j$$

$$\text{where} \quad h_j(s_0) = (\tilde{C}_p + s_0 C_v)^T [(-s_0^2 M - s_0 \tilde{D} - \tilde{K})^{-1} M]^j (-s_0^2 M - s_0 \tilde{D} - \tilde{K})^{-1} B,$$

with $\tilde{D} = 2s_0 M + D$, $\tilde{K} = s_0^2 M + s_0 D + K$ and $\tilde{C}_p = (C_p + s_0 C_v)$. Here we consider the special case of systems with Rayleigh damping. In [7] it is shown, that using

$$P = -(s_0^2 M + s_0 D + K)^{-1} M \text{ and } Q = -(s_0^2 M + s_0 D + K)^{-1} B$$

in (8) to generate k column vectors for matrix V the transfer function of the system reduced by applying the congruence transformation (7) with V, $W = V$ and $V^T V = I_k$ matches at least the first $\lfloor k/p \rfloor$ moments of the transfer function of the original system [7].

In case $V \neq W$, but $W^T V = I$ is chosen, W is determined from the block Krylov subspace generated by

$$P = -(s_0^2 M + s_0 D + K)^{-T} M^T \text{ and } Q = -(s_0^2 M + s_0 D + K)^{-T} (C_p + s_0 C_v).$$

The transfer function of the system obtained by applying the congruence transformation (7) with V and W matches at least the first $\lfloor k/p + k/m \rfloor$ moments of the transfer function of the original system [7].

When more than one expansion point is used the column vectors of matrix V are determined from the block Krylov subspaces

$$P = -(s_i^2 M + s_i D + K)^{-1} M \text{ and } Q = -(s_i^2 M + s_i D + K)^{-1} B,$$

while the column vectors of matrix W are determined from the block Krylov subspaces

$$P = -(s_i^2 M + s_i D + K)^{-T} M^T \text{ and } Q = -(s_i^2 M + s_i D + K)^{-T} (C_p + s_i C_v).$$

3 Implementation

To reduce first and second order systems methods based on the Interpolation-Rational-Krylov-Algorithm (IRKA) [6] were implemented.

3.1 First order systems

To reduce the state space systems the following Krylov subspace methods were used:

1. Padé-via-Arnoldi method (PVA) by Odabasioglu and Celik [10].
2. Padé-via-Lanczos method (PVL) by Feldmann and Freund [8].
3. Rational-Arnoldi method (RA) by Grimme [5].
4. Rational-Lanczos method (RL) by Gallivan, Grimme and Dooren [9].

The first two methods use only one expansion point, while the last two methods deal with more than one expansion point. In the following algorithms pseudo code of the implemented methods are given.

- The code of methods with one expansion point is given as Algorithm 1.
- The code of the Arnoldi algorithm is given as Algorithm 2.
- The code of the Lanczos algorithm is given as Algorithm 3.
- The code for the methods with more than one expansion point is given as Algorithm 4.

The methods are implemented to handle the sparse structure of the system matrices. To handle systems with one input but more than one output the block forms of the Krylov-methods were implemented.

Algorithm 1 Padé-via-Arnoldi method / Padé-via-Lanczos method

---

Input: system matrices $\mathbf{E}, \mathbf{A}, \mathbf{G}, \mathbf{C}$; expansion point $\mathbf{s}$; reduced order $\mathbf{r}$;
    deflation tolerance **defltol**
Output: reduced system matrices $\hat{\mathbf{E}}, \hat{\mathbf{A}}, \hat{\mathbf{G}}, \hat{\mathbf{C}}$
  1: function $[\hat{\mathbf{E}}, \hat{\mathbf{A}}, \hat{\mathbf{G}}, \hat{\mathbf{C}}] = \text{PvA}(\mathbf{E}, \mathbf{A}, \mathbf{G}, \mathbf{C}, \mathbf{s}, \mathbf{r}, \textbf{defltol})$
           [$[\hat{\mathbf{E}}, \hat{\mathbf{A}}, \hat{\mathbf{G}}, \hat{\mathbf{C}}] = \text{PvL}(\mathbf{E}, \mathbf{A}, \mathbf{G}, \mathbf{C}, \mathbf{s}, \mathbf{r}, \textbf{defltol})$ **]**
  2: $\mathbf{V} = [\,], \mathbf{W} = [\,]$
  3: $[\mathbf{L}, \mathbf{U}] = lu(\mathbf{A} - \mathbf{sE})$
  4: **for** $i = 1$ to $\mathbf{r}$ **do**
  5:      $\mathbf{V} = \text{Krylov}(\mathbf{L}, \mathbf{U}, \mathbf{E}, \mathbf{G}, \mathbf{V}, (\mathbf{W}), \textbf{defltol}, i)$
  6:      $\mathbf{W} = \text{Krylov}(\mathbf{U}^T, \mathbf{L}^T, \mathbf{E}^T, \mathbf{C}^T, (\mathbf{W}), \mathbf{V}, \textbf{defltol}, i)$
  7: **end for**
  8: **if** $\#columns(\mathbf{V}) \neq \#columns(\mathbf{W})$ **then**
  9:      make $\mathbf{V}$ and $\mathbf{W}$ of the same dimension
10: **end if**
11: $[\hat{\mathbf{E}}, \hat{\mathbf{A}}, \hat{\mathbf{G}}, \hat{\mathbf{C}}] = \text{transform}(\mathbf{E}, \mathbf{A}, \mathbf{G}, \mathbf{C}, \mathbf{V}, \mathbf{W})$
12: if the reduced system is unstable reduce dimension of $\mathbf{V}$ and $\mathbf{W}$ until the system is stable.

Some remarks on Algorithm 1:

In line 3 the LU decomposition of the matrix $(\mathbf{A} - \mathbf{sE})$ is computed.

In line 5 and 6 the matrices $\mathbf{V}$ and $\mathbf{W}$ are computed by the Arnoldi or the Lanczos algorithm. When the Arnoldi algorithm is used the matrices $\mathbf{V}$ and $\mathbf{W}$ are orthonormal, i.e. $\mathbf{V}^T \mathbf{V} = I$ and $\mathbf{W}^T \mathbf{W} = I$. When the Lanczos Algorithm is used the matrices $\mathbf{V}$ and $\mathbf{W}$ are biorthonormal, i.e. $\mathbf{W}^T \mathbf{V} = \Delta$, where Δ is a diagonal matrix.

In line 9 the dimensions of $\mathbf{V}$ and $\mathbf{W}$ were adapted: The dimensions of the matrices may be not equal. The cause are deflated vectors or an unequal number of input and output vectors $(p \neq m)$. There are two possibilities to adapt the dimension of $\mathbf{V}$ and $\mathbf{W}$: either reduce the number of columns of the matrix with higher dimension by truncating the last columns, or use the Krylov algorithm to generate additional columns for the matrix at lower dimension.

In line 11 the congruence transformation (6) is computed.

In line 12 the dimension of the matrices $\mathbf{V}$ and $\mathbf{W}$ is decreased by truncating the last columns, until the reduced system is stable.

Algorithm 2 Block Arnoldi method

Input: matrices **L,U,E,G,V,W**; deflation tolerance **defltol**; index **i**
Output: transformation matrix **V**
  1: function $[\mathbf{V}] = \text{Block\_Arnoldi}(\textbf{L,U,E,G,V,defltol,i})$
  2: **if** $(\mathbf{i} == 1)$ **then**
  3:      $\mathbf{R} = \mathbf{U}\backslash\mathbf{L}\backslash\mathbf{G}$
  4: **else**
  5:      $r = \#columns(\mathbf{V})$
  6:      $s = r - \#columns(\mathbf{G}) + 1$
  7:      $\tilde{\mathbf{V}} = [\mathbf{V}_{*s}, \cdots, \mathbf{V}_{*r}]$
  8:      $\mathbf{R} = \mathbf{U}\backslash\mathbf{L}\backslash(\mathbf{E} \cdot \tilde{\mathbf{V}})$
  9: **end if**
10: **for** $j = 1$ to $\#columns(\mathbf{R})$ **do**
11:      **for** $k = 1$ to $\#columns(\tilde{\mathbf{V}})$ **do**
12:          $h = (\mathbf{R}_{*j}, \tilde{\mathbf{V}}_{*k})_2$
13:          $\mathbf{R}_{*j} = \mathbf{R}_{*j} - h \cdot \tilde{\mathbf{V}}_{*k}$
14:      **end for**
15:      $h = \sqrt{(\mathbf{R}_{*j}, \mathbf{R}_{*j})_2}$
16:      **if** $(h > \textbf{defltol})$ **then**
17:          $\hat{\mathbf{V}} = \frac{1}{h}\mathbf{R}_{*j}$
18:          $\mathbf{V} = [\mathbf{V}\ \hat{\mathbf{V}}]$
19:      **else**
20:          Deflation
21:      **end if**
22: **end for**

Algorithm 3 Block Lanczos method

Input: matrices **L,U,E,G,V,W**; deflation tolerance **defltol**; index **i**
Output: transformation matrix **V**
1: function [**V**] = Block_Lanczos(**L,U,E,G,V,W,defltol,i**)
2: **if** (**i** == 1) **then**
3:     **R** = **U****L****G**
4: **else**
5:     $r = \#columns(\mathbf{V})$
6:     $s = r - \#columns(\mathbf{G}) + 1$
7:     $\tilde{\mathbf{V}} = [\mathbf{V}_{*s}, \cdots, \mathbf{V}_{*r}]$
8:     **R** = **U****L**\(**E** · $\tilde{\mathbf{V}}$)
9: **end if**
10: **if** $i \neq 1$ **then**
11:     **for** $j = 1$ to $\#columns(\mathbf{R})$ **do**
12:         **for** $k = (\#columns(\tilde{\mathbf{V}}) - 2)$ to $\#columns(\tilde{\mathbf{V}})$ **do**
13:             $h = (\mathbf{R}_{*j}, \mathbf{W}_{*k})_2$
14:             $\mathbf{R}_{*j} = \mathbf{R}_{*j} - h \cdot \tilde{\mathbf{V}}_{*k}$
15:         **end for**
16:         $h = \sqrt{(\mathbf{R}_{*j}, \mathbf{W}_{*j})_2}$
17:         **if** ($h >$ **defltol**) **then**
18:             $\hat{\mathbf{V}} = \frac{1}{h}\mathbf{R}_{*j}$
19:             $\mathbf{V} = [\mathbf{V} \ \hat{\mathbf{V}}]$
20:         **else**
21:             Deflation
22:         **end if**
23:     **end for**
24: **end if**

Some remarks on Algorithm 2 and 3:

In line 3 the solution of $(\mathbf{A} - s\mathbf{E})x = \mathbf{G}$ is computed via forward and backward substitution using the LU decomposition of the matrix $(\mathbf{A} - s\mathbf{E})$.

In line 5 and 6 the vectors to generate the block matrix are detected and in line 7 the block matrix is generated.

In line 8 the solution of $(\mathbf{A} - s\mathbf{E})x = (\mathbf{E} \cdot \tilde{\mathbf{V}})$ is computed via forward and backward substitution with the LU decomposition of the matrix $(\mathbf{A} - s\mathbf{E})$.

In line 20 respective 21 vectors $\mathbf{R}_{*j}$ with $h <$ **defltol** are deflated, they are not used to generate the matrix V.

Algorithm 4 Rational-Arnoldi method / Rational-Lanczos method

---

Input: system matrices $\mathbf{E}, \mathbf{A}, \mathbf{G}, \mathbf{C}$; expansion points: $\mathbf{s}_k$, $k = 1, \ldots, K$; tolerance **tol** ; vector with number of expansions **J** for every $\mathbf{s}_k$; deflation tolerance **defltol**
Output: reduced system matrices $\hat{\mathbf{E}}, \hat{\mathbf{A}}, \hat{\mathbf{G}}, \hat{\mathbf{C}}$
1: function $[\hat{\mathbf{E}}, \hat{\mathbf{A}}, \hat{\mathbf{G}}, \hat{\mathbf{C}}]$ = RA(**E, A, G, C, s, J, defltol**)
        [$[\hat{\mathbf{E}}, \hat{\mathbf{A}}, \hat{\mathbf{G}}, \hat{\mathbf{C}}]$ = RL(**E, A, G, C, s, J, defltol**)]
2: **while** relative change in one $\mathbf{s}_k >$ **tol do**
3:     $\mathbf{V} = [\ ], \mathbf{W} = [\ ]$
4:     **for** $k = 1$ to K **do**
5:         $[\mathbf{L}, \mathbf{U}] = lu(\mathbf{A} - \mathbf{s}_k\mathbf{E})$
6:         **for** $i = 1$ to $\mathbf{J}[k]$ **do**
7:             $\mathbf{V}$ = Krylov(**L, U, E, G,** , **V, (W), defltol**, i)
8:             $\mathbf{W}$ = Krylov($\mathbf{U}^T, \mathbf{L}^T, \mathbf{E}^T, \mathbf{C}^T$, **W, (V), defltol**, i)
9:         **end for**
10:         **if** ($\#columns(\mathbf{V}) \neq \#columns(\mathbf{W})$) **then**
11:             make **V** and **W** of the same dimension
12:         **end if**
13:     **end for**
14:     $\mathbf{E}_r = \mathbf{W}^T\mathbf{E}\mathbf{V}$, $A_r = \mathbf{W}^T\mathbf{A}\mathbf{V}$
15:     assign $\mathbf{s}_k \leftarrow -\lambda_k(\mathbf{E}_r, \mathbf{A}_r)$ for $k = 1, \ldots, K$
16: **end while**
17: $[\hat{\mathbf{E}}, \hat{\mathbf{A}}, \hat{\mathbf{G}}, \hat{\mathbf{C}}]$ = transform(**E, A, G, C, V, W**)
18: if the reduced system is unstable reduce dimension of **V** and **W**

Some remarks on Algorithm 4:

$\mathbf{s}_k$, $k = 1, \ldots, K$ are the expansion points. $\mathbf{J}$ is a column vectors with the number of expansions computed at expansion point $\mathbf{s}_k$.

In line 5 the LU decomposition of the matrix $(\mathbf{A} - \mathbf{s}_k\mathbf{E})$ is computed.

In line 7 and 8 the matrices $\mathbf{V}$ and $\mathbf{W}$ are computed by the Arnoldi or the Lanczos algorithm. When the Arnoldi algorithm is used the matrices $\mathbf{V}$ and $\mathbf{W}$ are orthonormal, i.e. $\mathbf{V}^T\mathbf{V} = I$ and $\mathbf{W}^T\mathbf{W} = I$. When the Lanczos algorithm is used the matrices $\mathbf{V}$ and $\mathbf{W}$ are biorthonormal, i.e. $\mathbf{W}^T\mathbf{V} = \Delta$, where Δ is a diagonal matrix.

In line 11 the dimensions of $\mathbf{V}$ and $\mathbf{W}$ were made equal, see remarks on Algorithm 1 for details.

In line 14 the congruence transformation (6) for $\mathbf{E}$ and $\mathbf{A}$ is computed.

In line 15 new expansion points $\mathbf{s}_k$ are chosen from the eigenvalues of $(\mathbf{E}_r, \mathbf{A}_r)$ in a relevant frequency interval computed by a generalized eigenproblem.

In line 17 the congruence transformation (6) is computed.

In step 18 the dimension of the matrices $\mathbf{V}$ and $\mathbf{W}$ is reduced by truncating the last columns, until the reduced system is stable.

3.2 Second order systems

To reduce the second order system the following Krylov subspace methods were used:

1. Rational Arnoldi for systems without damping matrix (RA_2o) [5]
2. Rational Lanczos for systems without damping matrix (RL_2o) [9]
3. Rational Arnoldi for systems with proportional damping (RA_PD) by Gugercin [7]
4. Rational Lanczos for systems with proportional damping (RL_PD)

The methods have more than one expansion point. The first two methods are modified versions of the Rational-Arnoldi and the Rational-Lanczos method for second order systems without damping matrix. The last two methods exploit the special structure of the proportional damping matrix. The pseudo code for the first two methods is given as Algorithm 5 and for the last two methods as Algorithm 6.

Algorithm 5 Rational-Arnoldi method / Rational-Lanczos method for second order systems

---

Input: system matrices $\mathbf{M}, \mathbf{K}, \mathbf{B}, \mathbf{C}_p, \mathbf{C}_v$; expansion points: $\mathbf{s}_k$, $k = 1, \ldots, K$; tolerance **tol** ; vector with number of expansions $\mathbf{J}$ for every $\mathbf{s}_k$; deflation tolerance **defltol**

Output: reduced system matrices $\hat{\mathbf{M}}, \hat{\mathbf{K}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}_p, \hat{\mathbf{C}}_v$

1: function $[\hat{\mathbf{M}}, \hat{\mathbf{K}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}_p, \hat{\mathbf{C}}_v]$ = RA_2o($\mathbf{M}, \mathbf{K}, \mathbf{B}, \mathbf{C}_p, \mathbf{C}_s, \mathbf{J}, $**defltol**)
   $[\ [\hat{\mathbf{M}}, \hat{\mathbf{K}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}_p, \hat{\mathbf{C}}_v]$ = RL_2o($\mathbf{M}, \mathbf{K}, \mathbf{B}, \mathbf{C}_p, \mathbf{C}_s, \mathbf{J}, $**defltol**) $]$

2: **while** relative change in one $\mathbf{s}_k >$ **tol** **do**

3:    $\mathbf{V} = [\ ], \mathbf{W} = [\ ]$

4:    **for** $k = 1$ to K **do**

5:       $[\mathbf{L}, \mathbf{U}] = lu(-\mathbf{s}_k^2\mathbf{M} - \mathbf{K})$

6:       $\mathbf{C} = \mathbf{C}_p + \mathbf{s}_k\mathbf{C}_v$

7:       **for** $i = 1$ to $\mathbf{J}[k]$ **do**

8:         $\mathbf{V} = $Krylov($\mathbf{L}, \mathbf{U}, \mathbf{M}, \mathbf{B}, , \mathbf{V}, (\mathbf{W}), $**defltol**$, i$)

9:         $\mathbf{W} = $Krylov($\mathbf{U}^T, \mathbf{L}^T, \mathbf{M}^T, \mathbf{C}^T, \mathbf{W}, (\mathbf{V}), $**defltol**$, i$)

10:       **end for**

11:       **if** ($\#columns(\mathbf{V}) \neq \#columns(\mathbf{W})$) **then**

12:         make $\mathbf{V}$ and $\mathbf{W}$ of the same dimension

13:       **end if**

14:    **end for**

15:    $\mathbf{M}_r = \mathbf{W}^T\mathbf{M}\mathbf{V}$, $\mathbf{K}_r = \mathbf{W}^T\mathbf{K}\mathbf{V}$

16:    assign $\mathbf{s}_k \leftarrow -\lambda_k(\mathbf{M}_r, \mathbf{K}_r)$ for $k = 1, \ldots, K$

17: **end while**

18: $[\hat{\mathbf{M}}, \hat{\mathbf{K}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}_p, \hat{\mathbf{C}}_v]$ = transform($\mathbf{M}, \mathbf{K}, \mathbf{B}, \mathbf{C}_p, \mathbf{C}_v, \mathbf{V}, \mathbf{W}$)

19: if the reduced system is unstable reduce dimension of $\mathbf{V}$ and $\mathbf{W}$

Some remarks on Algorithm 5:

$\mathbf{s}_k$, $k = 1, \ldots, K$ are the expansion points. $\mathbf{J}$ is a column vectors with the number of expansions computed at expansion point $\mathbf{s}_k$.

In line 5 the LU decomposition of the matrix $-(\mathbf{s}_k^2\mathbf{M} + \mathbf{K})$ is computed.

In line 8 and 9 the matrices $\mathbf{V}$ and $\mathbf{W}$ are computed by the Arnoldi or the Lanczos algorithm. When the Arnoldi algorithm is used the matrices $\mathbf{V}$ and $\mathbf{W}$ are orthonormal, i.e. $\mathbf{V}^T\mathbf{V} = I$ and $\mathbf{W}^T\mathbf{W} = I$. When the Lanczos algorithm is used the matrices $\mathbf{V}$ and $\mathbf{W}$ are biorthonormal, i.e. $\mathbf{W}^T\mathbf{V} = \Delta$, where Δ is a diagonal matrix.

In line 12 the dimensions of $\mathbf{V}$ and $\mathbf{W}$ were made equal, see remarks on Algorithm 1 for details.

In line 15 the congruence transformation (7) for $\mathbf{M}$ and $\mathbf{K}$ is computed.

In line 16 new expansion points $\mathbf{s}_k$ are chosen from the eigenvalues of $(\mathbf{M}_r, \mathbf{K}_r)$ in a relevant frequency interval computed by a generalized eigenproblem.

In line 18 the congruence transformation (7) is computed.

In step 19 the dimension of the matrices $\mathbf{V}$ and $\mathbf{W}$ is reduced by truncating the last columns, until the reduced system is stable.

Algorithm 6 Rational-Arnoldi method / Rational-Lanczos method for second order systems with proportional damping

Input: system matrices $\mathbf{M}, \mathbf{D}, \mathbf{K}, \mathbf{B}, \mathbf{C}_p, \mathbf{C}_v$; expansion points: $\mathbf{s}_k$, $k = 1, \ldots, K$; tolerance **tol**; vector with numbers
    of expansions $\mathbf{J}$ for every $\mathbf{s}_k$; deflation tolerance **defltol**

Output: reduced system matrices of second order $\hat{\mathbf{M}}, \hat{\mathbf{D}}, \hat{\mathbf{K}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}_p, \hat{\mathbf{C}}_v$

  1: function $[\hat{\mathbf{M}}, \hat{\mathbf{D}}, \hat{\mathbf{K}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}_p, \hat{\mathbf{C}}_v] = \text{RA\_PD}(\mathbf{M}, \mathbf{D}, \mathbf{K}, \mathbf{B}, \mathbf{C}_p, \mathbf{C}_v, \mathbf{s}, \mathbf{J}, \mathbf{r}, \mathbf{defltol})$
        $[\, [\hat{\mathbf{M}}, \hat{\mathbf{D}}, \hat{\mathbf{K}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}_p, \hat{\mathbf{C}}_v] = \text{RL\_PD}(\mathbf{M}, \mathbf{D}, \mathbf{K}, \mathbf{B}, \mathbf{C}_p, \mathbf{C}_v, \mathbf{s}, \mathbf{J}, \mathbf{r}, \mathbf{defltol}) \,]$
  2: **while** relative change in one $\mathbf{s}_k > \textbf{tol}$ **do**
  3:    $\mathbf{V} = [\,], \mathbf{W} = [\,]$
  4:    **for** $k = 1$ to K **do**
  5:      $[\mathbf{L}, \mathbf{U}] = -lu(\mathbf{s}_k^2 \mathbf{M} + \mathbf{s}_k \mathbf{D} + \mathbf{K})$
  6:      **for** $i = 1$ to $\mathbf{J}[k]$ **do**
  7:        $\mathbf{C} = \mathbf{C}_p + \mathbf{s}_k \mathbf{C}_v$
  8:        $\mathbf{V} = \text{Krylov}(\mathbf{L}, \mathbf{U}, \mathbf{M}, \mathbf{B}, \mathbf{V}, (\mathbf{W}), \mathbf{defltol}, i)$
  9:        $\mathbf{W} = \text{Krylov}(\mathbf{U}^T, \mathbf{L}^T, \mathbf{M}^T, \mathbf{C}^T, \mathbf{W}, (\mathbf{V}), \mathbf{defltol}, i)$
10:      **end for**
11:      **if** ($\#columns(\mathbf{V}) \neq \#columns(\mathbf{W})$) **then**
12:        make $\mathbf{V}$ and $\mathbf{W}$ of the same dimension
13:      **end if**
14:    **end for**
15:    $\mathbf{M}_r = \mathbf{W}^T \mathbf{M} \mathbf{V}$, $\mathbf{D}_r = \mathbf{W}^T \mathbf{D} \mathbf{V}$, $\mathbf{K}_r = \mathbf{W}^T \mathbf{K} \mathbf{V}$
16:    assign $\mathbf{s}_k \leftarrow -\lambda_k(\mathbf{M}_r, \mathbf{D}_r, \mathbf{K}_r)$ for $k = 1, \ldots, K$
17: **end while**
18: $[\hat{\mathbf{M}}, \hat{\mathbf{D}}, \hat{\mathbf{K}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}_p, \hat{\mathbf{C}}_v] = \text{transform}(\mathbf{M}, \mathbf{D}, \mathbf{K}, \mathbf{B}, \mathbf{C}_p, \mathbf{C}_v, \mathbf{V}, \mathbf{W})$
19: **if** the reduced system is unstable reduce dimension of $\mathbf{V}$ and $\mathbf{W}$

Some remarks on Algorithm 6:

$\mathbf{s}_k$, $k = 1, \ldots, K$ are the expansion points. $\mathbf{J}$ is a column vector with the number of expansions computed at expansion point $\mathbf{s}_k$.

In line 5 the *LU* decomposition of the matrix $-(\mathbf{s}_k^2 \mathbf{M} + \mathbf{s}_k \mathbf{D} + \mathbf{K})$ is computed.

In line 8 and 9 the matrices $\mathbf{V}$ and $\mathbf{W}$ are computed successively by the Arnoldi or the Lanczos algorithm. When the Arnoldi algorithm is used the matrices $\mathbf{V}$ and $\mathbf{W}$ are orthonormal, i.e. $\mathbf{V}^T \mathbf{V} = I$ and $\mathbf{W}^T \mathbf{W} = I$. When the Lanczos algorithm is used the matrices $\mathbf{V}$ and $\mathbf{W}$ are biorthonormal, i.e. $\mathbf{W}^T \mathbf{V} = \Delta$, where Δ is a diagonal matrix.

In line 12 the dimensions of $\mathbf{V}$ and $\mathbf{W}$ were made equal, see remarks on Algorithm 1 for details.

In line 15 the congruence transformation (7) for $\mathbf{M}$, $\mathbf{D}$ and $\mathbf{K}$ is computed.

In line 16 new expansion points $\mathbf{s}_k$ are chosen from the eigenvalues of $(\mathbf{M}_r, \mathbf{D}_r, \mathbf{K}_r)$ in a relevant frequency interval computed by a generalized eigenproblem.

In step 19 the dimension of the matrices $\mathbf{V}$ and $\mathbf{W}$ is reduced by truncating the last columns, until the reduced system is stable.

4 Numerical Results

To test the approximation results of the different methods a single feed drive with ball screw drive and linear guide was designed using the CAD environment NASTRAN[©]. The structure of that part of the machine tool is shown in Figure 2. The structure is embedded in a control loop. As input the torque at the motor spindle was defined. The angle and the velocity of the motor spindle and the position and the velocity of the slide were defined as outputs.

The test model is of order $n = 738$, i.e. it has 738 degrees of freedom (DOF's), it has one input ($p = 1$) and four outputs ($m = 4$). The parameters for the proportional damping matrix were chosen as $\alpha = 0.02$ and $\beta = \alpha/100$. The algorithms were implemented in MATLAB[1] version 7.1 (R14). The computations were performed on a AMD Athlon(tm) 64 X2 Dual Core Processor 4400+ and 2 GB RAM.

All results reported here were obtained for first order systems of initial reduced dimension 20 and for second order systems of initial reduced dimension 10. Besides the Krylov subspace methods, a second order modal reduced system generated by NASTRAN of dimension 10 (modal_2o) and a first order BTA reduced model of dimension 20 (BTA) were used to compare the approximation results of various reduction methods. The BTA reduced model was generated by Benner and Saak (see their contribution) using an efficient application of BTA [3] to the specific

---

[1]MATLAB is a trademark of The MathWorks, Inc.
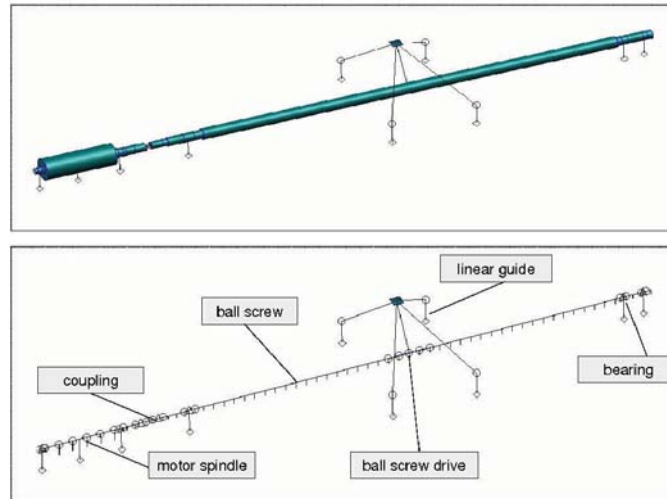
Figure 2: FE structure of the machine part: top: geometry of the test model, bottom: without the geometry.

problem considered here. The method is applied to a second order system resulting in a first order reduced one (unlike the other second order methods).

In Table 1 the input parameters and the computation times to reduce the first order system using different Krylov subspace methods are given. Table 2 gives the same information as Table 1, but for the corresponding second order system. In order to compare the different methods the approximation of the original transfer function and the time

	expansion points s_0 or s_i	expansions per point k or k_i	tolerance **tol**	time to reduce the system [s]
PVA	0	20	10^{-5}	12.2
PVL	0	20	10^{-5}	11.3
RA	$2\pi(0, \pm 125i, \pm 250i)^T$	2	10^{-5}	20.1
RL	$2\pi(0, \pm 125i, \pm 250i)^T$	2	10^{-5}	19.6

Table 1: Chosen initial expansion points, expansions per point, tolerance **tol** and computation time to reduce the first order systems.

	expansion points s_i	expansions per point k_i	tolerance **tol**	time to reduce the system [s]
RA	$2\pi(0, \pm 125i, \pm 250i)^T$	1	10^{-5}	8.8
RL	$2\pi(0, \pm 125i, \pm 250i)^T$	1	10^{-5}	8.6
RA_PD	$2\pi(0, \pm 125i, \pm 250i)^T$	1	10^{-5}	8.1
RL_PD	$2\pi(0, \pm 125i, \pm 250i)^T$	1	10^{-5}	7.9

Table 2: Chosen initial expansion points, expansion per point, tolerance **tol** and computation time to reduce the second order systems.

response by the reduced systems were analyzed.
To assess the quality of the reduced systems the following errors were used:

- The absolute time response error from the input to the j-th output of a reduced system was computed by

$$\varepsilon_{t,abs}(t) = |y_j(t) - \hat{y}_j(t)|.$$

Here $\varepsilon_{t,abs}(t)$ is the absolute error, $y_j(t)$ is the j-th output of the original system and $\hat{y}_j(t)$ is the j-th output of the reduced system at time t.

- The relative error of the transfer function from the j-th input to the k-th output of a reduced system was computed by

$$\varepsilon_{\omega,rel}(\omega) = \frac{|H_{k,j}(2\pi \omega i) - \hat{H}_{k,j}(2\pi \omega i)|}{|H_{k,j}(2\pi \omega i)|}.$$

Here $\varepsilon_{\omega,rel}(\omega)$ is the relative error, $H_{k,j}$ and $\hat{H}_{k,j}$ are the transfer functions from the j-th input to the k-th output of the original system respective of the reduced system.

The worst approximation results were observed for the input-output behavior from the input to the second output. Only this information is given in the following figures.

4.1 Approximation of the transfer function

In Figure 3 the transfer functions of the original and the reduced systems are given, while in Figure 4 the relative approximation errors of the transfer functions are given. Here the relevant frequency interval is from from 0 to 1000 Hz because this frequency range is most important to simulate the behavior of mechanical structures under consideration. In order to compare the different methods for first order systems the second order modal reduced system (modal_2o) was transformed into a first order system of dimension 20 (modal_1o).
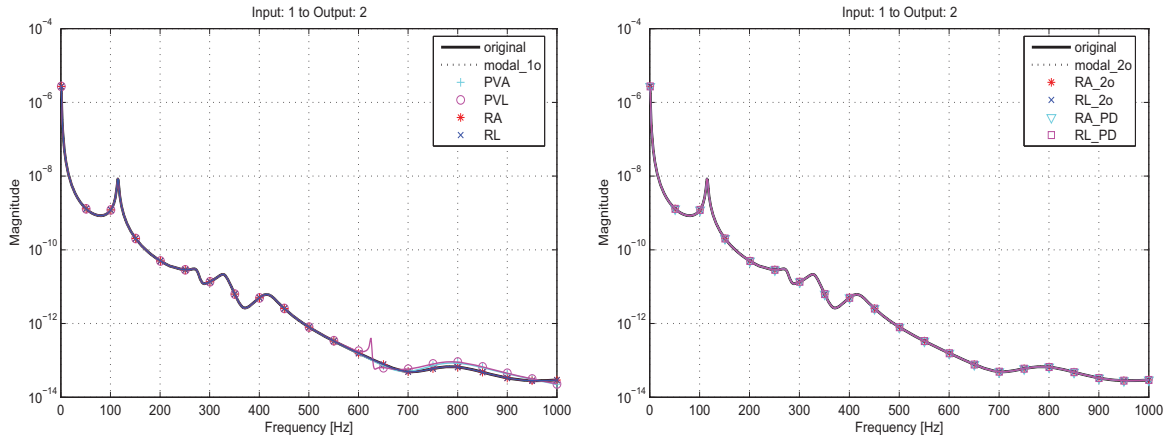
Figure 3: The second output transfer function of the original and the reduced systems: **left:** Transfer functions of the original first order system and the reduced systems obtained with methods to reduce first order systems. **right:** Transfer functions of the original second order system and the reduced systems obtained with methods to reduce second order systems.
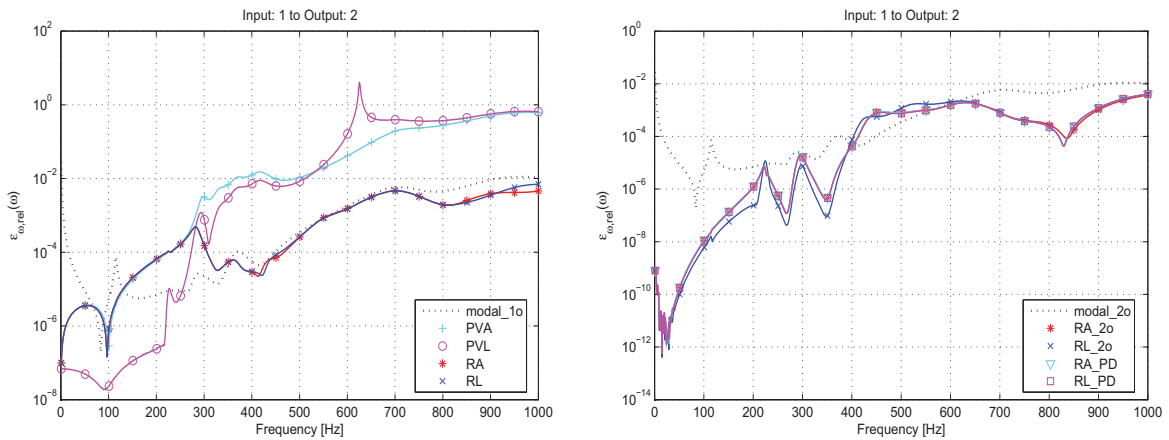
Figure 4: Relative error $\varepsilon_{\omega,rel}(\omega)$ of the reduced systems: **left:** Approximation errors of the reduced systems obtained with methods to reduce first order systems. **right:** Approximation errors of the reduced systems obtained with methods to reduce second order systems.

The results for the first order system is displayed on the left in each of the two figures. The approximation results obtained with the PVA and the PVL method get worse for frequencies far away from the expansion point 0. The maximum relative error of these methods is greater than 1. The Krylov subspace methods Rational-Arnoldi and Rational-Lanczos yield reduced systems with an approximation error similar to that of the modal reduced systems of order 20 (modal_20) with a maximum relative error of 10^{-2}. The reduction with Rational Krylov methods which use more than on expansion point yields better approximations in a wider range of frequencies.

The results for the second order system is displayed on the right in each of the two figures. All Krylov subspace methods to reduce second order systems considered here are Rational Krylov methods, that is, they use more than one expansion point. They yield good approximations in a wide range of frequencies. All four methods implemented here have almost the same approximation errors.

For the entire frequency range considered here the reduced second order systems have a better relative error than the reduced first order systems.

4.2   Approximation of the time response

To analyze the approximation abilities of the response behavior in time the reduced models were embedded into a control loop designed with MATLAB/SIMULINK. The input signal was an impulse from 0 to 1 in $t = 0$ s. In Figure 5 the time response of the reduced systems embedded in the control loop are shown. To compute the simulation in a reasonable time the time responses of the reduced systems were compared with the time response of a modal reduced system of order 472 which includes all modes of the original system. MATLAB/SIMULINK deals only with first order systems, so the second order reduced systems had to be transformed into first order systems by (3).

In Figure 6 the absolute errors of the time responses after embedding the reduced systems into the control loop are displayed. The Krylov subspace methods to reduce first order systems yield good approximations of the time
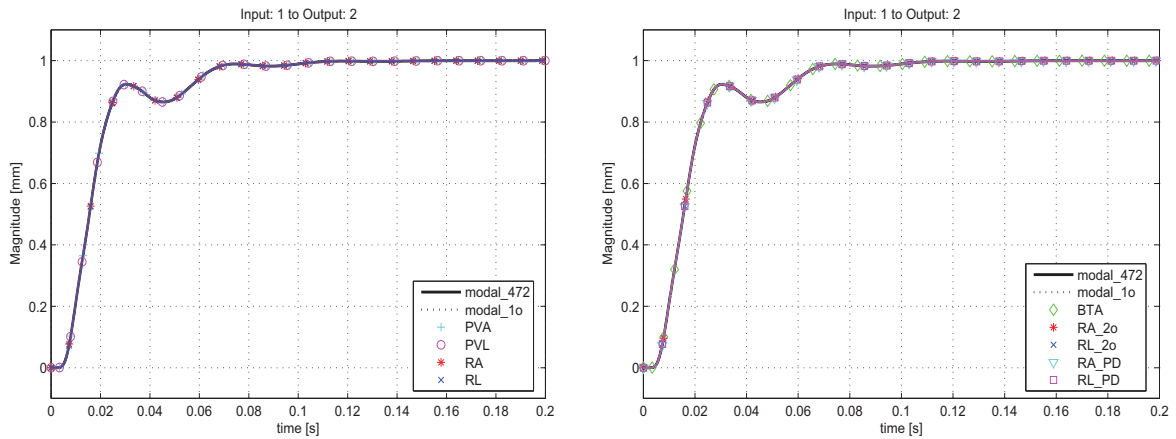
Figure 5: Time responses of the reduced systems, here the input-output behavior from the input to the second output is displayed: **left:** Time response of the modal reduced system with dimension 472 and the reduced systems obtained with methods to reduce first order systems. **right:** Time response of the modal reduced system with dimension 472 and the reduced systems obtained with methods to reduce second order systems.
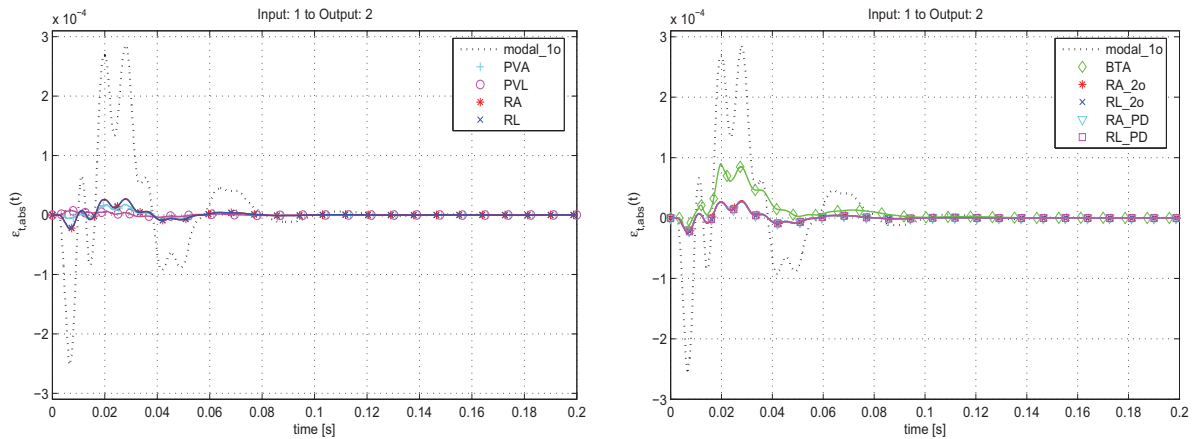
Figure 6: Absolute error $\varepsilon_{t,abs}(t)$ of the reduced systems: **left:** Approximation errors of the reduced systems obtained with methods to reduce first order systems. **right:** Approximation errors of the reduced systems obtained with methods to reduce second order systems.

response. The maximal absolute error is $2 \cdot 10^{-5}$ in contrast to $3 \cdot 10^{-4}$ after modal reduction of the original system to a system of order 20. The Krylov subspace methods to reduce second order systems yield better approximations of the time response of the original system compared to the modal and via BTA reduced systems of order 20. The maximal absolute error is $2 \cdot 10^{-5}$ in contrast to $3 \cdot 10^{-4}$ and $9 \cdot 10^{-5}$ obtained with the modal resp. the BTA reduced model of order 20.

5   Conclusions

Several Krylov methods to reduce first or second order systems were implemented. The reduction algorithms are based on IRKA and were adapted for the special systems of FE models occurring in the simulation of machine tools. For the problem at hand the methods with more than one expansion point should be used, because they give better results at a wider range of frequencies. In time response we obtained similar approximation results

whether by using reduction methods for first order systems or second order systems. For the entire frequency range considered here the reduced systems obtained with second order methods have a better relative error than the reduced systems obtained with first order methods. The second order form of a system is half as large as the first order form because by the transformation process the dimension of the system doubles. Therefore the reduction of the second order system may be more effective if the dimension of the original model is very large. So the methods for reducing second order systems are to prefer for problems with very large system matrices.

Krylov subspace methods yield reduced models approximating the original transfer function and time response quite well. Reduced systems obtained via Krylov subspace methods yield slightly better approximation results than systems reduced with BTA.

Krylov subspace methods are suitable for the reduction of structural mechanical FE models obtained by CAD environments like NASTRAN. With these methods reduced models are obtained in a very effective way which have very similar approximation abilities like modal or BTA reduced models.

6 Acknowledgements

7 References

[1] A.C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. SIAM, Philadelphia, 2005.

[2] R.R. Craig, M.C.C. Bampton. *Coupling of Substructures for Dynamic Analysis*. AIAA Journal, 1968, vol. 6, pp. 1313-1319.

[3] P. Benner and J. Saak. *Efficient solution of large scale Lyapunov and Riccati equations arising in model order reduction problems*. Submitted for publication in: Proceedings in Applied Mathematics and Mechanics, Wiley I nterScience, 2008.

[4] P. Feldmann, R. W. Freund. *Efficient linear circuit analysis by Pad*é *approximation via the Lanczos process*. IEEE Trans. Computer-Aided Design, 1995, vol. 14, pp. 639- 649.

[5] E. J. Grimme. *Krylov projection methods for model reduction*. Ph.D. dissertation, University of Illinois at Urbana-Champaign, 1997.

[6] S. Gugercin, A. C. Antoulas, C. Beattie. $\mathcal{H}_2$*-Model Reduction For Large-Scale Linear Systems*. SIAM Journal on Matrix Analysis and Applications, 2008, vol. 30(2), pp. 609-638.

[7] S. Gugercin, C. Beattie. *Krylov-based model reduction of second-order systems with proportional damping*. Proceedings of the 44th IEEE Conference on Decision and Control, 2005, pp. 5905-5910.

[8] P. Feldmann, R. W. Freund. *Efficient linear circuit analysis by Padé approximation via the Lanczos process*. IEEE Trans. Computer-Aided Design, 1995 vol. 14, pp. 639-649.

[9] K. Gallivan, E.J. Grimme, P. van Dooren. *A rational Lanczos algorithm for model reduction*. Numerical Algorithms, 1996, vol. 12, pp. 33-63.

[10] M. Odabasioglu, M. Celik. *Passive Reduced-Order Interconnect Macromodeling Algorithm*. IEEE Trans. Computer-Aided Design, 1999, pp. 645-654.

[11] B. Salimbahrami. *Structure Preserving Order Reduction of Large Scale Second Order Models*. PhD thesis, Technische Universität München, 2005.

[12] G.W. Stewart. *Matrix Algorithms, Volume II: Eigensystems*. SIAM, Philadelphia, 2001.