Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

Control and Automation Group
Department of Electrical Engineering
Indian Institute of Technology-Delhi

Automatic Adaptive Sampling in Parametric Model Order Reduction by Matrix Interpolation

Maria Cruz Varona*, Mashuq-un-Nabi[‡], Boris Lohmann[§]

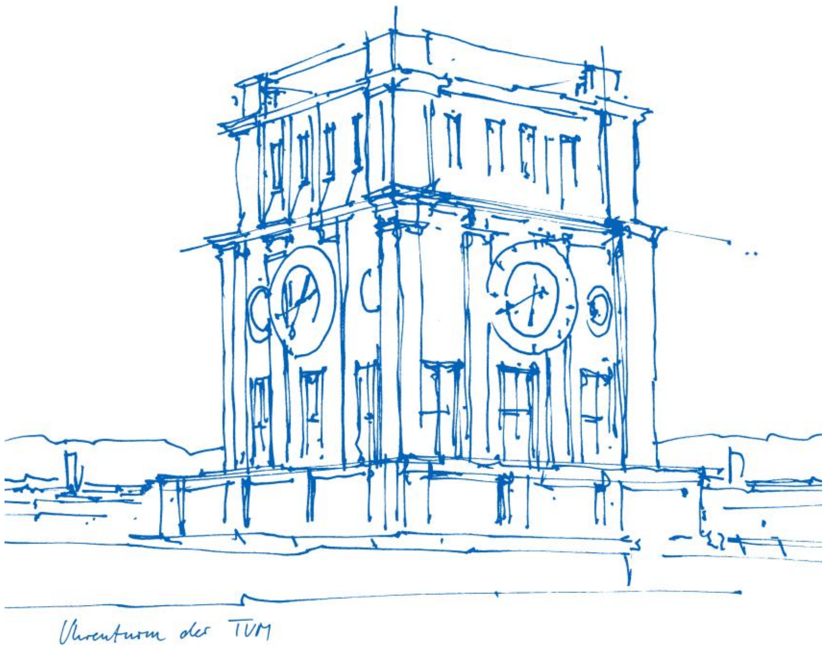
[*§]Chair of Automatic Control,

Technical University of Munich, Germany

[‡]Department of Electrical Engineering,

Indian Institute of Technology Delhi, India

July 4th, AIM 2017



Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

Control and Automation Group
Department of Electrical Engineering
Indian Institute of Technology-Delhi

Motivation

Large-scale full order model (FOM)

$$\mathbf{E}\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x}$$

$$\det(\mathbf{E}) \neq 0$$

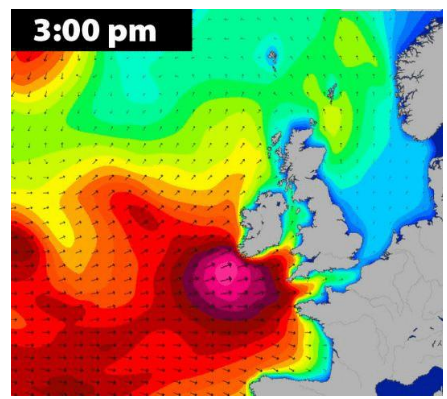
MOR

Reduced order model (ROM)

$$\mathbf{E}_r\dot{\mathbf{x}}_r = \mathbf{A}_r\mathbf{x}_r + \mathbf{B}_r\mathbf{u}$$

$$\mathbf{y}_r = \mathbf{C}_r\mathbf{x}_r$$

$$\mathbf{x}_r \in \mathbb{R}^r, \; r \ll n$$



n: 10^3 — 10^4 — 10^5 — 10^6

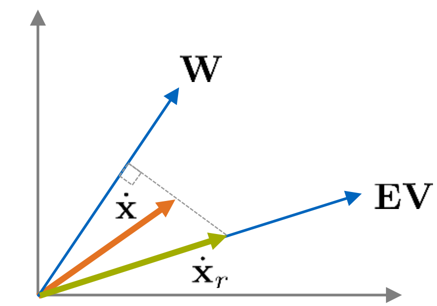
Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

Control and Automation Group
Department of Electrical Engineering
Indian Institute of Technology-Delhi

Projective Model Order Reduction

Approximation of the state vector:

$$\mathbf{x} = \mathbf{V}\mathbf{x}_r + \mathbf{e}, \qquad \mathbf{V} \in \mathbb{R}^{n\times r}$$



Petrov-Galerkin projection:

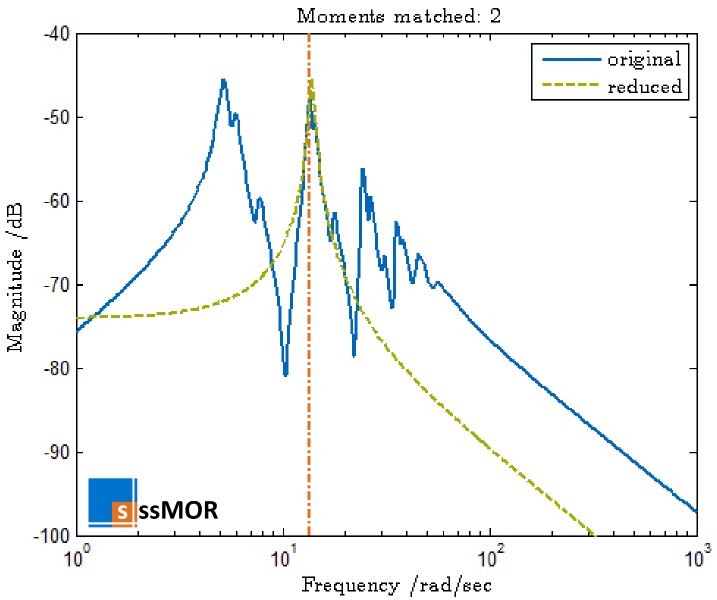
$$\overbrace{\mathbf{W}^T\mathbf{E}\mathbf{V}}^{\mathbf{E}_r}\dot{\mathbf{x}}_r = \overbrace{\mathbf{W}^T\mathbf{A}\mathbf{V}}^{\mathbf{A}_r}\mathbf{x}_r + \overbrace{\mathbf{W}^T\mathbf{B}}^{\mathbf{B}_r}\mathbf{u}$$

$$\mathbf{y} \approx \mathbf{y}_r = \underbrace{\mathbf{C}\mathbf{V}}_{\mathbf{C}_r}\mathbf{x}_r$$

Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

Control and Automation Group
Department of Electrical Engineering
Indian Institute of Technology-Delhi

Rational Interpolation by Krylov subspace methods



Moments of a transfer function

$$\mathbf{G}(s) = \mathbf{C}(s\mathbf{E}-\mathbf{A})^{-1}\mathbf{B}$$
$$= \mathbf{G}(\Delta s + s_0) = -\sum_{i=0}^{\infty} \mathbf{M}_i(s_0)(s-s_0)^i$$

$\mathbf{M}_i(s_0)$: i-th moment around s_0

Krylov subspace:

$$\mathcal{K}_r(\mathbf{M},\mathbf{v}) = \left[\mathbf{v}, \mathbf{M}\,\mathbf{v}, \mathbf{M}^2\,\mathbf{v}, \cdots, \mathbf{M}^{r-1}\mathbf{v}\right]$$

Moment Matching by Rational Krylov (RK) subspaces

Bases for input and output Krylov-subspaces:

$$\mathrm{Im}(\mathbf{V}) = \left[\mathbf{A}_{s_0}^{-1}\mathbf{B},\quad \mathbf{A}_{s_0}^{-1}\mathbf{E}\mathbf{A}_{s_0}^{-1}\mathbf{B},\quad \ldots,\quad (\mathbf{A}_{s_0}^{-1}\mathbf{E})^{r-1}\mathbf{A}_{s_0}^{-1}\mathbf{B}\right]$$
$$\mathrm{Im}(\mathbf{W}) = \left[\mathbf{A}_{s_0}^{-T}\mathbf{C}^T,\quad \mathbf{A}_{s_0}^{-T}\mathbf{E}^T\mathbf{A}_{s_0}^{-T}\mathbf{C}^T,\quad \ldots,\quad (\mathbf{A}_{s_0}^{-T}\mathbf{E}^T)^{r-1}\mathbf{A}_{s_0}^{-T}\mathbf{C}^T\right]$$

$$\Rightarrow \quad \boxed{\mathbf{M}_i(s_0) = \mathbf{M}_{r,i}(s_0)}$$
$$\text{for } i = 0, \ldots, 2r$$

Moments from full and reduced order model around certain shifts **match!**

Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

Control and Automation Group
Department of Electrical Engineering
Indian Institute of Technology-Delhi

Parametric Model Order Reduction (pMOR)

Large-scale parametric model

$$\mathbf{E}(\mathbf{p})\,\dot{\mathbf{x}} = \mathbf{A}(\mathbf{p})\,\mathbf{x} + \mathbf{B}(\mathbf{p})\,\mathbf{u}$$
$$\mathbf{y} = \mathbf{C}(\mathbf{p})\,\mathbf{x}$$

$$\boxed{\mathbf{p} \in \mathcal{D} \subset \mathbb{R}^d}$$

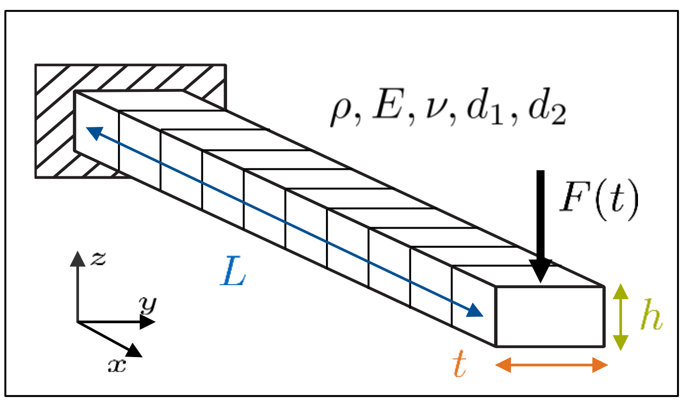
pMOR

Reduced order parametric model

$$\mathbf{E}_r(\mathbf{p})\,\dot{\mathbf{x}}_r = \mathbf{A}_r(\mathbf{p})\,\mathbf{x}_r + \mathbf{B}_r(\mathbf{p})\,\mathbf{u}$$
$$\mathbf{y}_r = \mathbf{C}_r(\mathbf{p})\,\mathbf{x}_r$$

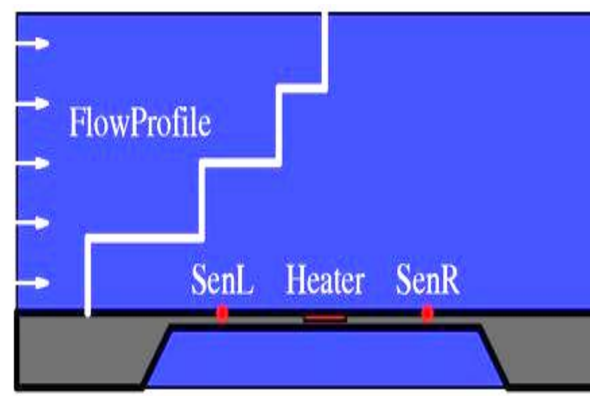
$$\mathbf{x}_r \in \mathbb{R}^r,\ r \ll n$$

- Linear dynamic systems with design parameters (e.g. material / geometry parameters,…)
- **Goal:** numerically efficient reduction with **preservation of the parameter dependency**



Timoshenko beam

Solar panels



Flow sensing anemometer

pMOR by Matrix Interpolation – Main Idea

$\mathbf{p}_1$ $\quad$ $\mathbf{p}^{\text{int}}$? $\quad$ $\mathbf{p}_2$ $\quad$ $\mathbf{p}$

Model at $\mathbf{p}_1$	Model at $\mathbf{p}^{\text{int}}$	Model at $\mathbf{p}_2$
$\mathbf{V}_1, \mathbf{W}_1$	$\mathbf{V}_{\mathbf{p}^{\text{int}}}, \mathbf{W}_{\mathbf{p}^{\text{int}}}$	$\mathbf{V}_2, \mathbf{W}_2$
Reduced model at $\mathbf{p}_1$	Reduced model at $\mathbf{p}^{\text{int}}$	Reduced model at $\mathbf{p}_2$

Interpolation of
reduced matrices

Reduced model at $\mathbf{p}^{\text{int}}$ from reduced models at sample points $\mathbf{p}_1$ & $\mathbf{p}_2$

Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

Control and Automation Group
Department of Electrical Engineering
Indian Institute of Technology-Delhi

pMOR by Matrix Interpolation – Procedure

[Panzer et al. '10]

$\mathbf{p}_1$ $\quad$ $\mathbf{p}^{\mathrm{int}}$? $\quad$ $\mathbf{p}_2$ $\quad$ $\mathbf{p}$

$$\mathbf{E}_1\dot{\mathbf{x}} = \mathbf{A}_1\mathbf{x} + \mathbf{B}_1\mathbf{u}$$
$$\mathbf{y} = \mathbf{C}_1\mathbf{x}$$

$\mathbf{V}_1, \mathbf{W}_1$

$$\mathbf{E}_{r,1}\dot{\mathbf{x}}_{r,1} = \mathbf{A}_{r,1}\mathbf{x}_{r,1} + \mathbf{B}_{r,1}\mathbf{u}$$
$$\mathbf{y}_{r,1} = \mathbf{C}_{r,1}\mathbf{x}_{r,1}$$

$$\mathbf{E}_2\dot{\mathbf{x}} = \mathbf{A}_2\mathbf{x} + \mathbf{B}_2\mathbf{u}$$
$$\mathbf{y} = \mathbf{C}_2\mathbf{x}$$

$\mathbf{V}_2, \mathbf{W}_2$

$$\mathbf{E}_{r,2}\dot{\mathbf{x}}_{r,2} = \mathbf{A}_{r,2}\mathbf{x}_{r,2} + \mathbf{B}_{r,2}\mathbf{u}$$
$$\mathbf{y}_{r,2} = \mathbf{C}_{r,2}\mathbf{x}_{r,2}$$

direct interpolation
not meaningful

1.) Individual reduction

$$\mathbf{E}_{r,i}\dot{\mathbf{x}}_{r,i}(t) = \mathbf{A}_{r,i}\mathbf{x}_{r,i}(t) + \mathbf{B}_{r,i}\mathbf{u}(t) \qquad \mathbf{E}_{r,i} = \mathbf{W}_i^T\mathbf{E}_i\mathbf{V}_i, \quad \mathbf{A}_{r,i} = \mathbf{W}_i^T\mathbf{A}_i\mathbf{V}_i$$
$$\mathbf{y}_{r,i}(t) = \mathbf{C}_{r,i}\mathbf{x}_{r,i}(t) \qquad \mathbf{B}_{r,i} = \mathbf{W}_i^T\mathbf{B}_i, \quad \mathbf{C}_{r,i} = \mathbf{C}_i\mathbf{V}_i$$

$$\mathbf{p}_i, \; i = 1, \ldots, K$$
$$\mathbf{V}_i := \mathbf{V}(\mathbf{p}_i)$$
$$\mathbf{W}_i := \mathbf{W}(\mathbf{p}_i)$$

pMOR by Matrix Interpolation – Procedure

[Panzer et al. '10]

$\mathbf{p}_i, \; i = 1, \dots, K$

$\mathbf{V}_i := \mathbf{V}(\mathbf{p}_i)$

$\mathbf{W}_i := \mathbf{W}(\mathbf{p}_i)$

1.) Individual reduction

$$\mathbf{E}_{r,i}\dot{\mathbf{x}}_{r,i}(t) = \mathbf{A}_{r,i}\mathbf{x}_{r,i}(t) + \mathbf{B}_{r,i}\mathbf{u}(t) \qquad \mathbf{E}_{r,i} = \mathbf{W}_i^T\mathbf{E}_i\mathbf{V}_i, \quad \mathbf{A}_{r,i} = \mathbf{W}_i^T\mathbf{A}_i\mathbf{V}_i$$

$$\mathbf{y}_{r,i}(t) = \mathbf{C}_{r,i}\mathbf{x}_{r,i}(t) \qquad \mathbf{B}_{r,i} = \mathbf{W}_i^T\mathbf{B}_i, \quad \mathbf{C}_{r,i} = \mathbf{C}_i\mathbf{V}_i$$

2.) Transformation to generalized coordinates

$$\overbrace{\mathbf{M}_i^T\mathbf{E}_{r,i}\mathbf{T}_i}^{\mathbf{E}_{r,i}^*}\dot{\mathbf{x}}_{r,i}^*(t) = \overbrace{\mathbf{M}_i^T\mathbf{A}_{r,i}\mathbf{T}_i}^{\mathbf{A}_{r,i}^*}\mathbf{x}_{r,i}^*(t) + \overbrace{\mathbf{M}_i^T\mathbf{B}_{r,i}}^{\mathbf{B}_{r,i}^*}\mathbf{u}(t)$$

$$\mathbf{y}_{r,i}(t) = \underbrace{\mathbf{C}_{r,i}\mathbf{T}_i}_{\mathbf{C}_{r,i}^*}\mathbf{x}_{r,i}^*(t)$$

$$\mathbf{T}_i = (\mathbf{R}^T\mathbf{V}_i)^{-1}$$

$$\mathbf{M}_i = (\mathbf{R}^T\mathbf{W}_i)^{-1}$$

$\mathbf{x}_{r,i} = \mathbf{T}_i\mathbf{x}_{r,i}^*$

$\mathbf{V}_{\text{all}} = [\mathbf{V}_1, \dots, \mathbf{V}_K]$

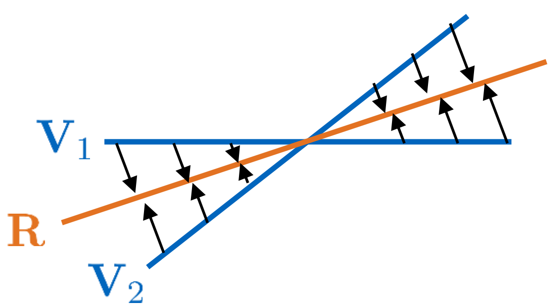
$\mathbf{V}_{\text{all}} \overset{\text{SVD}}{=} \mathbf{U\Sigma N}^T$

$\mathbf{R} = \mathbf{U}(:, 1:r)$

How to choose $\mathbf{T}_i$ and $\mathbf{M}_i$?

Goal: Adjustment of the local bases $\mathbf{V}_i$ to $\mathbf{V}_i^* = \mathbf{V}_i\mathbf{T}_i$, in order to make the gen. coordinates $\mathbf{x}_{r,i}^*$ compatible w.r.t. a reference subspace $\mathbf{R}$.

Dual procedure for the local bases $\mathbf{W}_i$



High correlation $\mathbf{V}_i^* \longleftrightarrow \mathbf{R}$:

$\mathbf{T}_i^T\mathbf{V}_i^T\mathbf{R} \overset{!}{=} \mathbf{I}$

pMOR by Matrix Interpolation – Procedure

[Panzer et al. '10]

1.) Individual reduction

$$\mathbf{E}_{r,i}\dot{\mathbf{x}}_{r,i}(t) = \mathbf{A}_{r,i}\mathbf{x}_{r,i}(t) + \mathbf{B}_{r,i}\mathbf{u}(t) \qquad \mathbf{E}_{r,i} = \mathbf{W}_i^T\mathbf{E}_i\mathbf{V}_i, \quad \mathbf{A}_{r,i} = \mathbf{W}_i^T\mathbf{A}_i\mathbf{V}_i$$

$$\mathbf{y}_{r,i}(t) = \mathbf{C}_{r,i}\mathbf{x}_{r,i}(t) \qquad \mathbf{B}_{r,i} = \mathbf{W}_i^T\mathbf{B}_i, \quad \mathbf{C}_{r,i} = \mathbf{C}_i\mathbf{V}_i$$

$$\mathbf{p}_i,\ i = 1,\ldots,K$$
$$\mathbf{V}_i := \mathbf{V}(\mathbf{p}_i)$$
$$\mathbf{W}_i := \mathbf{W}(\mathbf{p}_i)$$

2.) Transformation to generalized coordinates

$$\overbrace{\mathbf{M}_i^T\mathbf{E}_{r,i}\mathbf{T}_i}^{\mathbf{E}^*_{r,i}}\dot{\mathbf{x}}^*_{r,i}(t) = \overbrace{\mathbf{M}_i^T\mathbf{A}_{r,i}\mathbf{T}_i}^{\mathbf{A}^*_{r,i}}\mathbf{x}^*_{r,i}(t) + \overbrace{\mathbf{M}_i^T\mathbf{B}_{r,i}}^{\mathbf{B}^*_{r,i}}\mathbf{u}(t)$$

$$\mathbf{y}_{r,i}(t) = \underbrace{\mathbf{C}_{r,i}\mathbf{T}_i}_{\mathbf{C}^*_{r,i}}\mathbf{x}^*_{r,i}(t)$$

$$\mathbf{x}_{r,i} = \mathbf{T}_i\mathbf{x}^*_{r,i}$$

$$\mathbf{T}_i = (\mathbf{R}^T\mathbf{V}_i)^{-1}$$
$$\mathbf{M}_i = (\mathbf{R}^T\mathbf{W}_i)^{-1}$$

$$\mathbf{V}_{\text{all}} = [\mathbf{V}_1, \ldots, \mathbf{V}_K]$$
$$\mathbf{V}_{\text{all}} \overset{\text{SVD}}{=} \mathbf{U}\boldsymbol{\Sigma}\mathbf{N}^T$$
$$\mathbf{R} = \mathbf{U}(:, 1:r)$$

3.) Interpolation

$$\mathbf{E}^*_r(\mathbf{p}^{\text{int}}) = \sum_{i=1}^K \omega_i(\mathbf{p})\mathbf{E}^*_{r,i}, \quad \mathbf{A}^*_r(\mathbf{p}^{\text{int}}) = \sum_{i=1}^K \omega_i(\mathbf{p}^{\text{int}})\mathbf{A}^*_{r,i}$$

$$\mathbf{B}^*_r(\mathbf{p}^{\text{int}}) = \sum_{i=1}^K \omega_i(\mathbf{p})\mathbf{B}^*_{r,i}, \quad \mathbf{C}^*_r(\mathbf{p}^{\text{int}}) = \sum_{i=1}^K \omega_i(\mathbf{p}^{\text{int}})\mathbf{C}^*_{r,i}$$

$$\sum_{i=1}^K \omega_i(\mathbf{p}^{\text{int}}) = 1$$

But: how to choose the sample points??

Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

Control and Automation Group
Department of Electrical Engineering
Indian Institute of Technology-Delhi

Adaptive Sampling

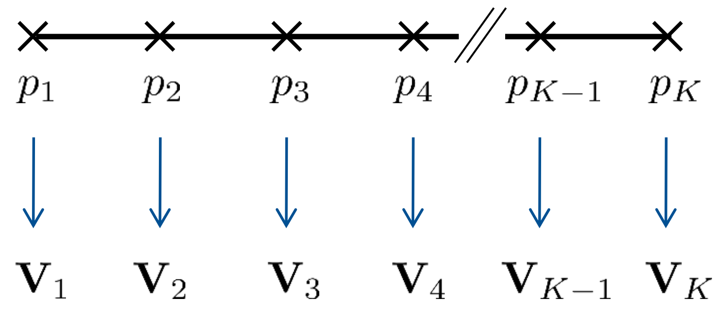
Requirements:

- Parametric space should be adequately sampled
- Avoid undersampling and oversampling
- More parameter samples should be placed in highly sensitive zones

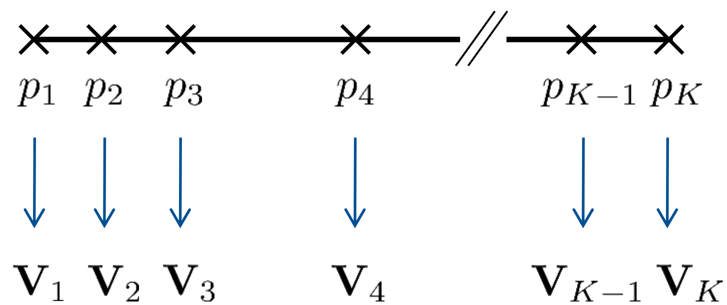
Quantification of parametric sensitivity:

- System-theoretic measure that quantifies the parametric sensitivity is needed in order to guide the adaptive refinement
- Adaptive sampling using angle between subspaces

Uniform Sampling:

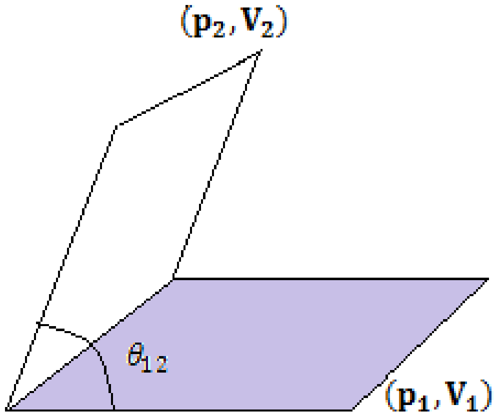


Adaptive Sampling:



Adaptive Sampling via subspace angles

Concept of subspace angles:



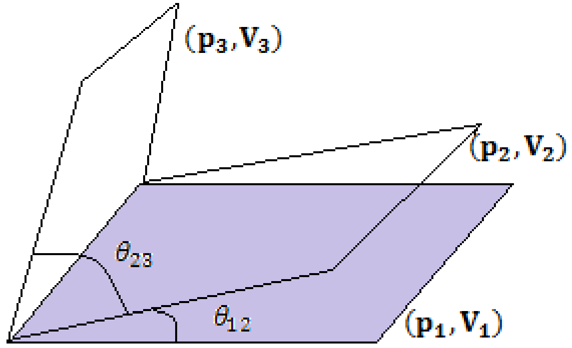
- $\mathbf{V}_1$ and $\mathbf{V}_2$ are orthonormal bases for the subspaces $\mathcal{V}_1$ and $\mathcal{V}_2$
- The largest angle between the subspaces can be determined by

$$\boxed{\theta_{12} = \arcsin\left(\sqrt{1-\sigma_r^2}\right) = \arccos(\sigma_r)}$$

σ_r : smallest singular value of $\mathbf{V}_1^T \mathbf{V}_2$

Usage for adaptive grid refinement:

- The larger the subspace angle, the more different are the projection matrices, and thus:
  - the higher the parametric sensitivity
  - and the more sample points can be introduced in the respective sub-span



Automatic Adaptive Sampling: Pseudo-Code

1) Input $\theta_{\max}$
2) Divide the entire parameter range into a uniform grid, calling it $p_1, p_2, \cdots, p_K$
3) **While** all $l_{i,i+1} > 1$ **do**
   a) Calculate the projection matrices $\mathbf{V}_1, \mathbf{V}_2, \cdots, \mathbf{V}_K$ corresponding to each of these values $p_1, p_2, \cdots, p_K$
   b) Compute subspace angles $\theta_{12}, \theta_{23}, \cdots, \theta_{K-1,K}$ between these $\mathbf{V}_i$´s, each taken pairwise
   c) Calculate

$$l_{12} = \left\lceil \frac{\theta_{12}}{\theta_{\max}} \right\rceil, l_{23} = \left\lceil \frac{\theta_{23}}{\theta_{\max}} \right\rceil, \cdots, l_{K-1,K} = \left\lceil \frac{\theta_{K-1,K}}{\theta_{\max}} \right\rceil$$

   d) Divide the interval between p_1 and p_2 into l_{12} further intervals. Likewise, do the same for all the other intervals.
   e) Obtain new grid points $p_1, p_2, \cdots, p_N$, whereas $N > K$

End While

Local reduction at sample points possible using any preferred MOR technique

```
theta(i) = subspace(Vp{i},Vp{i+1})
```

Quantitative indicator of how many pieces each parameter interval is to be further broken

Stopping criterion:
1. All ratios are equal to 1
2. Specified maximum number of samples points reached

Next iteration: local reduction, etc. <u>only</u> at points that got added in the last while-loop iteration (efficient!)

Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

Control and Automation Group
Department of Electrical Engineering
Indian Institute of Technology-Delhi

Numerical example: Timoshenko Beam

- Finite element 3D model of a Timoshenko beam
- Parameter is the length of the beam: $p \equiv L$
- One-sided Krylov reduction with shifts at $s_0 = 0$
- $\theta_{\max} = 10^{\circ}$ chosen

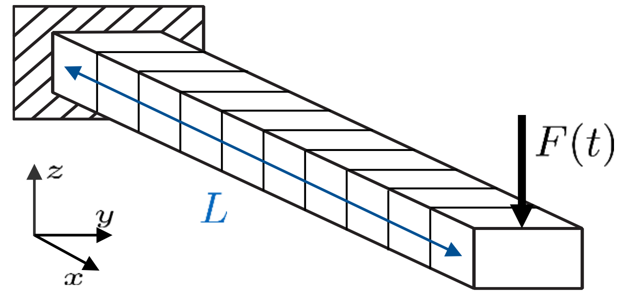


Table 1: Sample points p_i, subspace angles $\theta_{i,i+1}$ and ratios $l_{i,i+1}$

iter 1	p_i[m]	0.5								1.5				2.5		3.5		4.5		5.5
	$\theta_{i,i+1}[^{\circ}]$		25.79								14.20				8.61		7.06		5.73	
	$l_{i,i+1}$		3								2				1		1		1	
iter 2	p_i[m]	0.5				0.833		1.167		1.5		2		2.5		3.5		4.5		5.5
	$\theta_{i,i+1}[^{\circ}]$		10.15				8.59		7.05		8.18		6.03		8.61		7.06		5.73	
	$l_{i,i+1}$		2				1		1		1		1		1		1		1	
iter 3	p_i[m]	0.5		0.667		0.833		1.167		1.5		2		2.5		3.5		4.5		5.5
	$\theta_{i,i+1}[^{\circ}]$		5.26		4.89		8.59		7.05		8.18		6.03		8.61		7.06		5.73	
	$l_{i,i+1}$		1		1		1		1		1		1		1		1		1	

Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

Control and Automation Group
Department of Electrical Engineering
Indian Institute of Technology-Delhi

Numerical example: Timoshenko Beam

Initial uniform grid with $K = 6$

Adaptive Sampling Scheme

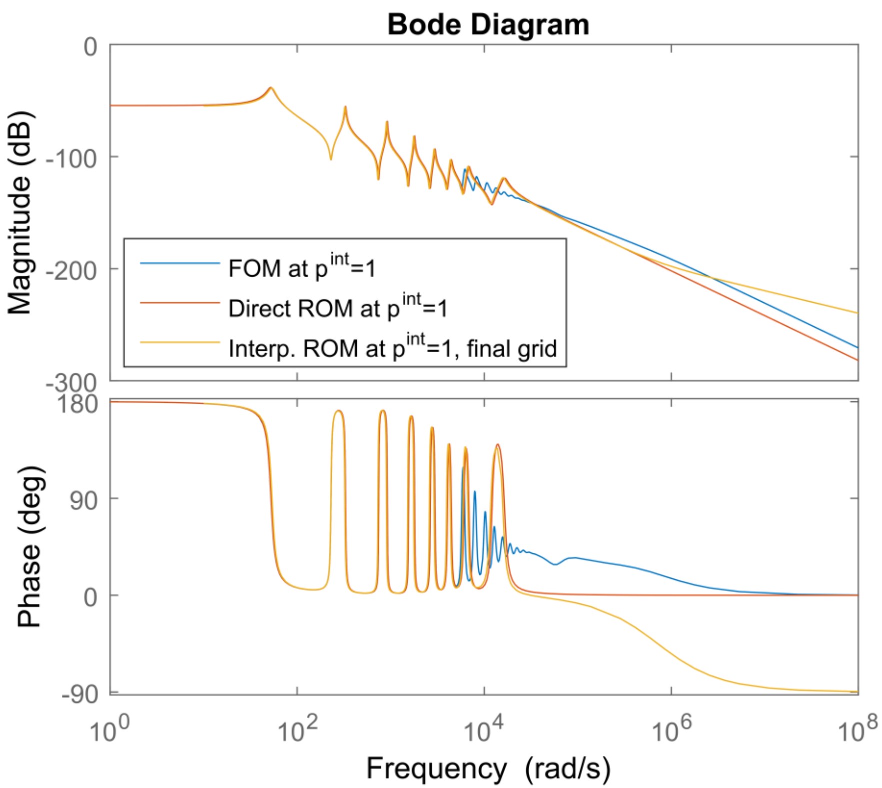
Table 1: Sample points p_i, subspace angles $\theta_{i,i+1}$ and ratios $l_{i,i+1}$

iter 1	p_i[m]	0.5								1.5				2.5		3.5		4.5		5.5
	$\theta_{i,i+1}[°]$								25.79			14.20			8.61		7.06		5.73	
	$l_{i,i+1}$								3			2			1		1		1	
iter 2	p_i[m]	0.5				0.833		1.167		1.5		2		2.5		3.5		4.5		5.5
	$\theta_{i,i+1}[°]$			10.15			8.59		7.05		8.18		6.03		8.61		7.06		5.73	
	$l_{i,i+1}$			2			1		1		1		1		1		1		1	
iter 3	p_i[m]	0.5		0.667		0.833		1.167		1.5		2		2.5		3.5		4.5		5.5
	$\theta_{i,i+1}[°]$		5.26		4.89		8.59		7.05		8.18		6.03		8.61		7.06		5.73	
	$l_{i,i+1}$		1		1		1		1		1		1		1		1		1	

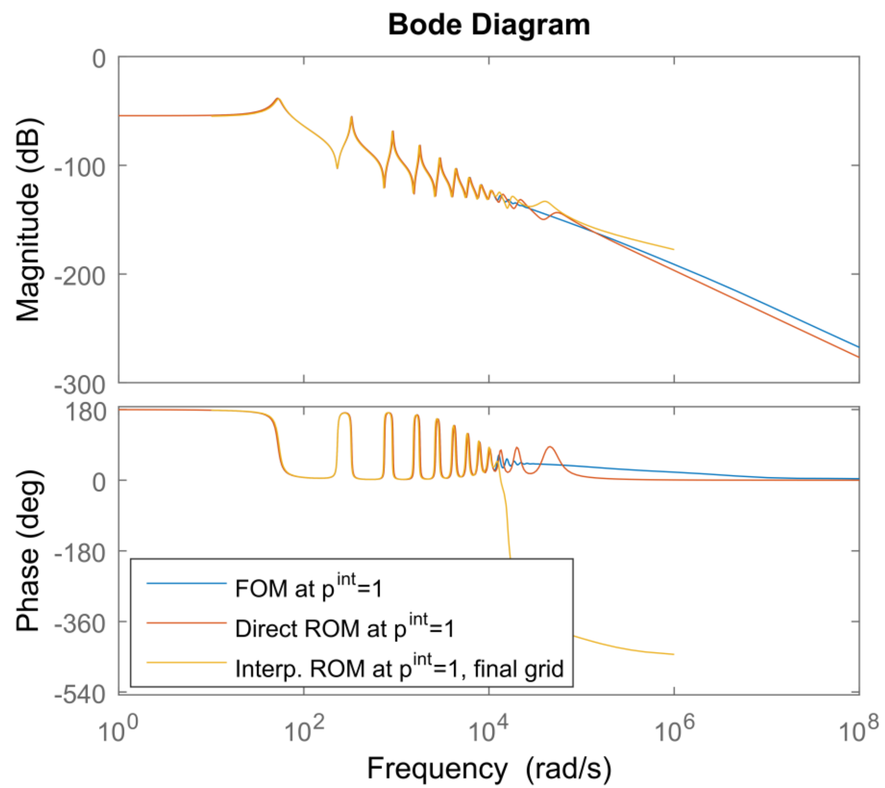
Interpolation point $p^{\text{int}} = 1.0$
between ROM 3 & ROM 4

Final refined grid with $N = 10$

Numerical results – Direct vs. Interpolated ROM



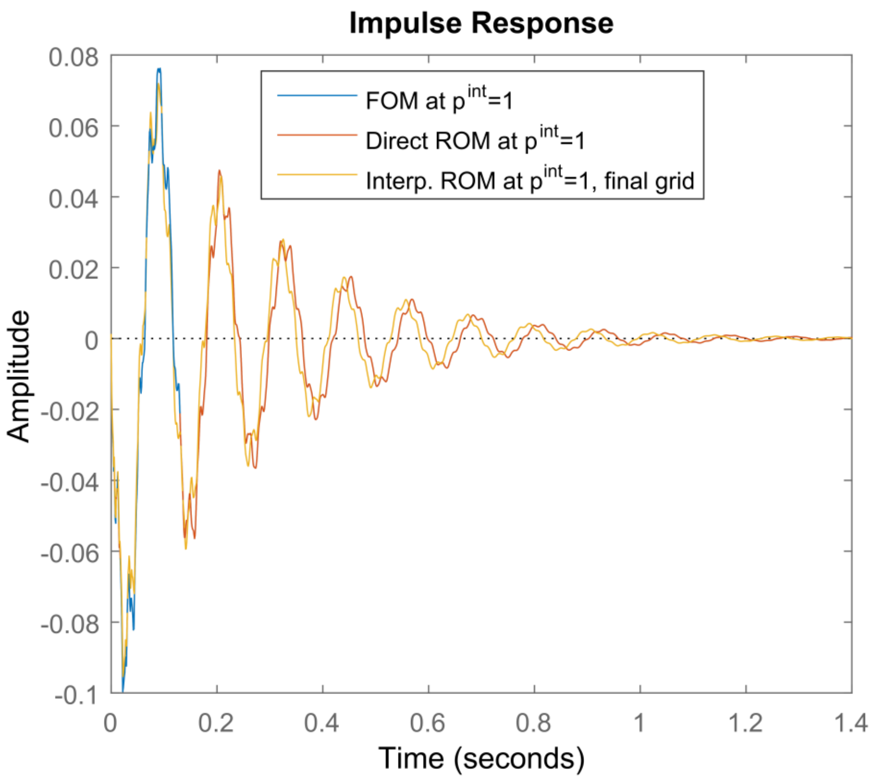
FOM size n = 240, ROMs size r = 17

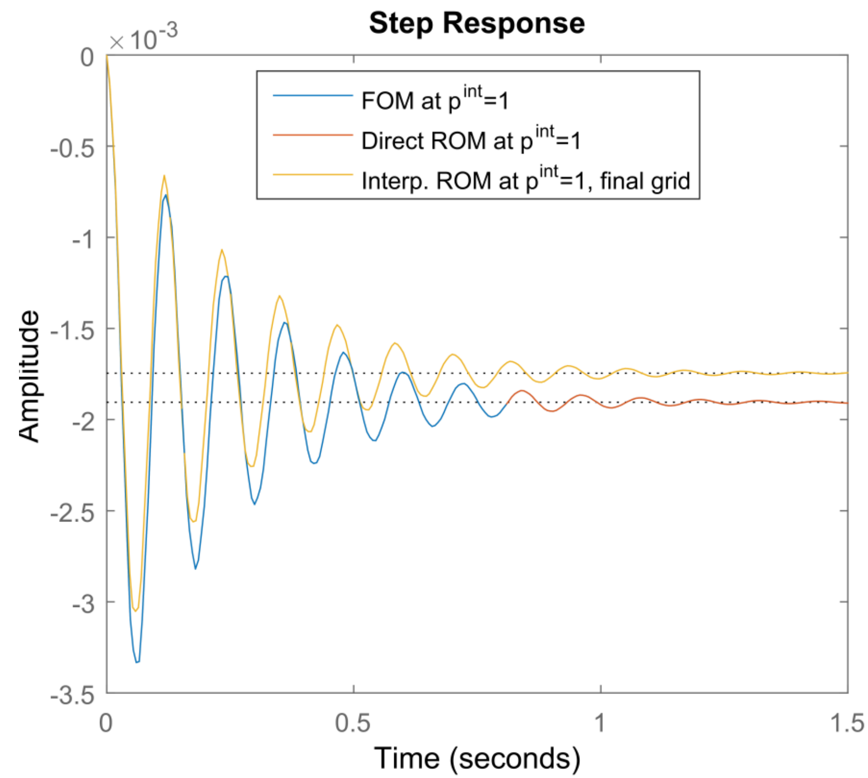


FOM size n = 2400, ROMs size r = 25

Two errors: model reduction error + interpolation error

Numerical results – Direct vs. Interpolated ROM



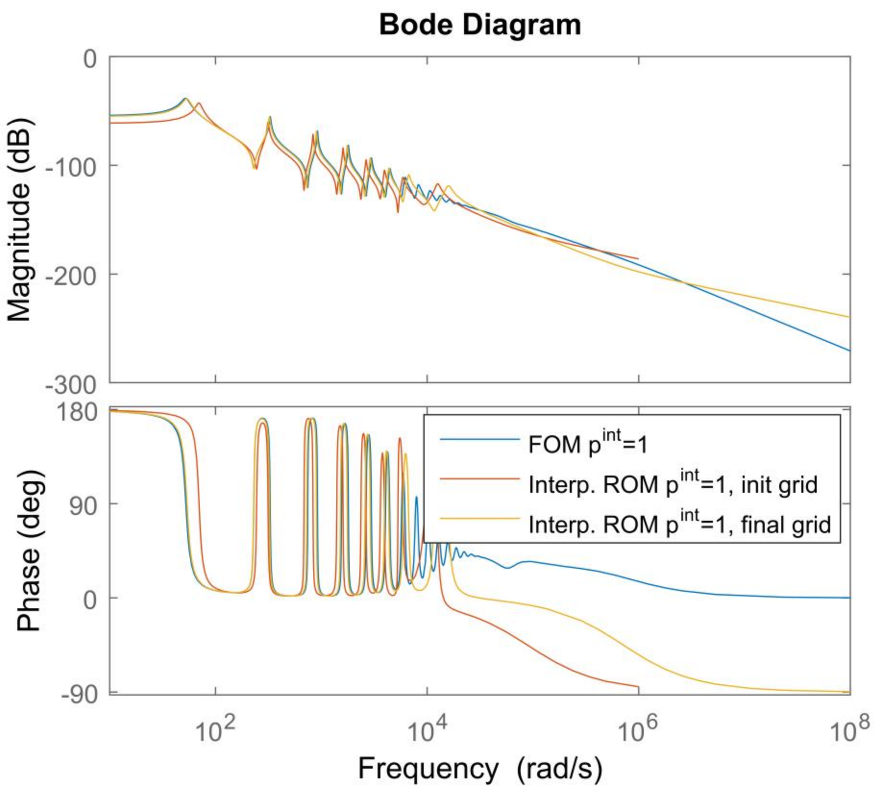


FOM size n = 2400, ROMs size r = 25

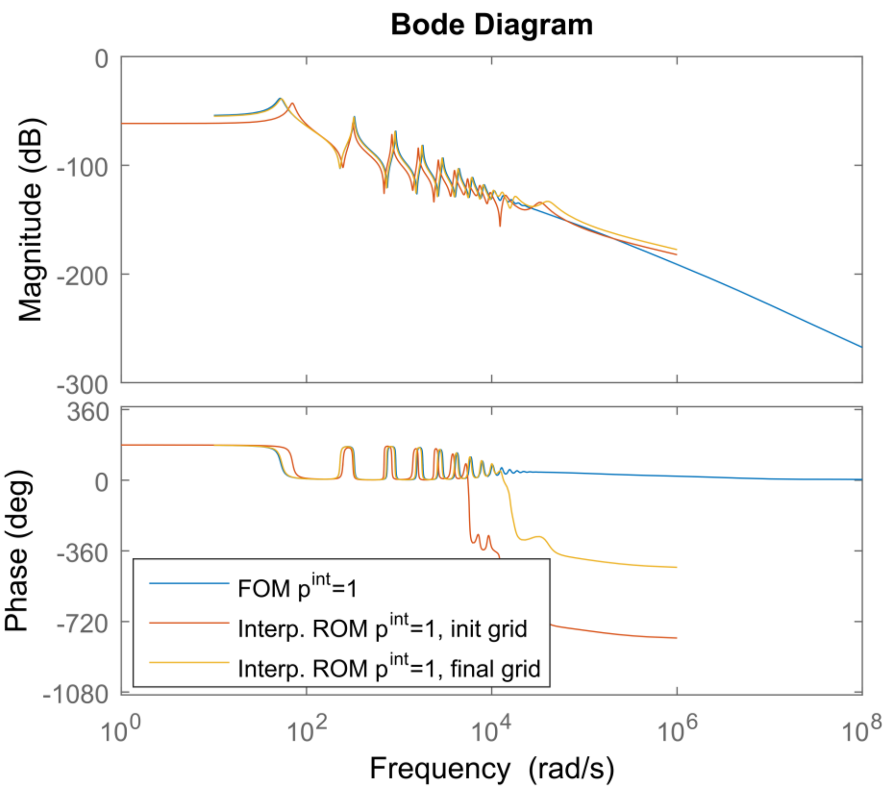
FOM size n = 2400, ROMs size r = 25

With MatrInterp: no need to reduce the model for every new parameter value

Numerical results – Initial vs. Final Grid



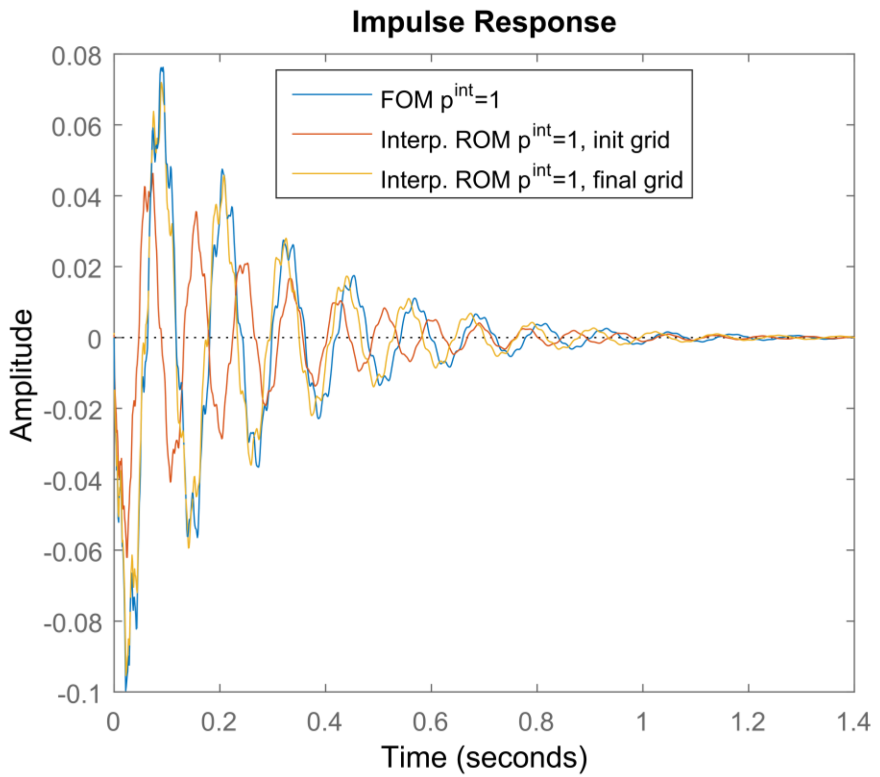
FOM size n = 240, ROMs size r = 17

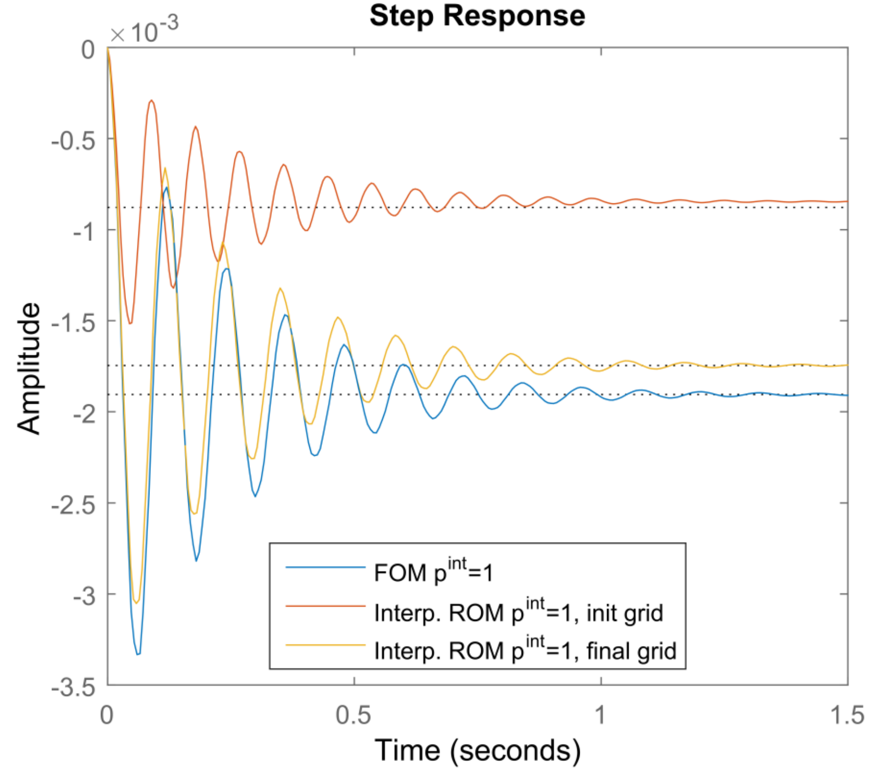


FOM size n = 2400, ROMs size r = 25

ROMs calculated with the final grid yield better approximations

Numerical results – Initial vs. Final Grid



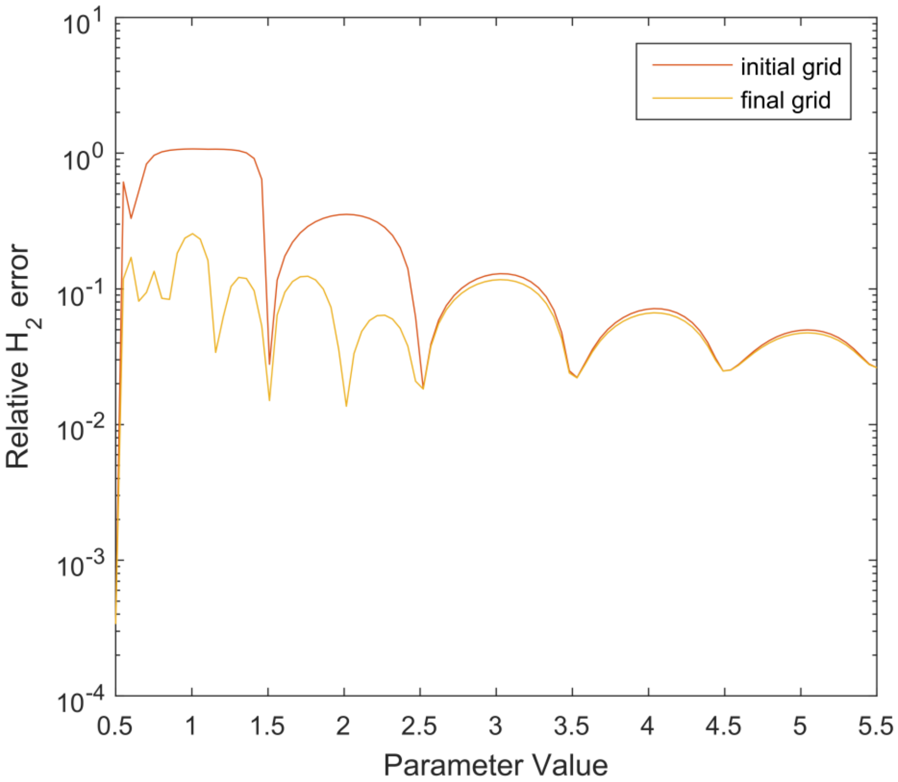


FOM size n = 2400, ROMs size r = 25

FOM size n = 2400, ROMs size r = 25

ROMs calculated with the final grid yield better approximations

Numerical results – Initial vs. Final Grid



Relative H2 error between FOMs and interpolated ROMs for different parameter values and grids: FOM size n = 240, ROM size r = 17

- **Quantitative evaluation of the approximation**
- **Relative H2 error for** `nP=100` **different query points** p^{int}
- **Errors particularly small in the proximity of the sample points**
- **Final grid yields smaller errors for smaller beam lengths due to the adaptive refinement in this region**

Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

Control and Automation Group
Department of Electrical Engineering
Indian Institute of Technology-Delhi

psssMOR Toolbox – Analysis and Reduction of Parametric Models in MATLAB

- ✓ Definition of **parametric sparse state-space models**

```
psys = loadFemBeam3D(Opts)
psys = loadAnemometer3parameter
```

- ✓ Manipulation of **psss-class objects**

```
psys = fixParameter(psys,2,1.7)
psys = unfixParameter(psys,3)
```

- ✓ **Compatible** with the sss & sssMOR toolboxes

```
param = [p1, p2, p3, p4]
sys = psys(param)

bode(psys,param); step(sys);
```

- ✓ Different **parametric reduction methods** available (offline- & online-phase)

```
psysr = matrInterpOffline
(psys,param,r,Opts);

psysr = globalPmorOffline
(psys,param,r,Opts)

sysr = psysr(pInterp)
```

- ✓ `localReduction` & `adaptiveSampling` as **core functions**

www.rt.mw.tum.de/?morlab

Summary & Outlook

Takehome Messages:

- A simple automatic sampling strategy is presented for **adaptively choosing sample points** in parametric model order reduction
- Scheme uses concept of **subspace angles** to measure need of further sampling points
- Adaptive approach is **fully automated** and **embedded in the matrix interpolation framework**
- Algorithm is applied to a **Timoshenko beam model**, achieving satisfactory results.

Future Extensions / Ongoing Work:

- Extension of proposed adaptive sampling scheme to 2D and 3D parametric case
- Higher dimensional case (d > 3) with adaptive sparse grids is topic of future research
- psssMOR toolbox is being actively developed and will be available open-source very soon!!

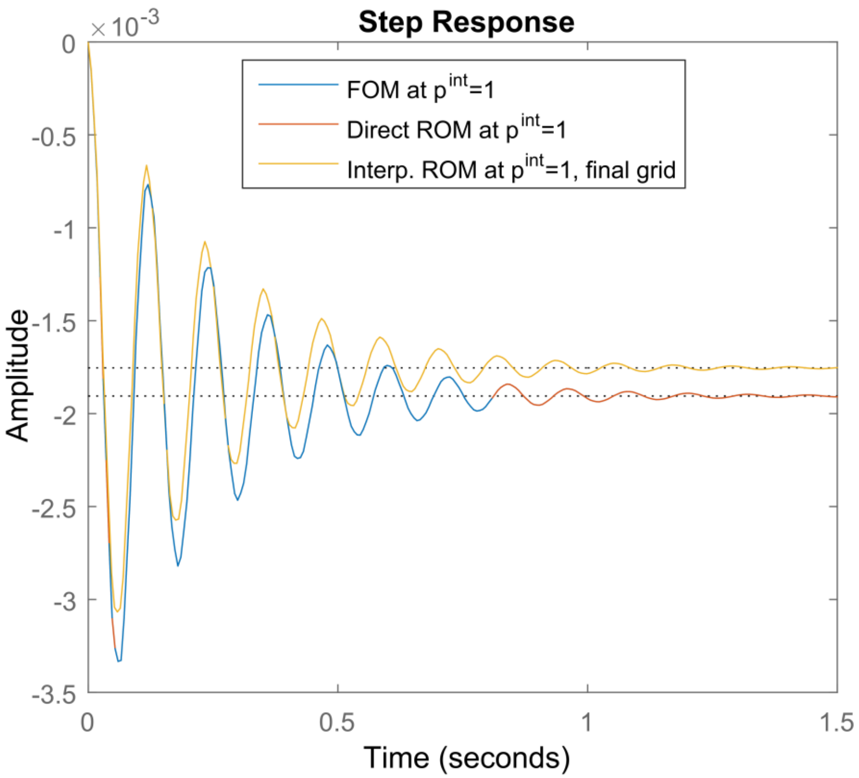
Thank you for your attention!

Backup

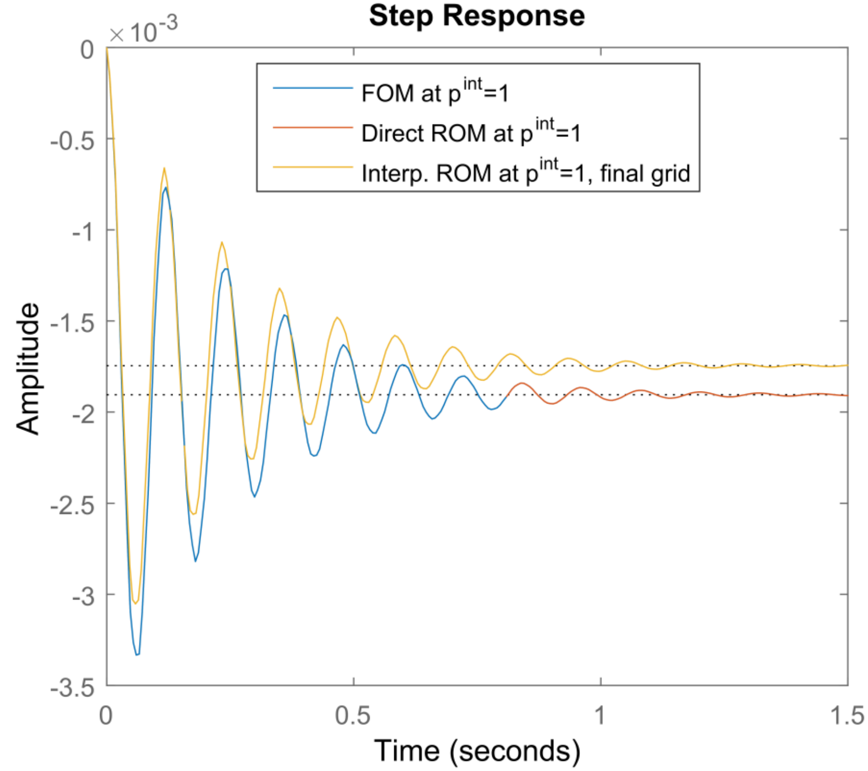
References

[Amsallem '10]	Interpolation on manifolds of CFD-based fluid and finite element-based structural reduced-order models for on-line….
[Bazaz et al. '15]	Adaptive Parameter Space Sampling in Matrix Interpolatory pMOR.
[Benner et al. '15]	A survey of projection-based model reduction methods for parametric dynamical systems.
[Baur et al. '15]	Comparison of methods for parametric model order reduction of instationary problems.
[Geuss et al. '08]	On Parametric Model Order Reduction by Matrix Interpolation.
[Panzer et al. '10]	Parametric Model Order Reduction by Matrix Interpolation.

Numerical results – Direct vs. Interpolated ROM

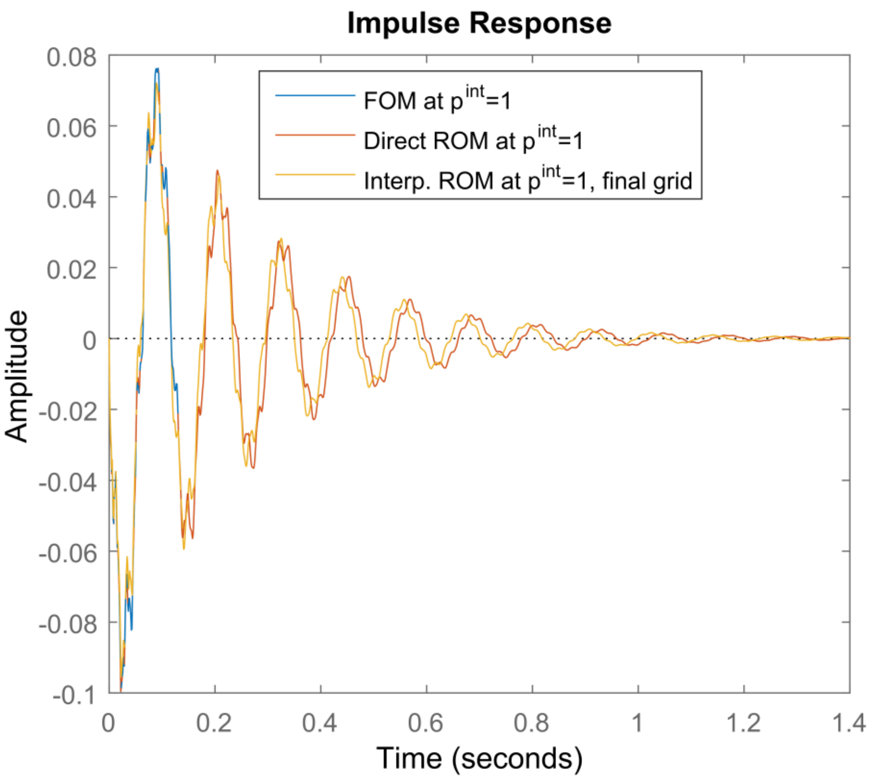


FOM size n = 240, ROMs size r = 17

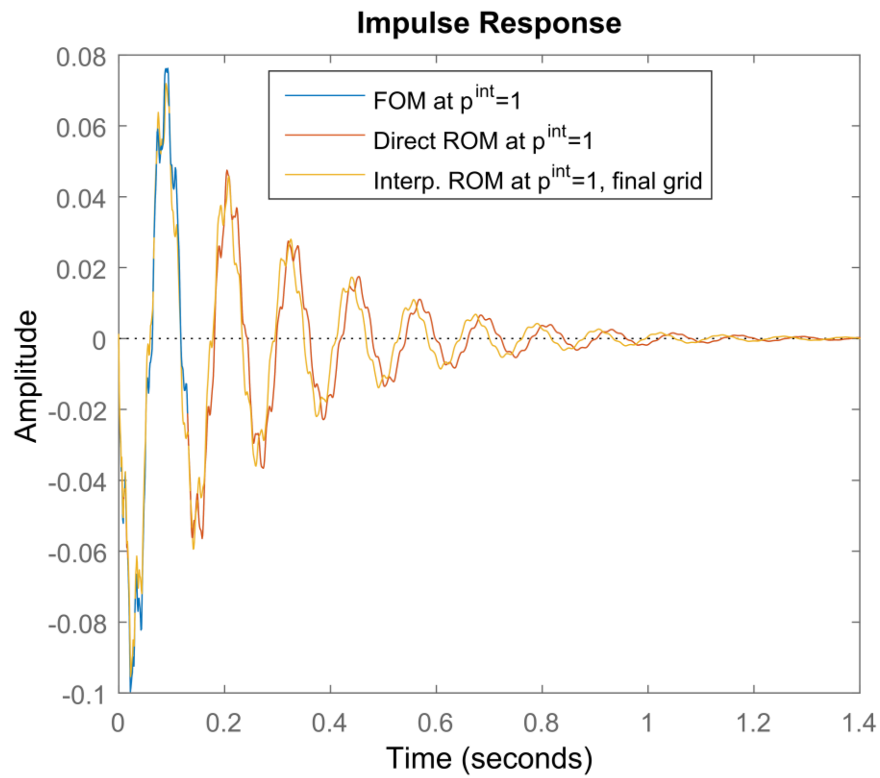


FOM size n = 2400, ROMs size r = 25

Numerical results – Direct vs. Interpolated ROM



FOM size n = 240, ROMs size r = 17

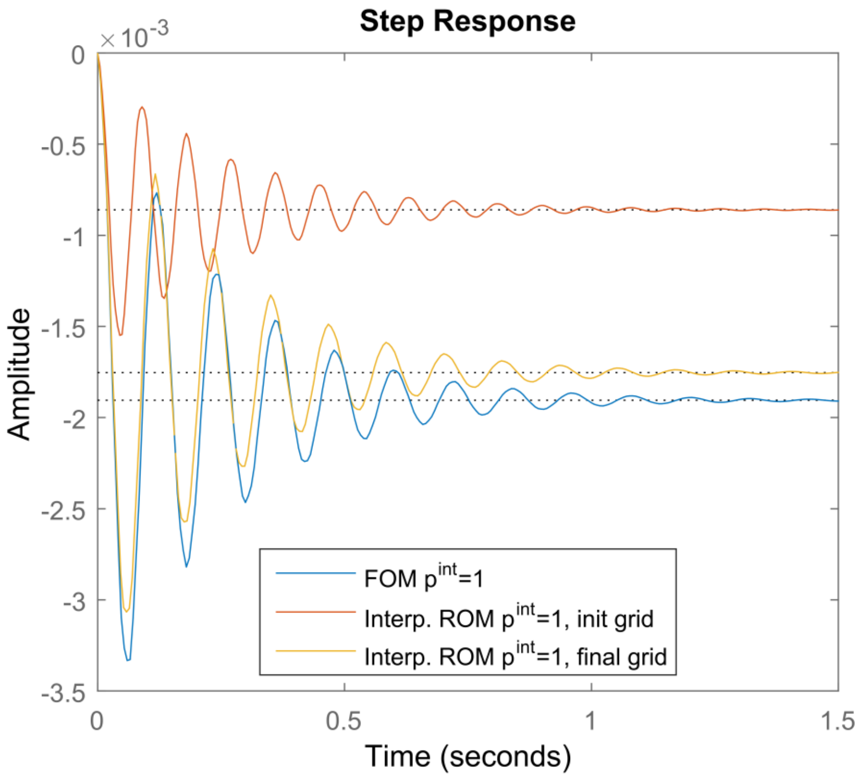


FOM size n = 2400, ROMs size r = 25

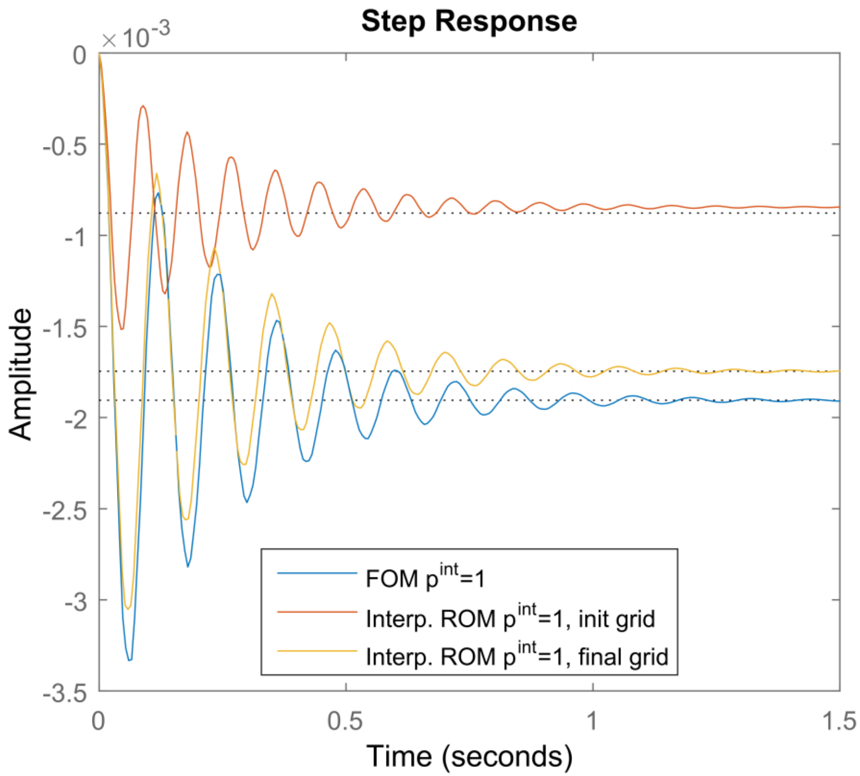
Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

Control and Automation Group
Department of Electrical Engineering
Indian Institute of Technology-Delhi

Numerical results – Initial vs. Final Grid



FOM size n = 240, ROMs size r = 17

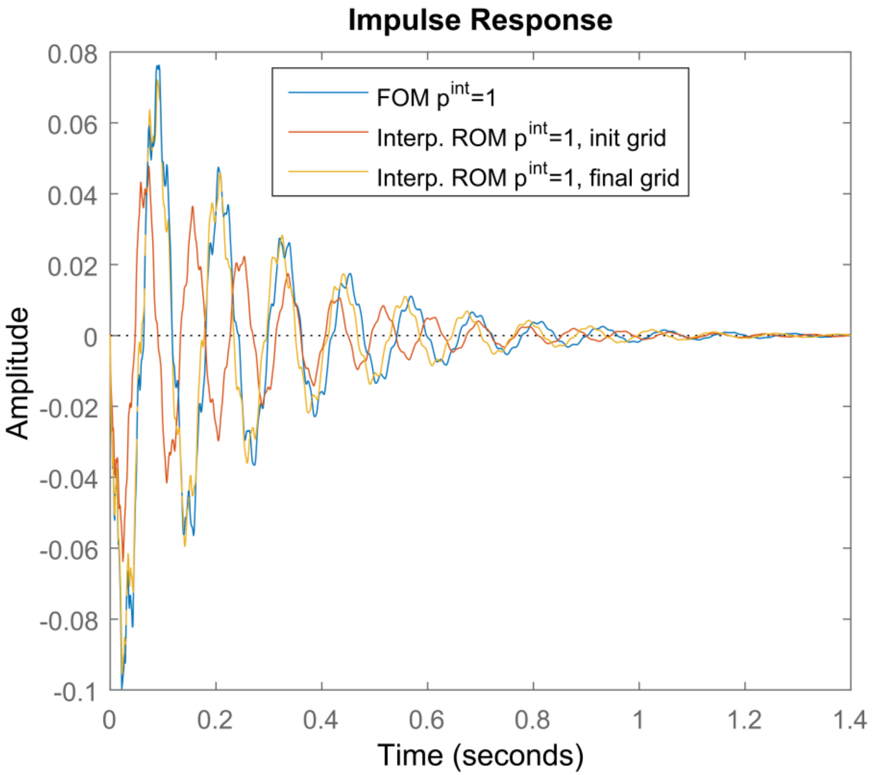


FOM size n = 2400, ROMs size r = 25

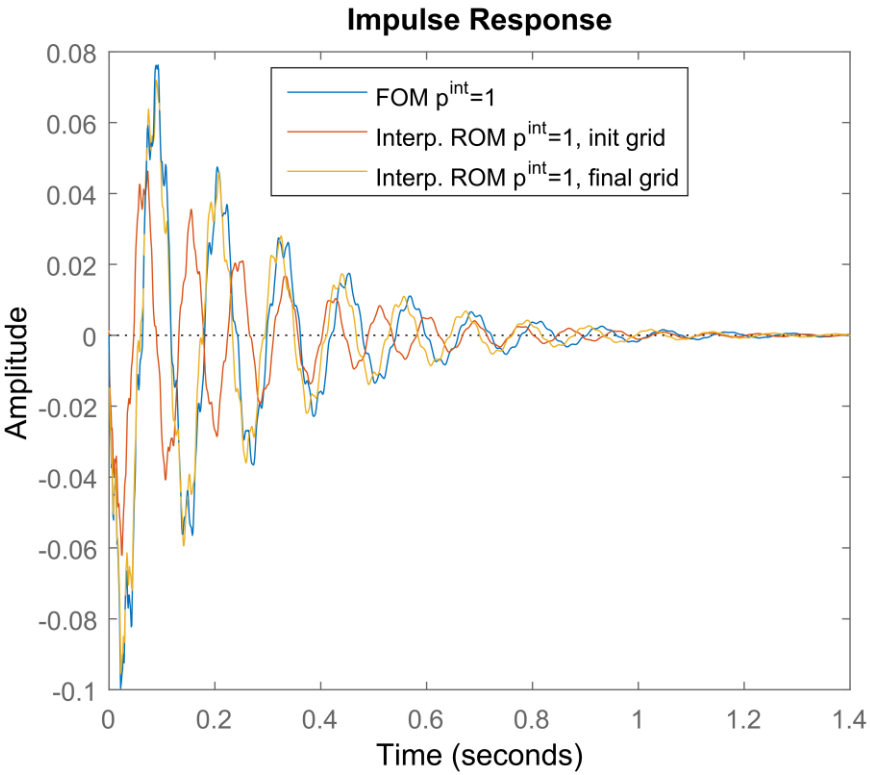
Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

Control and Automation Group
Department of Electrical Engineering
Indian Institute of Technology-Delhi

Numerical results – Initial vs. Final Grid

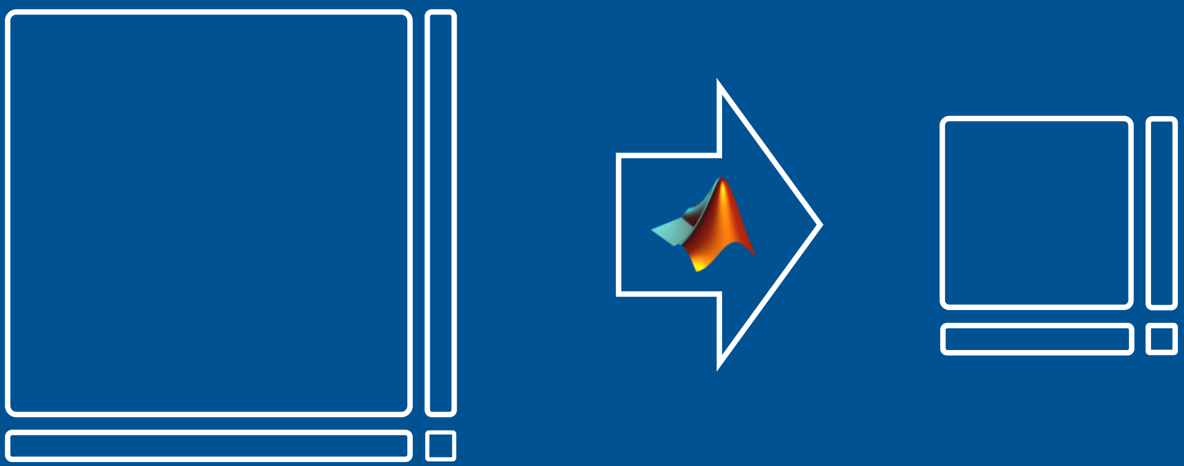


FOM size n = 240, ROMs size r = 17



FOM size n = 2400, ROMs size r = 25

sss & sssMOR – MATLAB Toolboxes



Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

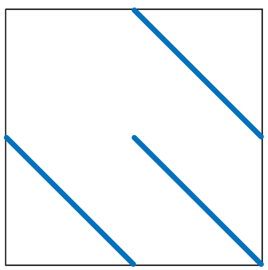
Control and Automation Group
Department of Electrical Engineering
Indian Institute of Technology-Delhi

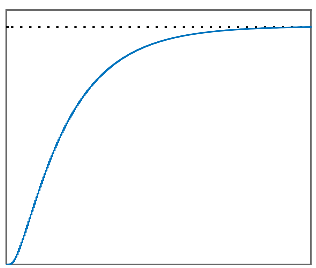
Toolboxes for sparse, large-scale models in

sss

sssMOR

```
sys = sss(A,B,C,D,E);
```

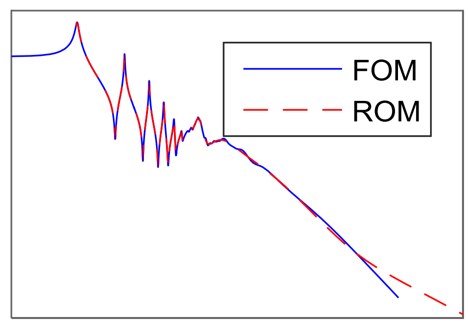


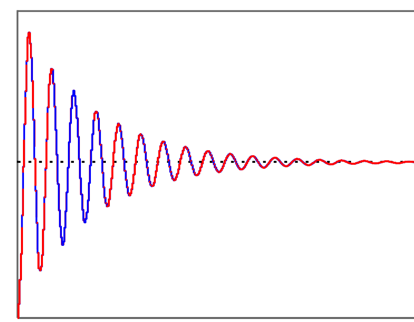


```
bode(sys), sigma(sys)
step(sys), impulse(sys)
norm(sys,2), norm(sys,inf)

c2d, lsim, eigs, connect,…
```

```
sysr = tbr(sys,r)
sysr = rk(sys,s0)
sysr = irka(sys,s0)
sysr = cure(sys)
sysr = cirka(sys,s0)
       ⋮
```





Powered by: **M-M.E.S.S. toolbox** [Saak, Köhler, Benner] for Lyapunov equations
Available at www.rt.mw.tum.de/?sssMOR
[Castagnotto/Cruz Varona/Jeschek/Lohmann ’17]: **„sss & sssMOR: Analysis and Reduction of Large-Scale Dynamic Systems in MATLAB“**, at-Automatisierungstechnik]

GitLab

Main characteristics

sss

- **State-space models** of very high order on a standard computer $\mathcal{O}\left(\mathbf{10^8}\right)$
- Many Control System Toolbox functions, revisited to **exploit sparsity**
- Allows system analysis in **frequency** (`bode, sigma`, …) and **time domain** (`step,norm,lsim`,…), as well as **manipulations** (`connect,truncate,`…)
- Is **compatible** with the built-in syntax
- **New functionality:** `eigs, residue, pzmap`,…

sssMOR

- **Classical** (`modalMor, tbr, rk`,…) and **state-of-the-art** (`isrk, irka, cirka, cure`,…) reduction methods
- Both **highly-automatized**

```
sysr = irka(sys,n)
```

  and **highly-customizable**

```
Opts.maxiter = 100
Opts.tol = 1e-6
Opts.stopcrit = 'combAll'
Opts.verbose = true
sysr = irka(sys,n,Opts)
```

- `solveLse` and `lyapchol` as **core functions**

Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

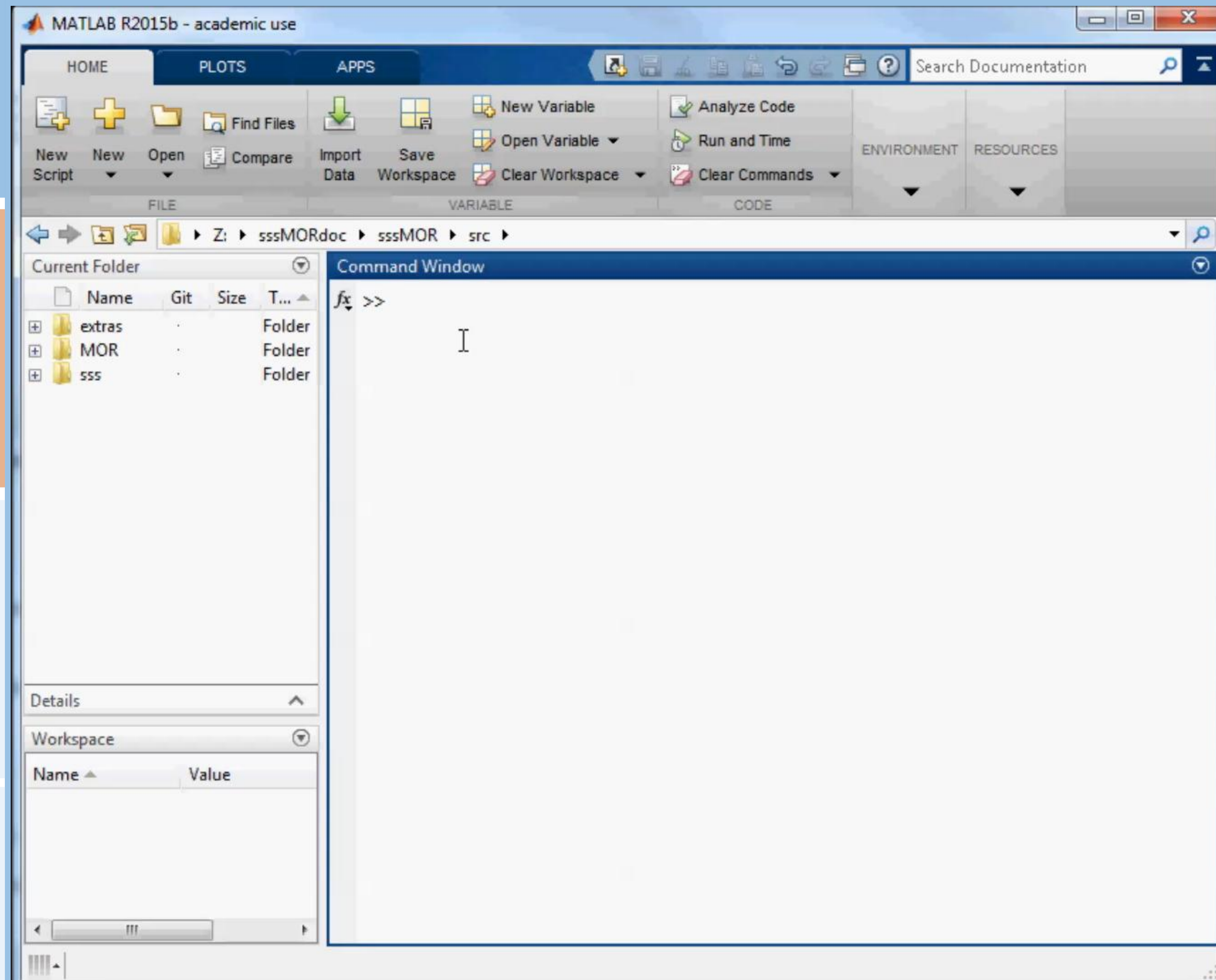
Control and Automation Group
Department of Electrical Engineering
Indian Institute of Technology-Delhi

sss sssMOR

Comprehensive **documentation** with examples and references

sssMOR App
graphical user interface

completely **free**
and **open source**
(contributions welcome)



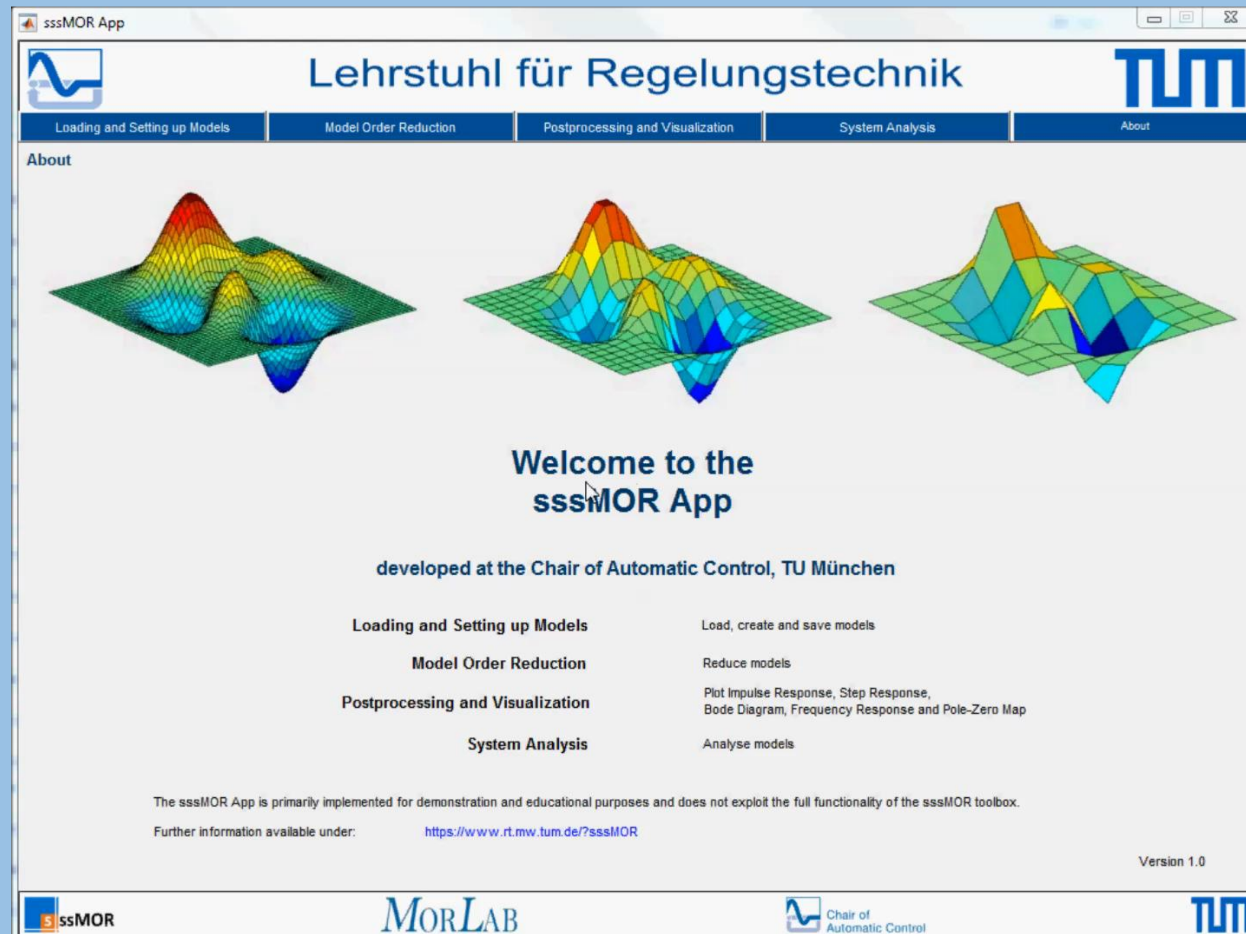
Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

Control and Automation Group
Department of Electrical Engineering
Indian Institute of Technology-Delhi

Comprehensive **documentation** with examples and references

sssMOR App
graphical user interface

completely **free** and **open source** (contributions welcome)



Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

Control and Automation Group
Department of Electrical Engineering
Indian Institute of Technology-Delhi

sss sssMOR

Comprehensive **documentation** with examples and references

sssMOR App
graphical user interface

completely **free** and **open source** (contributions welcome)