An Introduction to Model Order Reduction:
from linear to nonlinear dynamical systems

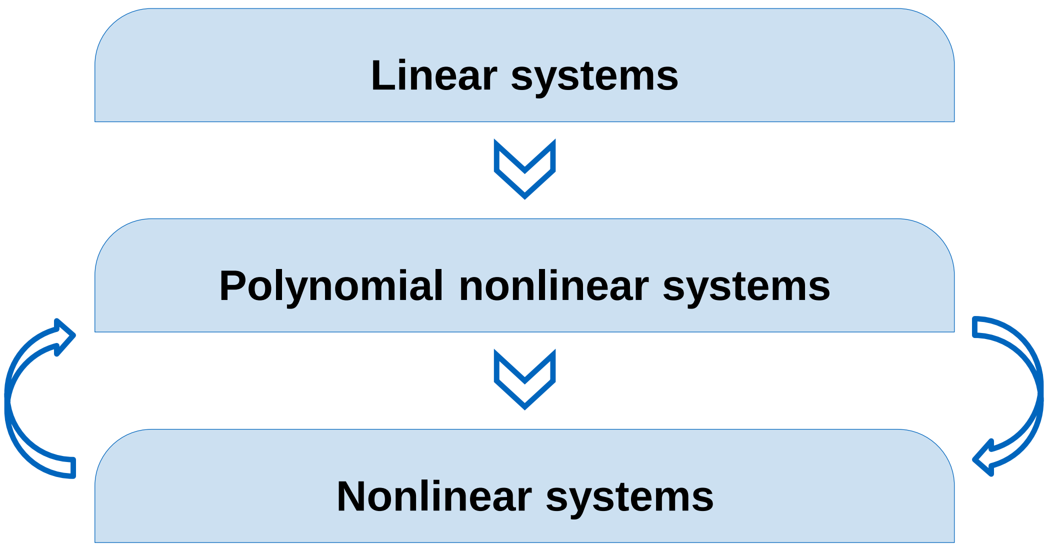
Maria Cruz Varona

Chair of Automatic Control

Department of Mechanical Engineering

Technical University of Munich

June 5th 2018



Brief personal introduction

Maria Cruz Varona
M.Sc. Electrical Engineering

University studies (10/08-03/14):

Electrical Engineering and Information Technology (KIT)

Study model 8: “Information and Automation”

Master thesis at IRS (group: “cooperative systems”)

Research assistant (since 08/14):

Chair of Automatic Control (Prof. Dr.-Ing. habil. B. Lohmann)

Technical University of Munich

maria.cruz@tum.de

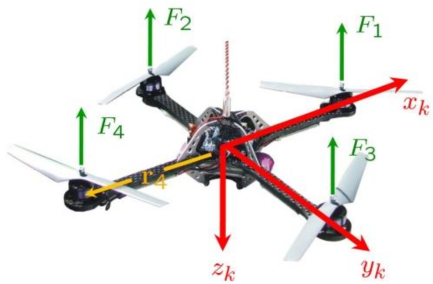
www.rt.mw.tum.de

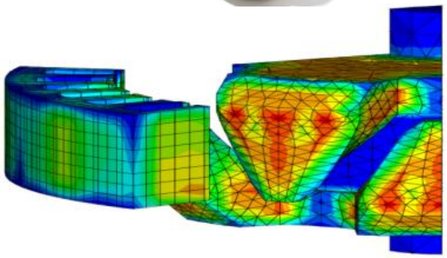
Research interests:

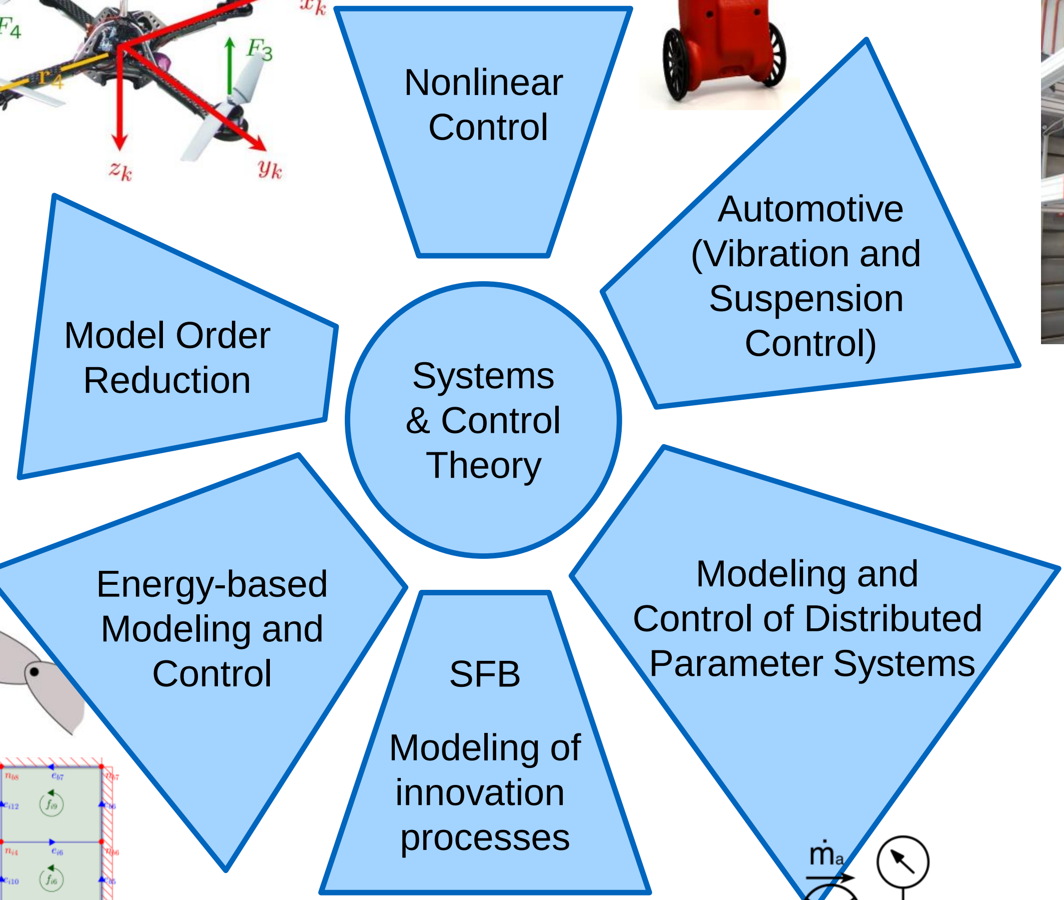
Systems theory, model order reduction, nonlinear dynamical systems, Krylov subspace methods

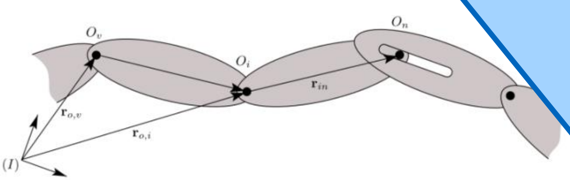
Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

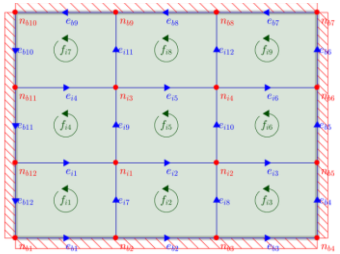
Chair of Automatic Control

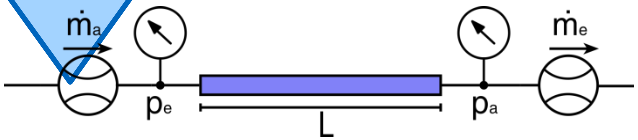






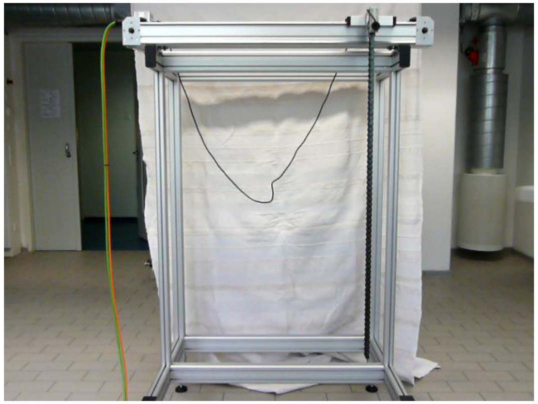


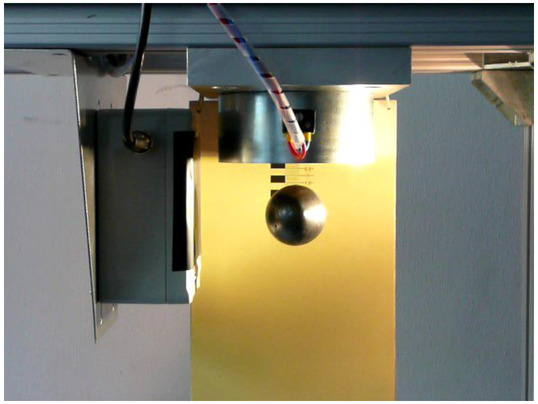


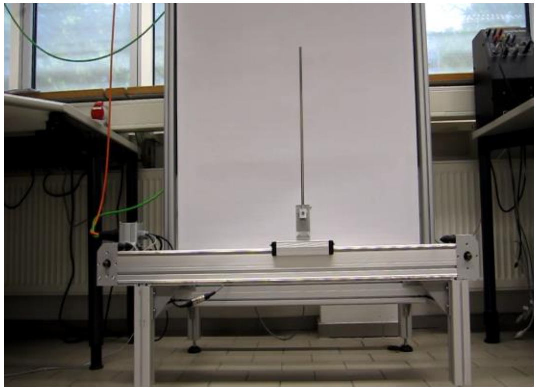


Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

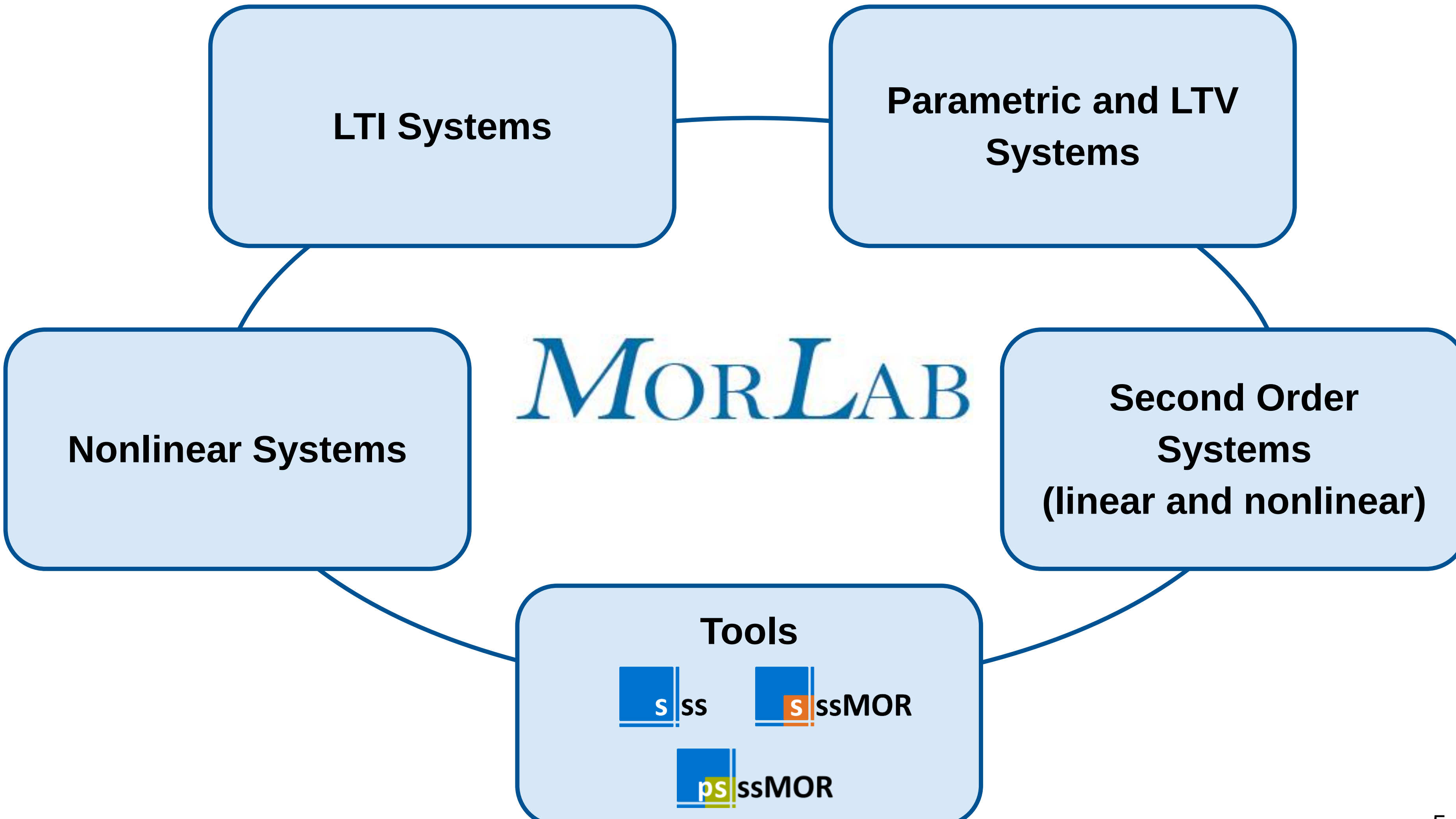
Chair of Automatic Control







Chair of Automatic Control – MORLab



Agenda

I. Motivation & Linear Model Order Reduction

- Modeling, Modeling Strategies
- Large-scale models, Sparsity
- Reduced order models, Applications
- Projective MOR, Linear MOR methods
- Numerical Examples, FEM & MOR software

II. Polynomial & Nonlinear Model Order Reduction

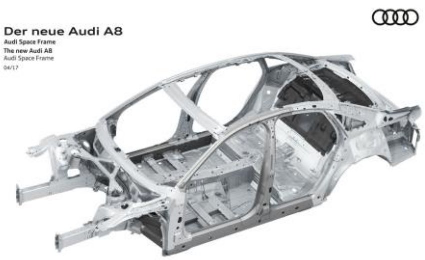
- Projective NLMOR, Overview NLMOR methods
- Polynomial Nonlinear Systems, Volterra series representation
- Nonlinear Systems, Proper Orthogonal Decomposition

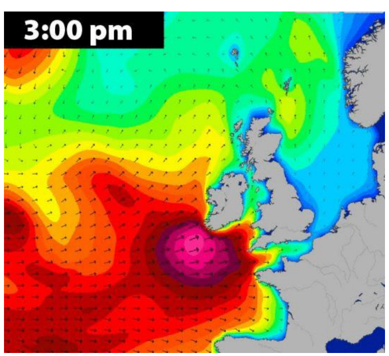
III. Summary & Outlook

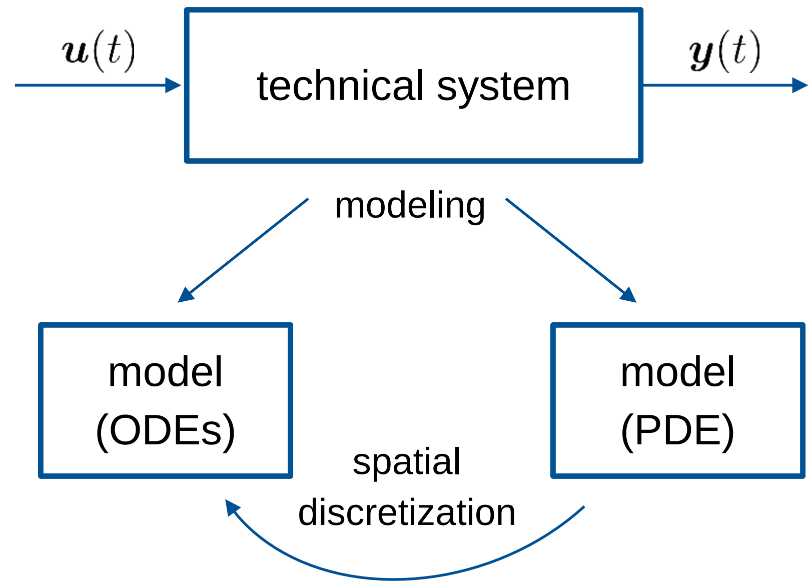
Motivation & Linear Model Order Reduction

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

Modeling of complex dynamical systems







- Models described by ODEs:

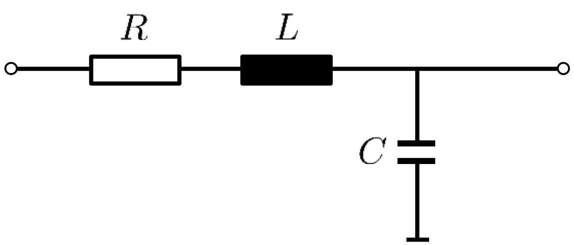
$$\frac{\mathrm{d}}{\mathrm{dt}}\boldsymbol{x}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t))$$

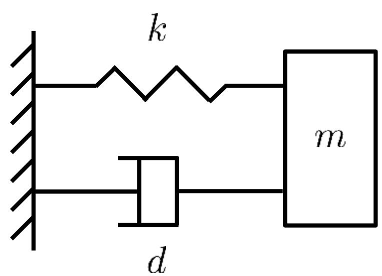
- Models described by PDEs:

$$\frac{\partial T(z,t)}{\partial t} = \frac{\partial^2 T(z,t)}{\partial z^2} + u(z,t)$$

Modeling – Strategies

0D modeling: lumped-parameter model





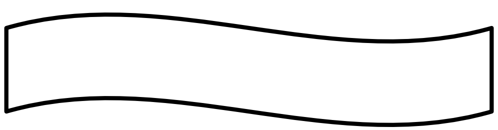
u : voltage ⇔ v : velocity

i : current ⇔ F : force

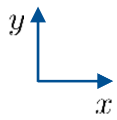
p : pressure ⇔ e : effort

q : flow rate ⇔ f : flow

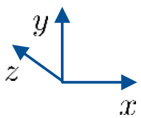
1D, 2D, 3D modeling: distributed-parameter model



$p(x,t),\ q(x,t)$



$p(x,y,t),\ q(x,y,t)$

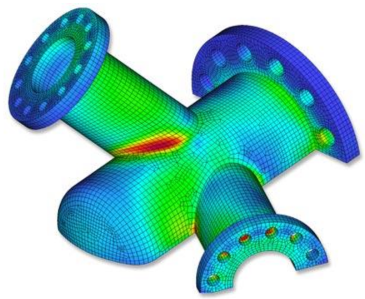


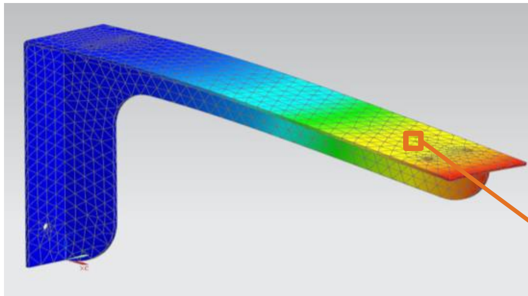
$p(x,y,z,t),\ q(x,y,z,t)$

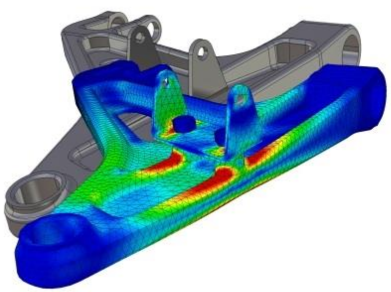
Data-driven modeling: identification of model using experimental data

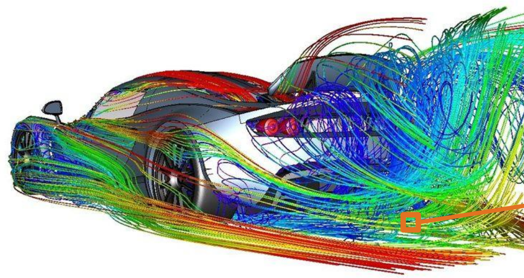
Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

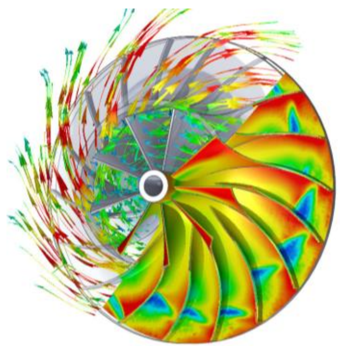
Large-scale models from spatial discretization

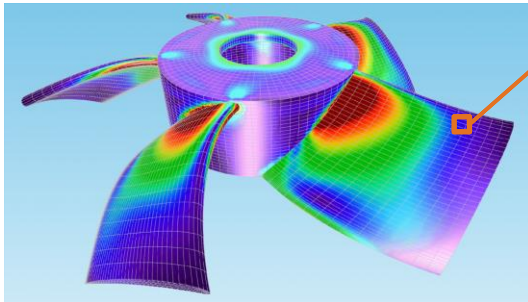












Spatial discretization using:

- Finite-Difference-Method (FDM)
- Finite-Element-Method (FEM)
- Finite-Volume-Method (FVM)

$$\boldsymbol{x}(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_k(t) \\ \vdots \\ x_n(t) \end{bmatrix} \qquad n = 10^3, \ldots, 10^6$$

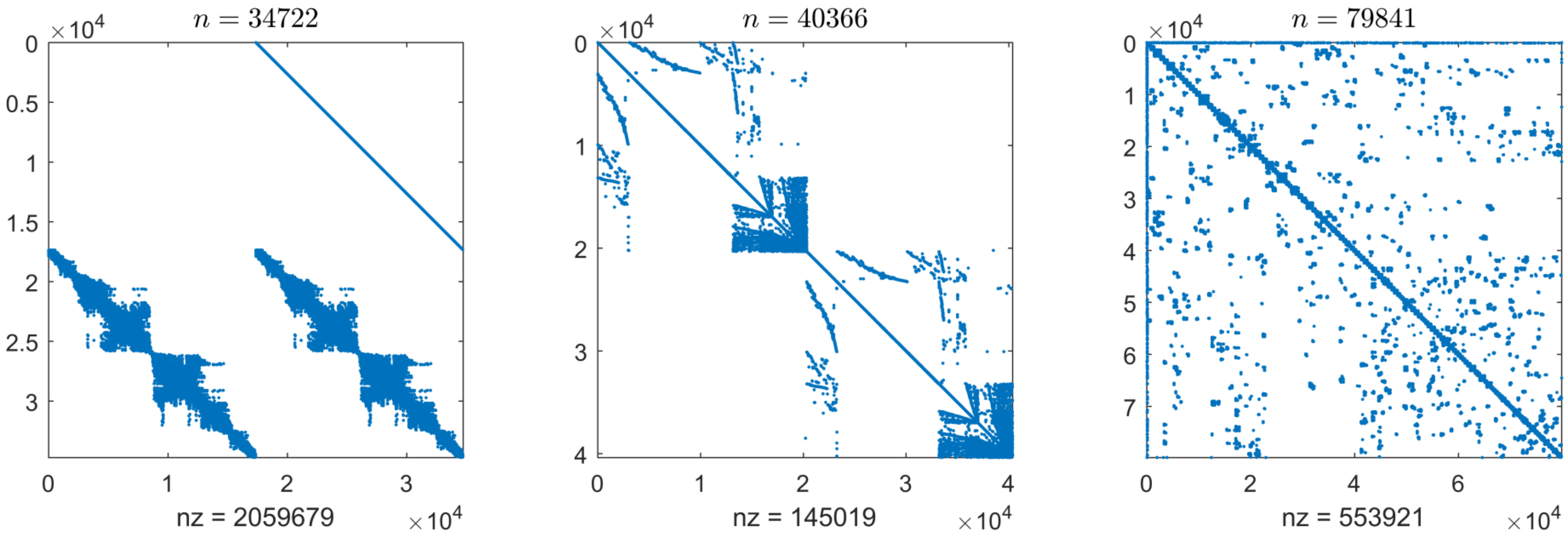
High dimension complicates:

- numerical simulation
- design optimization
- estimation, prediction & control

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

Sparsity of matrices

Matrices coming from FEM/FVM discretization are generally *sparse*



gyroscope   power system Bauru   steel profile rail_79841

Storage requirement: $\boldsymbol{A} \in \mathbb{R}^{34722 \times 34722}$

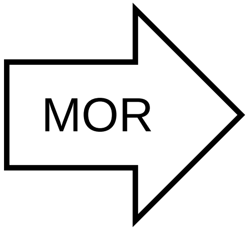
- Sparse: ~33.2 MB
- Full / Dense: 9.0 GB required!

Goal of Model Order Reduction (MOR)

Large-scale full order model (FOM)

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

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} \qquad \det(\boldsymbol{E}) \neq 0$$

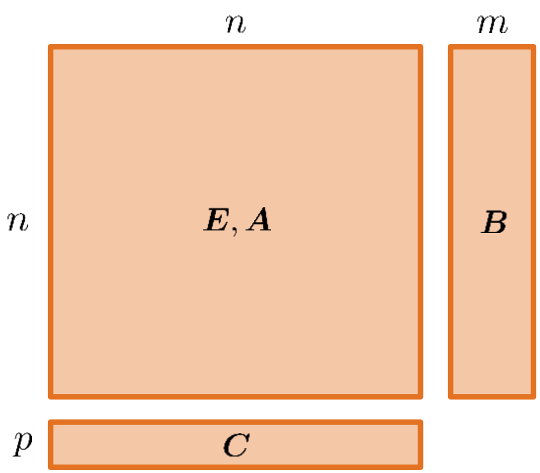


Reduced order model (ROM)

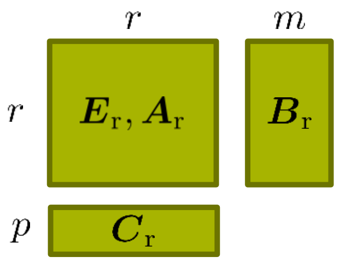
$$\boldsymbol{E}_{\mathrm{r}}\dot{\boldsymbol{x}}_{\mathrm{r}} = \boldsymbol{A}_{\mathrm{r}}\boldsymbol{x}_{\mathrm{r}} + \boldsymbol{B}_{\mathrm{r}}\boldsymbol{u}$$

$$\boldsymbol{y}_{\mathrm{r}} = \boldsymbol{C}_{\mathrm{r}}\boldsymbol{x}_{\mathrm{r}}$$

$$\boldsymbol{x}_{\mathrm{r}} \in \mathbb{R}^r, \; r \ll n$$



- ✓ good approximation
- ✓ preservation of properties
- ✓ numerically efficient



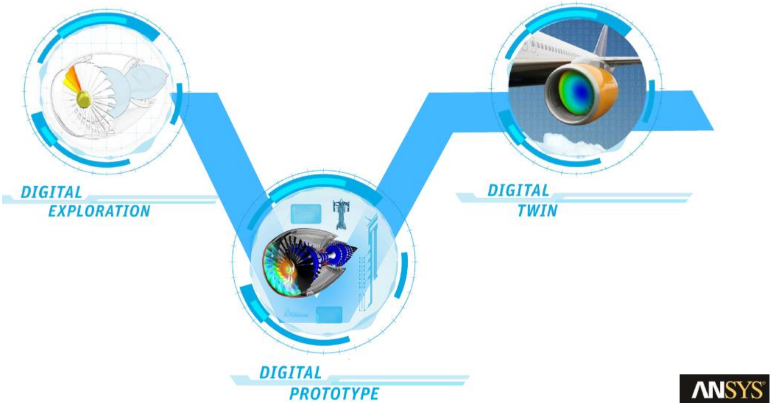
Applications of ROMs

Off-line applications:

- Efficient numerical simulation – "solves in seconds vs. hours"
- Design optimization – analysis for different parameters and "what if" scenarios
- Computer-aided failure mode and effects analysis (FMEA) – validation

On-line applications:

- Parameter estimation, Uncertainty Quantification
- Real-time optimization and control
- Digital Twin, Predictive Maintenance



Physical domains:

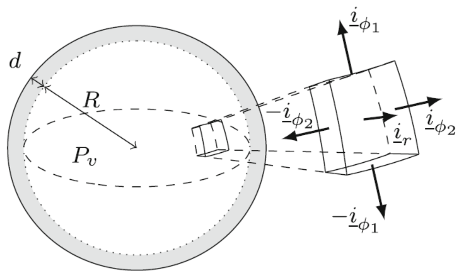
mechanical, electrical, thermal, fluid, acoustics, electromagnetism, …

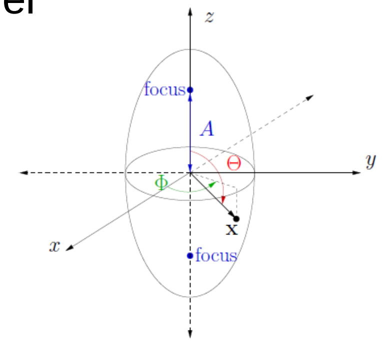
Application areas:

CSD, CFD, FSI, EMBS, MEMS, crash simulation, vibroacoustics, civil & geo, biomedical, …

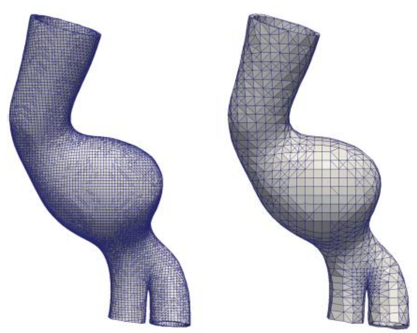
Reduced Order Modeling – Strategies

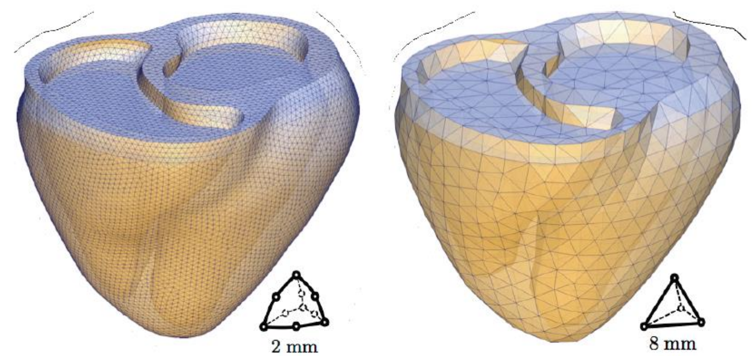
0D modeling: lumped-parameter / simplified model



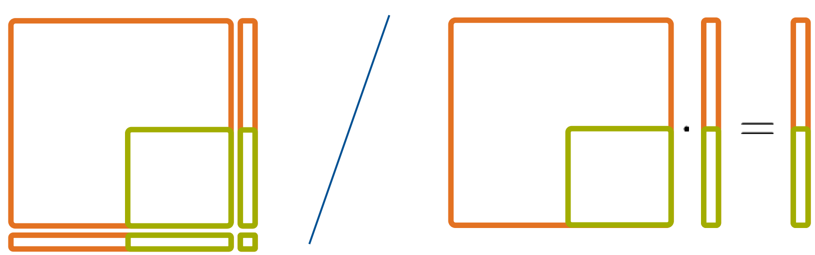


Coarse mesh:



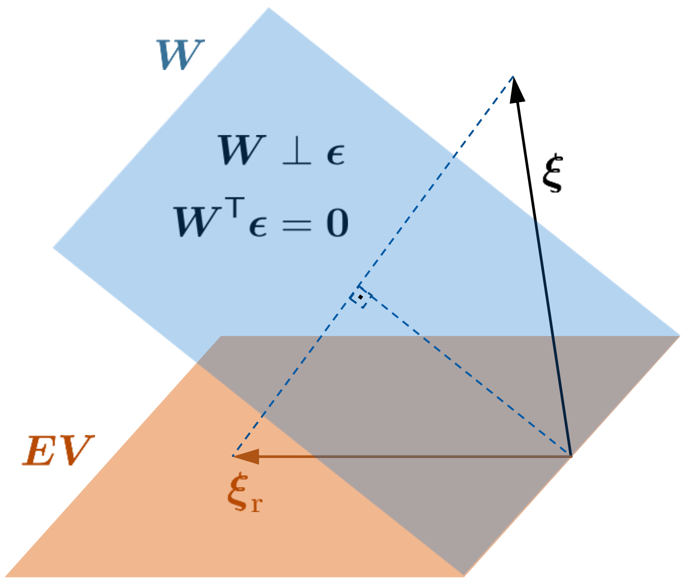


Fine mesh & Projection-based MOR:



Projective MOR

Assumption: Dynamical system does not transit all regions of the state-space equally often, but mainly stays and evolves in a subspace of lower dimension



Approximation of the state vector:

$$x = V x_r + e, \quad V \in \mathbb{R}^{n \times r}$$

Petrov-Galerkin projection: $\Pi = EV(W^\top EV)^{-1}W^\top$

$$\underbrace{W^\top E V}_{E_r} \dot{x}_r = \underbrace{W^\top A V}_{A_r} x_r + \underbrace{W^\top B}_{B_r} u$$

$$y \approx y_r = \underbrace{C V}_{C_r} x_r$$

How to choose **V** and **W**?

Linear MOR methods – Overview

1. Modal Reduction
   - Preservation of dominant eigenmodes
   - Frequently used in structural dynamics / second order systems
2. Truncated Balanced Realization / Balanced Truncation
   - Retention of state-space directions with highest energy transfer
   - Requires solution of Lyapunov equations, i.e. linear matrix equations (LMEs)
   - Applicable for medium-scale models: $n \approx 5000$
3. Rational Krylov subspaces
   - "Moment Matching": matching some Taylor-series coefficients of the transfer function
   - Requires solution of linear systems of equations (LSEs) – applicable for $n \approx 10^6$
   - Also employed for: approximate solution of eigenvalue problems, LSEs, LMEs,…
4. Iterative Krylov algorithm IRKA
   - H2-optimal reduction
   - Adaptive choice of Krylov reduction parameters (e.g. shifts)

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

Modal Reduction

Goal: Preserve dominant eigenmodes of the system

Procedure:

1 *Modal transformation:* Bring system into modal coordinates through state-transformation

$$[\boldsymbol{W}, \boldsymbol{\Lambda}, \boldsymbol{V}] = \lambda(\boldsymbol{A}, \boldsymbol{E})$$

$$\boldsymbol{W}^\top \boldsymbol{A} \boldsymbol{V} = \boldsymbol{\Lambda}, \quad \boldsymbol{W}^\top \boldsymbol{E} \boldsymbol{V} = \mathbf{I}_r$$

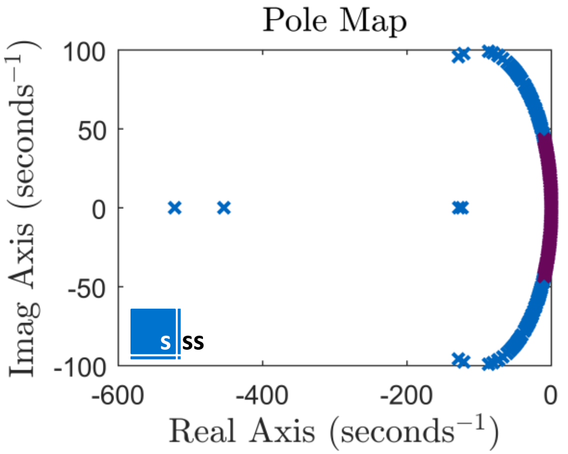
$$\hat{\boldsymbol{B}} = \boldsymbol{W}^\top \boldsymbol{B}, \qquad \hat{\boldsymbol{C}} = \boldsymbol{C} \boldsymbol{V}$$

$$\boldsymbol{x} = \boldsymbol{V} \boldsymbol{z} \quad \Rightarrow$$

$$\dot{\boldsymbol{z}} = \begin{bmatrix} \boldsymbol{\Lambda}_1 & \\ & \boldsymbol{\Lambda}_2 \end{bmatrix} \boldsymbol{z} + \begin{bmatrix} \hat{\boldsymbol{B}}_1 \\ \hat{\boldsymbol{B}}_2 \end{bmatrix} \boldsymbol{u}$$

$$\boldsymbol{y} = \begin{bmatrix} \hat{\boldsymbol{C}}_1 & \hat{\boldsymbol{C}}_2 \end{bmatrix} \boldsymbol{z}$$

2 *Truncation:* $\boldsymbol{A}_\mathrm{r} = \boldsymbol{\Lambda}_1, \ \boldsymbol{B}_\mathrm{r} = \hat{\boldsymbol{B}}_1, \ \boldsymbol{C}_\mathrm{r} = \hat{\boldsymbol{C}}_1, \ \boldsymbol{E}_\mathrm{r} = \mathbf{I}_r$



Practical implementation:

Entire modal transformation of FOM is expensive!

→ Only a few eigenvalues and left and right eigenvectors are computed via `eigs`

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

Truncated Balanced Realization (TBR)

Goal: Preserve state-space directions with highest enery transfer

Controllability and Observability Gramians:

$$\boldsymbol{A}\,\boldsymbol{P}\,\boldsymbol{E}^\mathsf{T} + \boldsymbol{E}\,\boldsymbol{P}\,\boldsymbol{A}^\mathsf{T} + \boldsymbol{B}\,\boldsymbol{B}^\mathsf{T} = \boldsymbol{0}$$

$$\boldsymbol{A}^\mathsf{T}\,\boldsymbol{Q}\,\boldsymbol{E} + \boldsymbol{E}^\mathsf{T}\,\boldsymbol{Q}\,\boldsymbol{A} + \boldsymbol{C}^\mathsf{T}\,\boldsymbol{C} = \boldsymbol{0}$$

Energy interpretation:

$$\min_{\boldsymbol{x}(0)=\boldsymbol{0},\ \boldsymbol{x}(\infty)=\boldsymbol{x}_\mathrm{e}} \int_0^\infty |\boldsymbol{u}(t)|^2\,\mathrm{d}t = \boldsymbol{x}_\mathrm{e}^\mathsf{T}\,\boldsymbol{P}^{-1}\,\boldsymbol{x}_\mathrm{e}$$

$$\|\boldsymbol{y}(t)\|_2^2 = \boldsymbol{x}_0^\mathsf{T}\,\boldsymbol{Q}\,\boldsymbol{x}_0$$

Procedure:

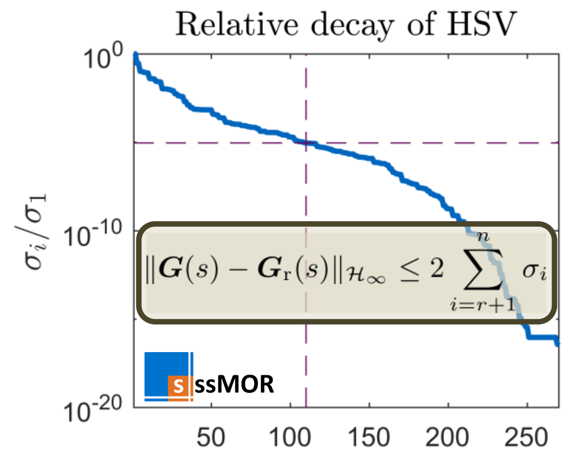
1. *Balancing step:* Compute balanced realization, where $\boldsymbol{P} = \boldsymbol{E}^\mathsf{T}\boldsymbol{Q}\boldsymbol{E} = \boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1, \dots, \sigma_n)$

$$\boldsymbol{P} = \boldsymbol{R}\boldsymbol{R}^\mathsf{T}, \quad \boldsymbol{Q} = \boldsymbol{S}\boldsymbol{S}^\mathsf{T}$$

$$\boldsymbol{S}^\mathsf{T}\boldsymbol{E}\boldsymbol{R} = \begin{bmatrix}\boldsymbol{U}_1 & \boldsymbol{U}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{\Sigma}_1 & \\ & \boldsymbol{\Sigma}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{N}_1^\mathsf{T} \\ \boldsymbol{N}_2^\mathsf{T}\end{bmatrix}$$

2. *Truncation step:* $\sigma_i \gg \sigma_j,\ i = 1, \dots, r,\ j = r+1, \dots, n$

$$\boldsymbol{W}^\mathsf{T} = \boldsymbol{\Sigma}_1^{-1/2}\,\boldsymbol{U}_1^\mathsf{T}\,\boldsymbol{S}^\mathsf{T}, \qquad \boldsymbol{V} = \boldsymbol{R}\,\boldsymbol{N}_1\,\boldsymbol{\Sigma}_1^{-1/2}$$



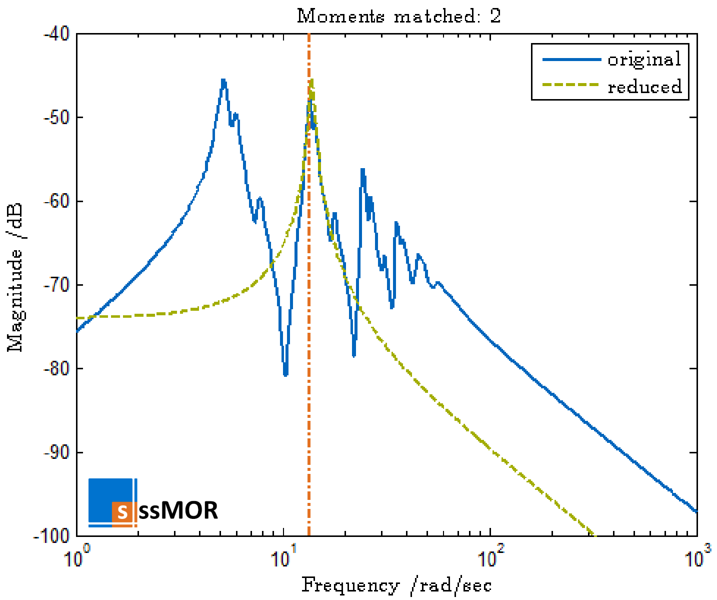
Rational Interpolation by Krylov subspace methods

Moments of a transfer function

$$G(s) = C(sE - A)^{-1}B$$
$$= G(\Delta s + \sigma) = \sum_{i=0}^{\infty} M_i(\sigma)\,(s-\sigma)^i$$

σ : interpolation point (shift)

$M_i(\sigma)$: i-th moment around σ



(Multi)-Moment Matching by Rational Krylov (RK) subspaces

Bases for input and output Krylov-subspaces:

$$\mathrm{Ran}(V) \supseteq \mathrm{span}\left\{A_\sigma^{-1}B,\; A_\sigma^{-1}EA_\sigma^{-1}B,\; \ldots,\; (A_\sigma^{-1}E)^{r-1}A_\sigma^{-1}B\right\}$$

$$\mathrm{Ran}(W) \supseteq \mathrm{span}\left\{A_\sigma^{-\mathsf{T}}C^\mathsf{T},\; A_\sigma^{-\mathsf{T}}E^\mathsf{T}A_\sigma^{-\mathsf{T}}C^\mathsf{T},\; \ldots,\; (A_\sigma^{-\mathsf{T}}E^\mathsf{T})^{r-1}A_\sigma^{-\mathsf{T}}C^\mathsf{T}\right\}$$

$\Rightarrow$ $M_i(\sigma) = M_{\mathrm{r},i}(\sigma)$

for $i = 0, \ldots, 2r-1$

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

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

$\mathcal{H}_2$-optimal model order reduction

Goal: Find ROM that minimizes the $\mathcal{H}_2$-error

$$\|G - G_\mathrm{r}\|_{\mathcal{H}_2} = \min_{\dim(\widetilde{G}_\mathrm{r})=r} \left\|G - \widetilde{G}_\mathrm{r}\right\|_{\mathcal{H}_2}$$

H2-norm: $\|G(s)\|^2_{\mathcal{H}_2} = \dfrac{1}{2\pi}\displaystyle\int_{-\infty}^{\infty} |G(\mathrm{i}\omega)|^2 \mathrm{d}\omega$

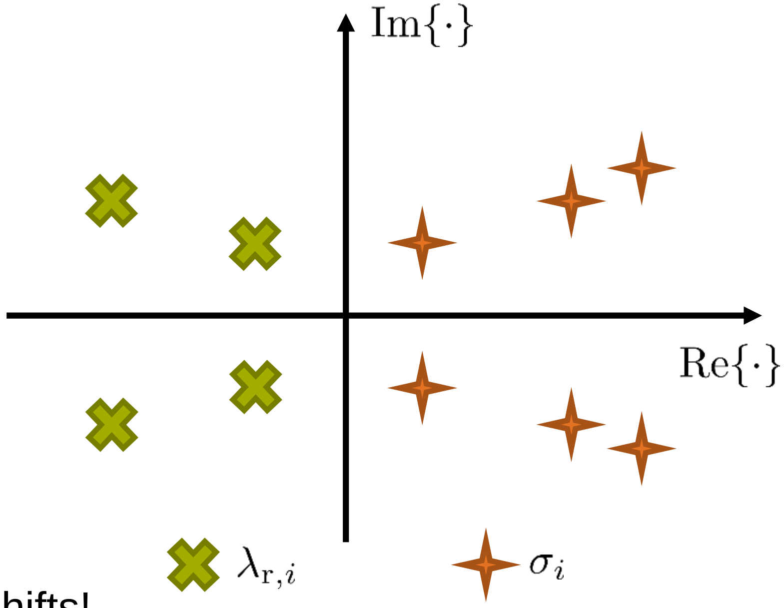
Necessary optimality conditions:

$$G(-\overline{\lambda}_{\mathrm{r},i}) = G_\mathrm{r}(-\overline{\lambda}_{\mathrm{r},i})$$
$$G'(-\overline{\lambda}_{\mathrm{r},i}) = G'_\mathrm{r}(-\overline{\lambda}_{\mathrm{r},i})$$
$$i = 1, \dots, r$$

Algorithm 1 Iterative Rational Krylov Algorithm (SISO)

Input: $\Sigma := (\boldsymbol{E}, \boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}^\mathsf{T})$, σ_i, tol
Output: locally $\mathcal{H}_2$-optimal ROM $\Sigma_\mathrm{r}^\mathrm{opt}$, σ_i^opt

while (relative change of $\sigma_i >$ tol) **do**
  $\Sigma_\mathrm{r} \leftarrow \mathrm{RK}(\Sigma, \sigma_i)$ ▷ Rational Krylov reduction
  $\boldsymbol{\Lambda}_\mathrm{r} = \lambda(\boldsymbol{A}_\mathrm{r}, \boldsymbol{E}_\mathrm{r})$ ▷ Compute eigenvalues of ROM
  $\sigma_i \leftarrow -\overline{\lambda}_{\mathrm{r},i}$ ▷ Mirror reduced eigenvalues
end while
$\Sigma_\mathrm{r}^\mathrm{opt} \leftarrow \Sigma_\mathrm{r}$, $\sigma_i^\mathrm{opt} \leftarrow \sigma_i$ ▷ Return optimal ROM and shifts



IRKA achieves multipoint moment matching at optimal shifts!

Comparison: BT vs. Krylov subspace methods

Balanced Truncation (BT)

- \+ stability preservation
- \+ automatable
- \+ error bound (a priori)

- – computing-intensive
- – storage-intensive
- – n < 5000

Rational Krylov (RK) subspaces

- \+ numerically efficient
- \+ $n \approx 10^6$
- \+ H_2-optimal (IRKA)
- \+ many degrees of freedom

- – many degrees of freedom
- – stability gen. not preserved
- – no error bounds

Subject of research

- Numerically efficient solution of large-scale Lyapunov equations

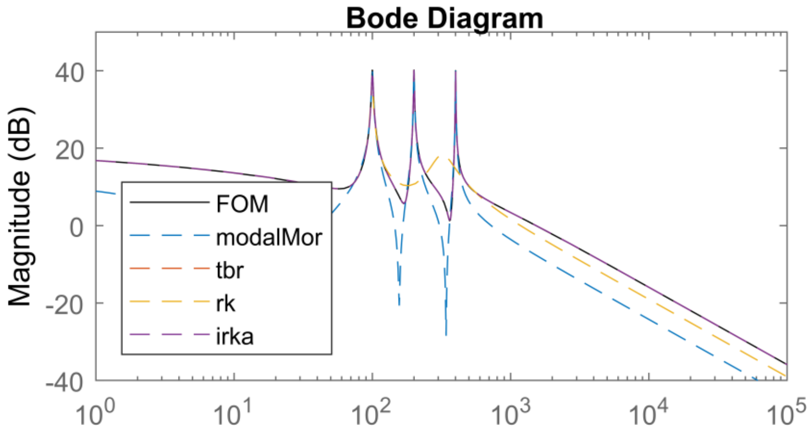
⇨ Krylov-based Low-Rank Approximation
  - ADI (Alternating Directions Implicit)
  - RKSM (Rational Krylov Subspace Method)

Subject of research

- Adaptive choice of reduction parameters
  - Reduced order
  - Interpolation data (shifts, etc.)
- Stability preservation
- Numerically efficient computation of rigorous error bounds

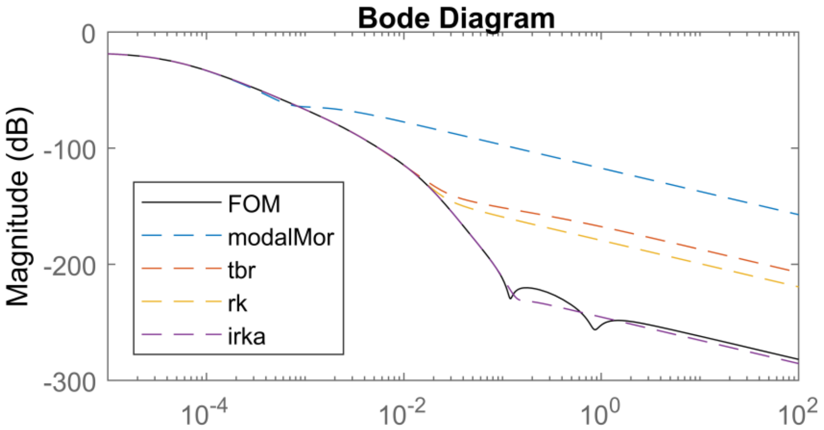
Numerical comparison

fom: $n = 1006, r = 20$



	red. time [s]	$\frac{\|G-G_\mathrm{r}\|_{\mathcal{H}_2}}{\|G\|_{\mathcal{H}_2}}$	$\frac{\|G-G_\mathrm{r}\|_{\mathcal{H}_\infty}}{\|G\|_{\mathcal{H}_\infty}}$
modalMor (lr)	0.40	19.40e-02	4.16e-02
tbr	0.20	1.18e-09	5.78e-09
rk	0.09	81.47e-02	96.73e-02
irka	0.60	8.56e-08	5.80e-09

steel profile rail_1357: $n = 1357, r = 20$



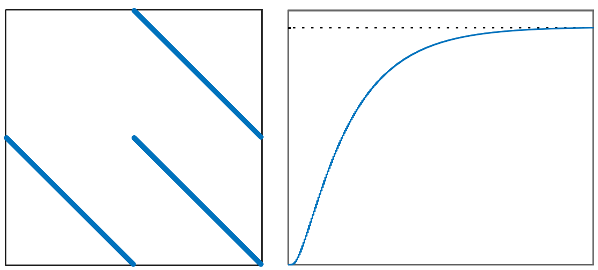
	red. time [s]	$\frac{\|G-G_\mathrm{r}\|_{\mathcal{H}_2}}{\|G\|_{\mathcal{H}_2}}$	$\frac{\|G-G_\mathrm{r}\|_{\mathcal{H}_\infty}}{\|G\|_{\mathcal{H}_\infty}}$
modalMor (lr)	1.21	4.61e-02	3.76e-03
tbr	0.49	3.47e-05	2.65e-06
rk	0.10	1.34e-07	3.36e-07
irka	1.32	2.38e-12	9.61e-11

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

Toolboxes for sparse, large-scale models in

sss

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

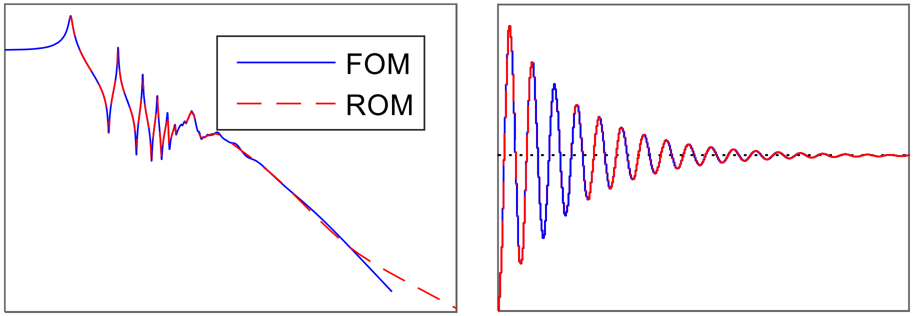


```
bode(sys), sigma(sys)
step(sys), impulse(sys)
norm(sys,2), norm(sys,inf)

c2d, lsim, eigs, connect,…
```

sssMOR

```
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 and https://github.com/MORLab.
[Castagnotto/Cruz Varona/Jeschek/Lohmann '17]: **„sss & sssMOR: Analysis and Reduction of Large-Scale Dynamic Systems in MATLAB“,** at-Automatisierungstechnik]

GitHub

Main characteristics

sss

- **State-space models** of very high order on a standard computer $(n \approx 10^8)$
- 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**

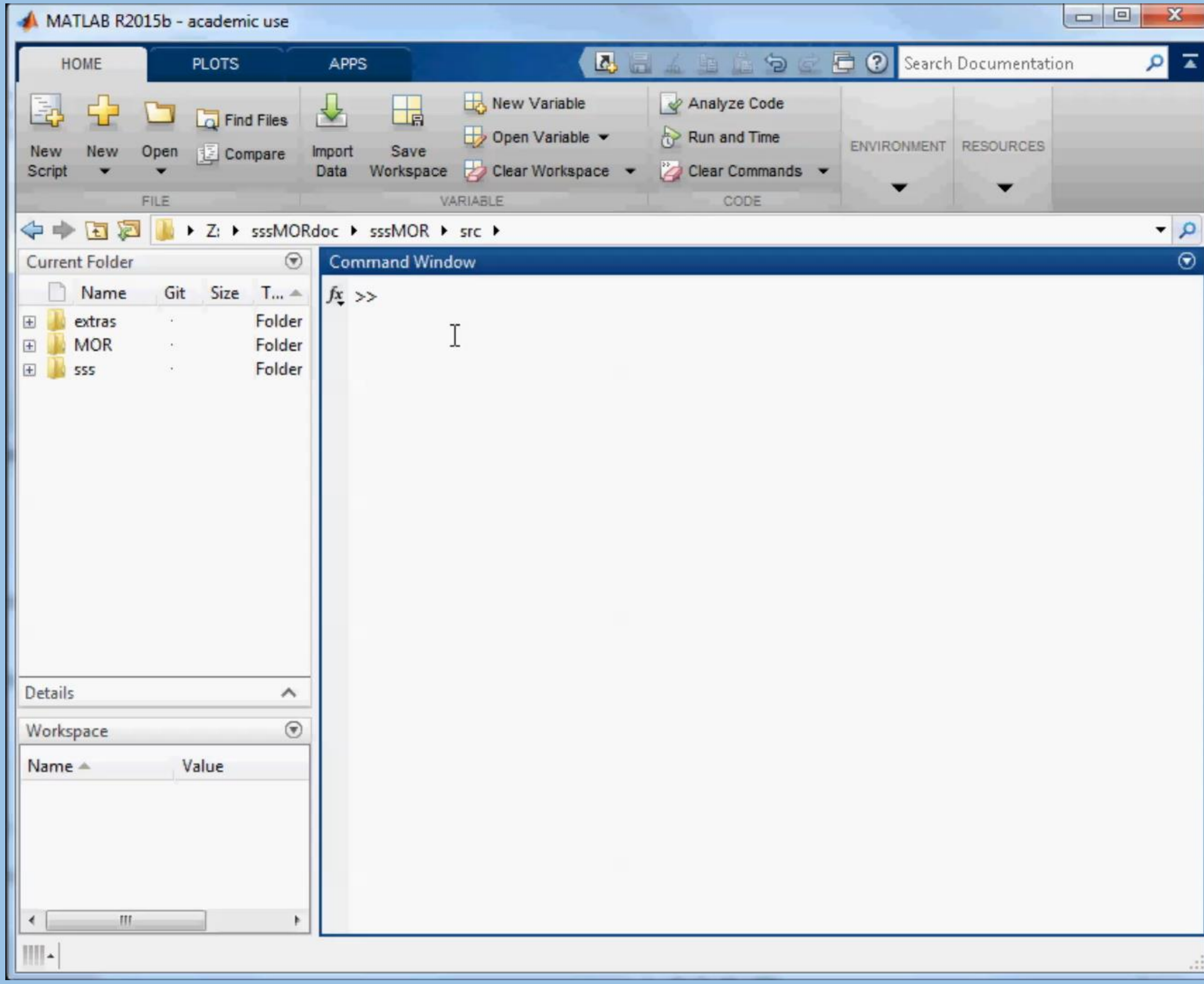
Chair of Automatic Control
Department of Mechanical Engineering
Technical University of Munich

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

sss sssMOR

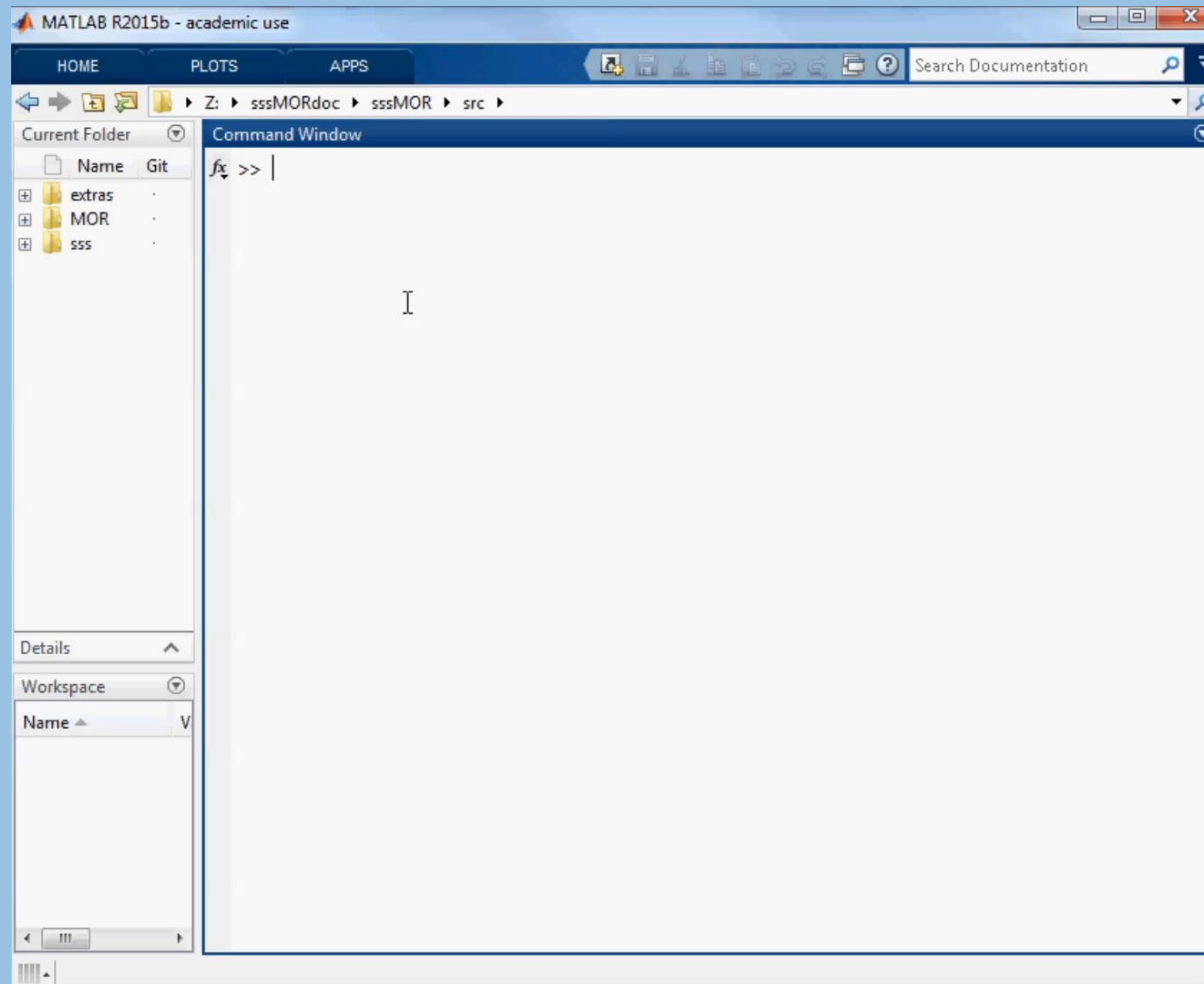
Comprehensive **documentation** with examples and references

sssMOR App
graphical user interface

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

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

sss sssMOR



Comprehensive **documentation** with examples and references

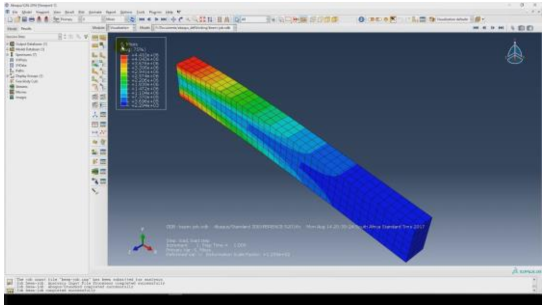
sssMOR App
graphical user interface

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

FEM & MOR software

Commercial FEM software:

ANSYS, Abaqus, COMSOL Multiphysics, LS-DYNA, Nastran, …

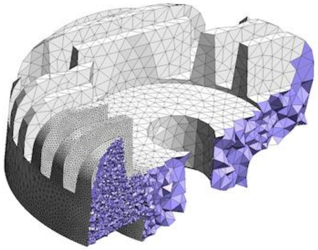


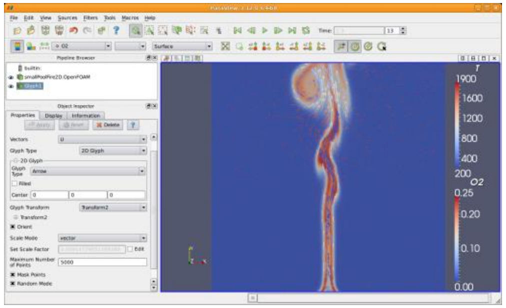
Open-source FEM software:

AMfe, CalculiX, FEniCS Project, FreeFEM++, JuliaFEM, KRATOS, OOFEM, OpenFOAM, …

Open-source Pre-/Post-Processing tools:

Gmsh, ParaView, …





Open-source MOR software:

pyMOR, sss, sssMOR, psssMOR, emgr, M.E.S.S., MOREMBS, MORE, RBmatlab, …

Polynomial & Nonlinear Model Order Reduction

Projective MOR for Nonlinear Systems

Given a large-scale nonlinear control system of the form

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

$$\boldsymbol{E}\,\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, u)$$
$$y = h(\boldsymbol{x})$$

$$\boldsymbol{x}(t) \in \mathbb{R}^n$$

with $\boldsymbol{f}(\boldsymbol{x},\, u) : \mathbb{R}^n \times \mathbb{R}^1 \to \mathbb{R}^n$ and $h(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}^1$

Simulation, design, control and optimization cannot be done efficiently!

MOR

$$\boldsymbol{x} \approx \boldsymbol{V}\,\boldsymbol{x}_{\mathrm{r}}\,, \quad \boldsymbol{V} \in \mathbb{R}^{n \times r}$$

Reduced order model (ROM)

$$\underbrace{\boldsymbol{V}^{\mathsf{T}}\;\boldsymbol{E}\;\boldsymbol{V}}_{\boldsymbol{E}_{\mathrm{r}}}\;\dot{\boldsymbol{x}}_{\mathrm{r}} = \underbrace{\boldsymbol{V}^{\mathsf{T}}\;\boldsymbol{f}(\boldsymbol{V}\,\boldsymbol{x}_{\mathrm{r}},\, u)}_{\boldsymbol{f}_{\mathrm{r}}(\boldsymbol{x}_{\mathrm{r}},\, u)}$$
$$y_{\mathrm{r}} = \underbrace{h(\boldsymbol{V}\,\boldsymbol{x}_{\mathrm{r}})}_{h(\boldsymbol{x}_{\mathrm{r}})}$$

$$\boldsymbol{x}_{\mathrm{r}}(t) \in \mathbb{R}^r,\; r \ll n$$

Goal:
$$y_{\mathrm{r}}(t) \approx y(t)$$

with $\boldsymbol{f}_{\mathrm{r}}(\boldsymbol{x}_{\mathrm{r}},\, u) : \mathbb{R}^r \times \mathbb{R}^1 \to \mathbb{R}^r$ and $h(\boldsymbol{x}_{\mathrm{r}}) : \mathbb{R}^r \to \mathbb{R}^1$

Challenges of Nonlinear MOR

Nonlinear systems can exhibit complex behaviours

- Strong nonlinearities
- Multiple equilibrium points
- Limit cycles
- Chaotic behaviours

Input-output behaviour of nonlinear systems **cannot** be described with transfer functions, the state-transition matrix or the convolution integral (only possible for special cases)

Choice of the reduced order basis

- Projection basis should comprise most dominant directions of the state-space
- Different existing approaches:
  - Simulation-based methods
  - System-theoretic techniques

Expensive evaluation of $\boldsymbol{f}(\boldsymbol{V}\boldsymbol{x}_\mathrm{r})$

- Vector of nonlinearities **f** still has to be evaluated in full dimension
- Approximation of **f** by so-called hyper-reduction techniques:
  → EIM, DEIM, GNAT, ECSW…

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

Nonlinear MOR methods – Overview

Polynomial nonlinear systems

Reduction of bilinear systems

$$\boldsymbol{E}\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{N}\boldsymbol{x}u + \boldsymbol{b}u$$
$$y = \boldsymbol{c}^{\mathsf{T}}\boldsymbol{x}$$

- ☑ Transfer of system-theoretic concepts
- ☑ Generalization of linear MOR methods:
  - Balanced truncation
  - **Krylov / $\mathcal{H}_2$-optimal approach**

Reduction of quadratic-bilinear systems

$$\boldsymbol{E}\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{H}(\boldsymbol{x} \otimes \boldsymbol{x}) + \boldsymbol{N}\boldsymbol{x}u + \boldsymbol{b}u$$
$$y = \boldsymbol{c}^{\mathsf{T}}\boldsymbol{x}$$

- ☑ Reduction methods for **MIMO** models
- ➢ Input-awareness:
  - signal generators
  - eigenfunctions

Nonlinear systems

Reduction of nonlinear (parametric) systems

$$\boldsymbol{E}\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, u) \qquad \boldsymbol{E}\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\, u$$
$$y = h(\boldsymbol{x}) \qquad y = \boldsymbol{c}^{\mathsf{T}}\boldsymbol{x}$$

- ☑ Simulation-based:
  - POD, TPWL
  - Reduced Basis, Empirical Gramians
- ➢ Simulation-free / System-theoretic

Polynomial Nonlinear Systems

Polynomialization / Carleman linearization

Starting point: $\boldsymbol{E}\,\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\,u$

$$y = \boldsymbol{c}^{\mathsf{T}}\boldsymbol{x}$$

Power series

Assumptions:

- $\boldsymbol{x}_{\mathrm{S}} = \boldsymbol{0}$
- $\boldsymbol{f}(\boldsymbol{x}_{\mathrm{S}}) = \boldsymbol{0}$

$$\boldsymbol{A}^{(1)} \in \mathbb{R}^{n \times n}$$
$$\boldsymbol{A}^{(2)} \in \mathbb{R}^{n \times n^2}$$
$$\boldsymbol{A}^{(3)} \in \mathbb{R}^{n \times n^3}$$
$$\vdots$$

$$\boldsymbol{x}^{(1)} = \boldsymbol{x} \in \mathbb{R}^{n}$$
$$\boldsymbol{x}^{(2)} = \boldsymbol{x} \otimes \boldsymbol{x} \in \mathbb{R}^{n^2}$$
$$\boldsymbol{x}^{(3)} = \boldsymbol{x} \otimes \boldsymbol{x} \otimes \boldsymbol{x} \in \mathbb{R}^{n^3}$$
$$\vdots$$

$$\boldsymbol{E}\,\dot{\boldsymbol{x}} = \boldsymbol{A}^{(1)}\,\boldsymbol{x} + \boldsymbol{A}^{(2)}\,(\boldsymbol{x} \otimes \boldsymbol{x}) + \cdots + \boldsymbol{N}^{(1)}\,\boldsymbol{x}\,u + \cdots + \boldsymbol{b}\,u$$
$$y = \boldsymbol{c}^{\mathsf{T}}\,\boldsymbol{x}$$

Bilinear dynamical systems

- Result from direct modeling or Carleman (bi)linearization
- Linear in input and linear in state, but not jointly linear in both
- Interface between fully nonlinear and linear systems

$$\boldsymbol{E}\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{N}\boldsymbol{x}u + \boldsymbol{b}u$$
$$y = \boldsymbol{c}^{\mathsf{T}}\boldsymbol{x}$$

Volterra series representation

[Rugh '81]

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\,\boldsymbol{x}(t) + \boldsymbol{N}\,\boldsymbol{x}(t)\,u(t) + \boldsymbol{b}\,u(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0,$$
$$y(t) = \boldsymbol{c}^{\mathsf{T}}\,\boldsymbol{x}(t).$$

Picard fixed-point iteration (successive approximation)

Approximate solution of the bilinear system

$$\boldsymbol{x}_1(t) = \int_{\tau=0}^{t} \mathrm{e}^{\boldsymbol{A}(t-\tau)}\boldsymbol{b}u(\tau)\,\mathrm{d}\tau + \mathrm{e}^{\boldsymbol{A}t}\boldsymbol{x}_0\,,$$

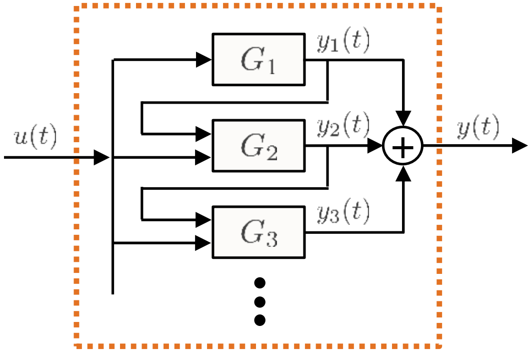
$$\boldsymbol{x}_k(t) = \int_{\tau=0}^{t} \mathrm{e}^{\boldsymbol{A}(t-\tau)}\boldsymbol{N}u(\tau)\boldsymbol{x}_{k-1}(\tau)\,\mathrm{d}\tau\,, \quad k \geq 2\,.$$

$$\boldsymbol{x}(t) = \sum_{k=1}^{\infty} \boldsymbol{x}_k(t)$$

Variational equations (subsystems)

Interpretation as a series of homogenous, cascaded subsystems:

$$\dot{\boldsymbol{x}}_1(t) = \boldsymbol{A}\,\boldsymbol{x}_1(t) + \boldsymbol{b}\,u(t)\,, \qquad \boldsymbol{x}_1(0) = \boldsymbol{x}_0\,,$$
$$\dot{\boldsymbol{x}}_k(t) = \boldsymbol{A}\,\boldsymbol{x}_k(t) + \boldsymbol{N}\,\boldsymbol{x}_{k-1}(t)\,u(t)\,, \quad \boldsymbol{x}_k(0) = \boldsymbol{0}\,, \quad k \geq 2$$



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

Systems Theory for Volterra systems (1)

[Rugh '81]

Input-Output behavior

$$y(t) = \sum_{k=1}^{\infty} y_k(t)$$

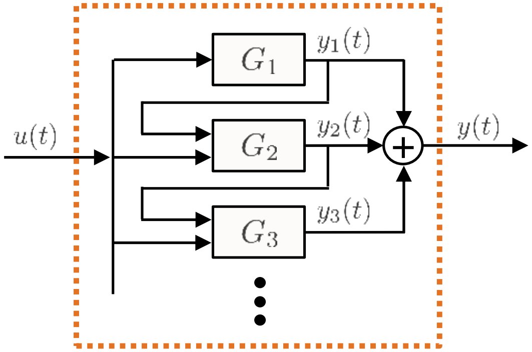
$$y(t) = \sum_{k=1}^{\infty} \int_{\tau_1=-\infty}^{\infty} \cdots \int_{\tau_k=-\infty}^{\infty} \underbrace{\boldsymbol{c}^\mathsf{T} \mathrm{e}^{\boldsymbol{A}\tau_k} \boldsymbol{N} \cdots \boldsymbol{N} \mathrm{e}^{\boldsymbol{A}\tau_2} \boldsymbol{N} \mathrm{e}^{\boldsymbol{A}\tau_1} \boldsymbol{b}}_{g_k(\tau_1,\ldots,\tau_k)} \times u(t-\tau_k) \cdots u(t-\tau_k-\ldots-\tau_1)\, \mathrm{d}\tau_k \cdots \mathrm{d}\tau_1$$

Kernels

$$k=1: \quad g_1(\tau_1) = \boldsymbol{c}^\mathsf{T} \mathrm{e}^{\boldsymbol{A}\tau_1} \boldsymbol{b}$$

$$k=2: \quad g_2(\tau_1,\tau_2) = \boldsymbol{c}^\mathsf{T} \mathrm{e}^{\boldsymbol{A}\tau_2} \boldsymbol{N} \mathrm{e}^{\boldsymbol{A}\tau_1} \boldsymbol{b}$$

$$k=3: \quad g_3(\tau_1,\tau_2,\tau_3) = \boldsymbol{c}^\mathsf{T} \mathrm{e}^{\boldsymbol{A}\tau_3} \boldsymbol{N} \mathrm{e}^{\boldsymbol{A}\tau_2} \boldsymbol{N} \mathrm{e}^{\boldsymbol{A}\tau_1} \boldsymbol{b}$$



Transfer functions

$$k=1: \quad G_1(s_1) = \boldsymbol{c}^\mathsf{T} (s_1 \mathbf{I} - \boldsymbol{A})^{-1} \boldsymbol{b}$$

$$k=2: \quad G_2(s_1,s_2) = \boldsymbol{c}^\mathsf{T} (s_2 \mathbf{I} - \boldsymbol{A})^{-1} \boldsymbol{N} (s_1 \mathbf{I} - \boldsymbol{A})^{-1} \boldsymbol{b}$$

$$k=3: \quad G_3(s_1,s_2,s_3) = \boldsymbol{c}^\mathsf{T} (s_3 \mathbf{I} - \boldsymbol{A})^{-1} \boldsymbol{N} (s_2 \mathbf{I} - \boldsymbol{A})^{-1} \boldsymbol{N} (s_1 \mathbf{I} - \boldsymbol{A})^{-1} \boldsymbol{b}$$

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

Systems Theory for Volterra systems (2)

[Rugh ’81]

Gramians

$$\boldsymbol{P} = \sum_{k=1}^{\infty} \boldsymbol{P}_k\,, \quad \boldsymbol{Q} = \sum_{k=1}^{\infty} \boldsymbol{Q}_k\,,$$

$$g_k(\tau_1,\ldots,\tau_k) = \underbrace{\boldsymbol{c}^\mathsf{T} \mathrm{e}^{\boldsymbol{A}\tau_k} \boldsymbol{N} \cdots \overbrace{\boldsymbol{N} \mathrm{e}^{\boldsymbol{A}\tau_2} \boldsymbol{N} \mathrm{e}^{\boldsymbol{A}\tau_1} \boldsymbol{b}}^{\overline{\boldsymbol{p}}_k(\tau_1,\ldots,\tau_k)}}_{\overline{\boldsymbol{q}}_k(\tau_1,\ldots,\tau_k)^\mathsf{T}}$$

$$\boldsymbol{P}_k = \int_{\tau_1=0}^{\infty} \cdots \int_{\tau_k=0}^{\infty} \overline{\boldsymbol{p}}_k(\tau_1,\ldots,\tau_k)\overline{\boldsymbol{p}}_k(\tau_1,\ldots,\tau_k)^\mathsf{T}\, \mathrm{d}\tau_1 \cdots \mathrm{d}\tau_\mathrm{k}$$

$$\boldsymbol{Q}_k = \int_{\tau_1=0}^{\infty} \cdots \int_{\tau_k=0}^{\infty} \overline{\boldsymbol{q}}_k(\tau_1,\ldots,\tau_k)\overline{\boldsymbol{q}}_k(\tau_1,\ldots,\tau_k)^\mathsf{T}\, \mathrm{d}\tau_1 \cdots \mathrm{d}\tau_\mathrm{k}$$

$$\boldsymbol{A}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}^\mathsf{T} + \boldsymbol{N}\boldsymbol{P}\boldsymbol{N}^\mathsf{T} + \boldsymbol{b}\boldsymbol{b}^\mathsf{T} = \boldsymbol{0}$$

$$\boldsymbol{A}^\mathsf{T}\boldsymbol{Q} + \boldsymbol{Q}\boldsymbol{A} + \boldsymbol{N}^\mathsf{T}\boldsymbol{Q}\boldsymbol{N} + \boldsymbol{c}\boldsymbol{c}^\mathsf{T} = \boldsymbol{0}$$

H2-norm

$$\|\zeta\|_{\mathcal{H}_2}^2 = \sum_{k=1}^{\infty} \int_{\tau_1=0}^{\infty} \cdots \int_{\tau_k=0}^{\infty} g_k(\tau_1,\ldots,\tau_k) g_k(\tau_1,\ldots,\tau_k)^\mathsf{T} \mathrm{d}\tau_1 \cdots \mathrm{d}\tau_k$$

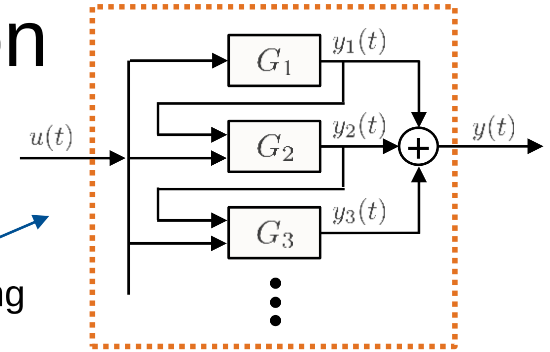
$$= \sum_{k=1}^{\infty} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \frac{1}{(2\pi)^k} G_k(\mathrm{i}\omega_1,\ldots,\mathrm{i}\omega_k)\, G_k(-\mathrm{i}\omega_1,\ldots,-\mathrm{i}\omega_k)^\mathsf{T}\, \mathrm{d}\omega_1 \cdots \mathrm{d}\omega_k$$

$$= \boldsymbol{c}^\mathsf{T}\, \boldsymbol{P}\, \boldsymbol{c} = \boldsymbol{b}^\mathsf{T}\, \boldsymbol{Q}\, \boldsymbol{b}$$

Multipoint Volterra Series Interpolation

Goal: Enforcing multipoint interpolation of the underlying Volterra series

Moment matching



Multipoint Volterra series interpolation [Flagg/Gugercin '15]

Set of interpolation points: $S = \{\sigma_1, \dots, \sigma_r\}, \quad i = 1, \dots, r$

$$\sum_{k=1}^{\infty} \sum_{l_1=1}^{r} \cdots \sum_{l_{k-1}=1}^{r} \eta_{l_1,\dots,l_{k-1},i} \, G_k(\sigma_{l_1}, \dots, \sigma_{l_{k-1}}, \sigma_i) = \sum_{k=1}^{\infty} \sum_{l_1=1}^{r} \cdots \sum_{l_{k-1}=1}^{r} \eta_{l_1,\dots,l_{k-1},i} \, G_{k,\mathrm{r}}(\sigma_{l_1}, \dots, \sigma_{l_{k-1}}, \sigma_i)$$

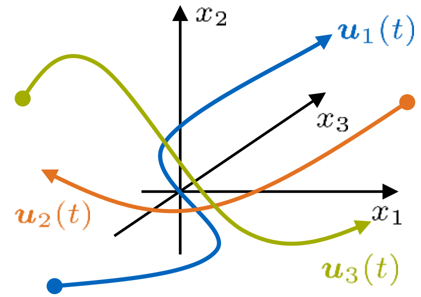
This approach interpolates the weighted series at the interpolation points $\sigma_1, \dots, \sigma_r$

Projection matrices for Volterra series interpolation

$$\boldsymbol{v}_i = \sum_{k=1}^{\infty} \sum_{l_1=1}^{r} \cdots \sum_{l_{k-1}=1}^{r} \eta_{l_1,\dots,l_{k-1},i} (\sigma_i \boldsymbol{E} - \boldsymbol{A})^{-1} \boldsymbol{N} (\sigma_{l_{k-1}} \boldsymbol{E} - \boldsymbol{A})^{-1} \boldsymbol{N} \cdots \boldsymbol{N} (\sigma_{l_1} \boldsymbol{E} - \boldsymbol{A})^{-1} \boldsymbol{b}$$

$$\boldsymbol{w}_i = \sum_{k=1}^{\infty} \sum_{l_1=1}^{r} \cdots \sum_{l_{k-1}=1}^{r} \vartheta_{l_1,\dots,l_{k-1},i} (\mu_{l_1} \boldsymbol{E} - \boldsymbol{A})^{-\mathsf{T}} \boldsymbol{N}^{\mathsf{T}} (\mu_{l_2} \boldsymbol{E} - \boldsymbol{A})^{-\mathsf{T}} \boldsymbol{N}^{\mathsf{T}} \cdots \boldsymbol{N}^{\mathsf{T}} (\mu_i \boldsymbol{E} - \boldsymbol{A})^{-\mathsf{T}} \boldsymbol{c}$$

Proper Orthogonal Decomposition (POD)



Starting point: $\boldsymbol{E}\,\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x})$$

1. Choose suitable training input signals $\boldsymbol{u}_1(t), \boldsymbol{u}_2(t), \ldots, \boldsymbol{u}_t(t)$
2. Take snapshots from simulated full order state trajectories

$$\underset{(n,n_\mathrm{s})}{\boldsymbol{X}} = \left[\boldsymbol{x}^{\boldsymbol{u}_1}(t_1),\, \boldsymbol{x}^{\boldsymbol{u}_1}(t_2),\, \cdots,\, \boldsymbol{x}^{\boldsymbol{u}_1}(t_N) \quad \boldsymbol{x}^{\boldsymbol{u}_2}(t_1),\, \boldsymbol{x}^{\boldsymbol{u}_2}(t_2),\, \cdots\right]$$

3. Perform singular value decomposition (SVD) of snapshot matrix $\boldsymbol{X}$

$$\boldsymbol{X} = \underset{(n,n)}{\boldsymbol{M}}\ \underset{(n,n_\mathrm{s})}{\boldsymbol{\Sigma}}\ \underset{(n_\mathrm{s},n_\mathrm{s})}{\boldsymbol{N}^\top} \approx \underset{(n,r)}{\boldsymbol{M}_\mathrm{r}}\ \underset{(r,n_\mathrm{s})}{\boldsymbol{\Sigma}_\mathrm{r}}\ \underset{(n_\mathrm{s},n_\mathrm{s})}{\boldsymbol{N}_\mathrm{r}^\top}$$

4. Reduced order basis: $\boldsymbol{V} = \boldsymbol{M}_\mathrm{r} \in \mathbb{R}^{n\times r}$

Advantages:

- ☑ Straightforward data-driven method
- ☑ Choice of reduced order from singular values / error bound for approx. error
- ☑ Optimal in least squares sense:

$$\min_{\mathrm{rank}(\boldsymbol{X}_\mathrm{r})=r} ||\boldsymbol{X} - \boldsymbol{X}_\mathrm{r}||_2$$

Disadvantages:

- ! Simulation of full order model for different input signals required
- ! SVD of large snapshot matrix required
- ! Training input dependency

Summary & Outlook

Take-Home Messages:

- Modeling via FEM/FVM is becoming more and more important!
- Applicable for several physical domains and many technical applications!
- Model Order Reduction is indispensable to reduce the computational effort
- Reduction is done via projection
- Linear MOR is well developed
- Generalization of system-theoretic concepts and MOR methods to polynomial systems
- POD is still the most employed nonlinear MOR method
- Simulation-free / Sytem-theoretic nonlinear MOR techniques are aimed

Ongoing work:

- Polynomial nonlinear systems
- Simulation-free / System-theoretic NLMOR

Thank you for your attention!

References

[Antoulas ’05] *Approximation of Large-Scale Dynamical Systems.* SIAM.

[Astolfi ’10] *Model reduction by moment matching for linear and nonlinear systems.* IEEE TAC.

[Beattie/Gugercin ’17] *Model reduction by rational interpolation.* Model Reduction and Algorithms: Theory and Applications. SIAM.

[Chaturantabut et al. ’10] *Nonlinear model reduction via discrete emperical interpolation.* SIAM Journal on Scientific Computing.

[Flagg/Gugercin ’15] *Multipoint Volterra series interpolation and H2 optimal model reduction of bilinear systems*, SIAM Journal on Matrix…

[Isidori ’95] *Nonlinear Control S*ystems. Springer, Third edition.

[Rugh ’81] *Nonlinear system theory. The Volterra/Wiener Approach*. The Johns Hopkins University Press