MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

CSC
COMPUTATIONAL METHODS IN
SYSTEMS AND CONTROL THEORY

Balanced Truncation for Linear and Nonlinear Systems on Industrial Scale

Peter Benner   Igor Pontes Duff   Pawan K. Goyal   Jens Saak

...

Workshop in Industrial Mathematics (WIM 2021)
Catania, October 20–22, 2021

Supported by:

DFG-Graduiertenkolleg
MATHEMATISCHE
KOMPLEXITÄTSREDUKTION

IMPRS
ProEng
Magdeburg

Original System

$$\Sigma : \begin{cases} \dot{x}(t) &= f(t, x(t), u(t)), \\ y(t) &= g(t, x(t), u(t)), \end{cases}$$

- states $x(t) \in \mathbb{R}^n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $y(t) \in \mathbb{R}^p$.

Original System

$$\Sigma : \begin{cases} \dot{x}(t) &= f(t, x(t), u(t)), \\ y(t) &= g(t, x(t), u(t)), \end{cases}$$

- states $x(t) \in \mathbb{R}^n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $y(t) \in \mathbb{R}^p$.

Reduced-Order Model (ROM)

$$\widehat{\Sigma} : \begin{cases} \dot{\hat{x}}(t) &= \widehat{f}(t, \hat{x}(t), u(t)), \\ \hat{y}(t) &= \widehat{g}(t, \hat{x}(t), u(t)), \end{cases}$$

- states $\hat{x}(t) \in \mathbb{R}^r$, $r \ll n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $\hat{y}(t) \in \mathbb{R}^p$.

Original System

$$\Sigma : \begin{cases} \dot{x}(t) &= f(t, x(t), u(t)), \\ y(t) &= g(t, x(t), u(t)), \end{cases}$$

- states $x(t) \in \mathbb{R}^n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $y(t) \in \mathbb{R}^p$.

Reduced-Order Model (ROM)

$$\widehat{\Sigma} : \begin{cases} \dot{\hat{x}}(t) &= \widehat{f}(t, \hat{x}(t), u(t)), \\ \hat{y}(t) &= \widehat{g}(t, \hat{x}(t), u(t)), \end{cases}$$

- states $\hat{x}(t) \in \mathbb{R}^r$, $r \ll n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $\hat{y}(t) \in \mathbb{R}^p$.

Goals:

$\|y - \hat{y}\| <$ tolerance $\cdot \|u\|$ for all admissible input signals.

Introduction

Model Order Reduction of Dynamical Systems

Original System

$$\Sigma : \begin{cases} \dot{x}(t) &= f(t, x(t), u(t)), \\ y(t) &= g(t, x(t), u(t)), \end{cases}$$

- states $x(t) \in \mathbb{R}^n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $y(t) \in \mathbb{R}^p$.

Reduced-Order Model (ROM)

$$\widehat{\Sigma} : \begin{cases} \dot{\hat{x}}(t) &= \widehat{f}(t, \hat{x}(t), u(t)), \\ \hat{y}(t) &= \widehat{g}(t, \hat{x}(t), u(t)), \end{cases}$$

- states $\hat{x}(t) \in \mathbb{R}^r$, $r \ll n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $\hat{y}(t) \in \mathbb{R}^p$.

Goals:

$\|y - \hat{y}\| <$ tolerance $\cdot \|u\|$ for all admissible input signals.

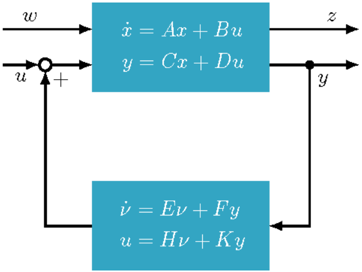
Secondary goal: reconstruct approximation of x from $\hat{x}$.

Feedback Control

A feedback controller (dynamic compensator) is a linear system of order N, where

- input $=$ output of plant,
- output $=$ input of plant.

Modern (LQG-/$\mathcal{H}_2$-/$\mathcal{H}_\infty$-) control design: $N \geq n$.



Sources: MPI Magdeburg (pendulum), https://tinyurl.com/9n75h5dd under CC BY 2.0 license (drone).

Industrial Application Areas

Computer-aided Control System Design (CACSD)

Feedback Control

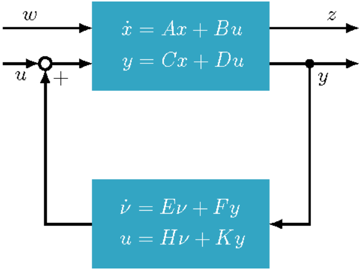
A feedback controller (dynamic compensator) is a linear system of order N, where

- input $=$ output of plant,
- output $=$ input of plant.

Modern (LQG-/$\mathcal{H}_2$-/$\mathcal{H}_\infty$-) control design: $N \geq n$.

Practical controllers require small N ($N \sim 10$, say) due to

- real-time constraints,
- increasing fragility for larger N.



Sources: MPI Magdeburg (pendulum), https://tinyurl.com/9n75h5dd under CC BY 2.0 license (drone).

Feedback Control

A feedback controller (dynamic compensator) is a linear system of order N, where

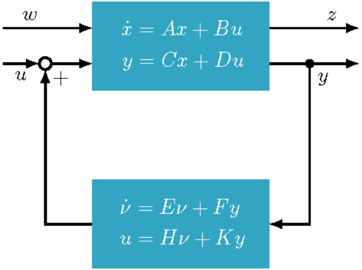
- input $=$ output of plant,
- output $=$ input of plant.

Modern (LQG-/$\mathcal{H}_2$-/$\mathcal{H}_\infty$-) control design: $N \geq n$.

Practical controllers require small N ($N \sim 10$, say) due to

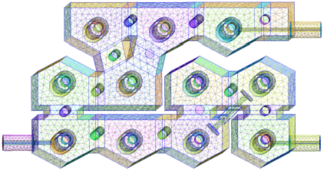
- real-time constraints,
- increasing fragility for larger N.

$\Longrightarrow$ **reduce order of plant (n) and/or controller (N).**

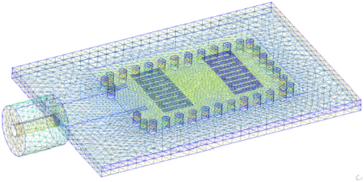


Sources: MPI Magdeburg (pendulum), https://tinyurl.com/9n75h5dd under CC BY 2.0 license (drone).

- Progressive miniaturization: **Moore's Law** states that the number of on-chip transistors doubles each 12 (now: 18) months.



Combline Diplexer:
$n = 270,446$, [Zhao/Wu, IEEE-MTT 2014]

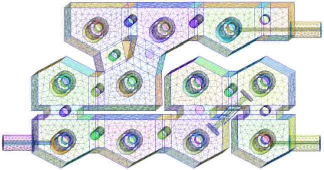


Substrate Integrated Waveguide Antenna:
$n = 390,302$, [Dong/Itoh, IEEE-TAP 2010]

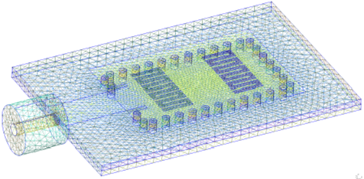
Industrial Application Areas

Nanoelectronics/Electromagnetics

- Progressive miniaturization: **Moore's Law** states that the number of on-chip transistors doubles each 12 (now: 18) months.
- Verification of VLSI/ULSI chip design requires high number of simulations for different input signals.

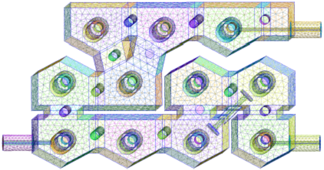


Combline Diplexer:
$n = 270,446$, [Zhao/Wu, IEEE-MTT 2014]

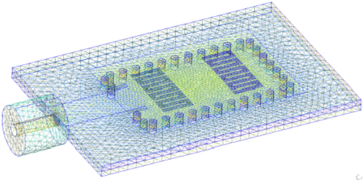


Substrate Integrated Waveguide Antenna:
$n = 390,302$, [Dong/Itoh, IEEE-TAP 2010]

- Progressive miniaturization: **Moore's Law** states that the number of on-chip transistors doubles each 12 (now: 18) months.
- Verification of VLSI/ULSI chip design requires high number of simulations for different input signals.
- Increase in packing density requires modeling of interconnect to ensure that thermic/electro-magnetic effects do not disturb signal transmission.

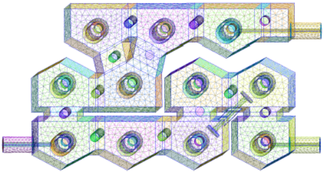


Combline Diplexer:
$n = 270,446$, [Zhao/Wu, IEEE-MTT 2014]

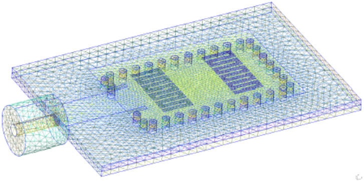


Substrate Integrated Waveguide Antenna:
$n = 390,302$, [Dong/Itoh, IEEE-TAP 2010]

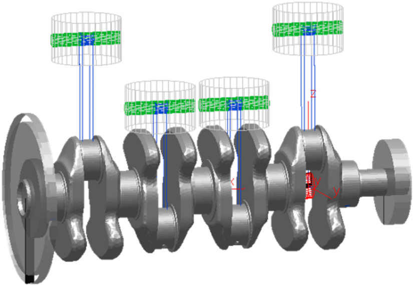
- Progressive miniaturization: **Moore's Law** states that the number of on-chip transistors doubles each 12 (now: 18) months.
- Verification of VLSI/ULSI chip design requires high number of simulations for different input signals.
- Increase in packing density requires modeling of interconnect to ensure that thermic/electro-magnetic effects do not disturb signal transmission.
- Linear systems in nanoelectronics occur through
  - modified nodal analysis (MNA) for RLC networks, e.g., when decoupling large linear subcircuits,
  - modeling transmission lines (interconnect, powergrid), parasitic effects,
  - modeling circuit elements described by Maxwell's equation using partial element equivalent circuits (PEEC), e.g., microwave devices like splitters and diplexers, electromagnetic devices like antennas.



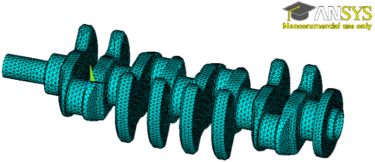
Combline Diplexer:
$n = 270, 446$, [Zhao/Wu, IEEE-MTT 2014]



Substrate Integrated Waveguide Antenna:
$n = 390, 302$, [Dong/Itoh, IEEE-TAP 2010]



$\rightsquigarrow$



- Resolving complex 3D geometries $\Rightarrow$ millions of degrees of freedom.
- Analysis of elastic deformations requires many simulation runs for varying external forces.
- Additional goal: Preserve second-order structure in reduced-order model, i.e.,

$$M\ddot{x}(t) + D\dot{x}(t) + \mathcal{K}(x(t)) = Bu(t)$$

for seamless integration into commercial EMBD simulation software.

System Classes

Control-Affine (Autonomous) Systems

$$\begin{aligned}\dot{x}(t) &= f(t,x,u) &&= \mathcal{A}(x(t)) + \mathcal{B}(x(t))u(t), &&\mathcal{A} : \mathbb{R}^n \to \mathbb{R}^n,\ \mathcal{B} : \mathbb{R}^n \to \mathbb{R}^{n\times m},\\ y(t) &= g(t,x,u) &&= \mathcal{C}(x(t)) + \mathcal{D}(x(t))u(t), &&\mathcal{C} : \mathbb{R}^n \to \mathbb{R}^q,\ \mathcal{D} : \mathbb{R}^n \to \mathbb{R}^{q\times m}.\end{aligned}$$

System Classes

Control-Affine (Autonomous) Systems

$$
\begin{aligned}
\dot{x}(t) &= f(t,x,u) &= \mathcal{A}(x(t)) + \mathcal{B}(x(t))u(t), \quad & \mathcal{A}:\mathbb{R}^n \to \mathbb{R}^n,\ \mathcal{B}:\mathbb{R}^n \to \mathbb{R}^{n\times m},\\
y(t) &= g(t,x,u) &= \mathcal{C}(x(t)) + \mathcal{D}(x(t))u(t), \quad & \mathcal{C}:\mathbb{R}^n \to \mathbb{R}^q,\ \mathcal{D}:\mathbb{R}^n \to \mathbb{R}^{q\times m}.
\end{aligned}
$$

Linear Time-Invariant (LTI) Systems

$$
\begin{aligned}
\dot{x}(t) &= f(t,x,u) &= Ax(t) + Bu(t), \qquad & A \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times m},\\
y(t) &= g(t,x,u) &= Cx(t) + Du(t), \qquad & C \in \mathbb{R}^{q\times n},\ D \in \mathbb{R}^{q\times m}.
\end{aligned}
$$

Control-Affine (Autonomous) Systems

$$\begin{aligned} \dot{x}(t) &= f(t,x,u) = \mathcal{A}(x(t)) + \mathcal{B}(x(t))u(t), \quad \mathcal{A}: \mathbb{R}^n \to \mathbb{R}^n,\ \mathcal{B}: \mathbb{R}^n \to \mathbb{R}^{n\times m}, \\ y(t) &= g(t,x,u) = \mathcal{C}(x(t)) + \mathcal{D}(x(t))u(t), \quad \mathcal{C}: \mathbb{R}^n \to \mathbb{R}^q,\ \mathcal{D}: \mathbb{R}^n \to \mathbb{R}^{q\times m}. \end{aligned}$$

Linear Time-Invariant (LTI) Systems

$$\begin{aligned} \dot{x}(t) &= f(t,x,u) = Ax(t) + Bu(t), \quad A \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times m}, \\ y(t) &= g(t,x,u) = Cx(t) + Du(t), \quad C \in \mathbb{R}^{q\times n},\ D \in \mathbb{R}^{q\times m}. \end{aligned}$$

Bilinear Systems

$$\begin{aligned} \dot{x}(t) &= f(t,x,u) = Ax(t) + \sum_{i=1}^{m} u_i(t)A_i x(t) + Bu(t), \quad A, A_i \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times m}, \\ y(t) &= g(t,x,u) = Cx(t) + Du(t), \quad C \in \mathbb{R}^{q\times n},\ D \in \mathbb{R}^{q\times m}. \end{aligned}$$

System Classes

Linear Time-Invariant (LTI) Systems

$$
\begin{aligned}
\dot{x}(t) &= f(t,x,u) = Ax(t) + Bu(t), \qquad A \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times m},\\
y(t) &= g(t,x,u) = Cx(t) + Du(t), \qquad C \in \mathbb{R}^{q\times n},\ D \in \mathbb{R}^{q\times m}.
\end{aligned}
$$

Bilinear Systems

$$
\begin{aligned}
\dot{x}(t) &= f(t,x,u) = Ax(t) + \sum_{i=1}^{m} u_i(t)A_i x(t) + Bu(t), \qquad A, A_i \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times m},\\
y(t) &= g(t,x,u) = Cx(t) + Du(t), \qquad C \in \mathbb{R}^{q\times n},\ D \in \mathbb{R}^{q\times m}.
\end{aligned}
$$

Quadratic-Bilinear (QB) Systems

$$
\begin{aligned}
\dot{x}(t) &= f(t,x,u) = Ax(t) + H\left(x(t)\otimes x(t)\right) + \sum_{i=1}^{m} u_i(t)A_i x(t) + Bu(t),\\
&\qquad A, A_i \in \mathbb{R}^{n\times n},\ H \in \mathbb{R}^{n\times n^2},\ B \in \mathbb{R}^{n\times m},\\
y(t) &= g(t,x,u) = Cx(t) + Du(t), \qquad C \in \mathbb{R}^{q\times n},\ D \in \mathbb{R}^{q\times m}.
\end{aligned}
$$

System Classes

Quadratic-Bilinear (QB) Systems

$$\begin{aligned}
\dot{x}(t) &= f(t,x,u) = Ax(t) + H\left(x(t) \otimes x(t)\right) + \textstyle\sum_{i=1}^{m} u_i(t) A_i x(t) + Bu(t), \\
&\qquad\qquad A, A_i \in \mathbb{R}^{n \times n},\ H \in \mathbb{R}^{n \times n^2},\ B \in \mathbb{R}^{n \times m}, \\
y(t) &= g(t,x,u) = Cx(t) + Du(t), \qquad C \in \mathbb{R}^{q \times n},\ D \in \mathbb{R}^{q \times m}.
\end{aligned}$$

Polynomial Systems

$$\begin{aligned}
\dot{x}(t) &= f(t,x,u) = Ax(t) + \sum_{j=2}^{n_p} H_j \left(\otimes^j x(t)\right) + \sum_{j=2}^{n_p} \sum_{k=1}^{m} A_j^k \left(\otimes^j x(t)\right) u_k(t) + Bu(t), \\
&\qquad\qquad H_j, A_j^k \text{ of "appropriate size"}, \\
y(t) &= g(t,x,u) = Cx(t) + Du(t), \qquad C \in \mathbb{R}^{q \times n},\ D \in \mathbb{R}^{q \times m}.
\end{aligned}$$

Polynomial-bilinear systems can be obtained from smooth nonlinear systems by *lifting* without approximation error! [Gu 2011].

Control-Affine (Autonomous) Systems

$$\begin{aligned} \dot{x}(t) &= f(t,x,u) &= \mathcal{A}(x(t)) + \mathcal{B}(x(t))u(t), \quad \mathcal{A}: \mathbb{R}^n \to \mathbb{R}^n,\ \mathcal{B}: \mathbb{R}^n \to \mathbb{R}^{n\times m}, \\ y(t) &= g(t,x,u) &= \mathcal{C}(x(t)) + \mathcal{D}(x(t))u(t), \quad \mathcal{C}: \mathbb{R}^n \to \mathbb{R}^q,\ \mathcal{D}: \mathbb{R}^n \to \mathbb{R}^{q\times m}. \end{aligned}$$

Quadratic-Bilinear (QB) Systems

$$\begin{aligned} \dot{x}(t) &= f(t,x,u) &= Ax(t) + H\left(x(t)\otimes x(t)\right) + \textstyle\sum_{i=1}^m u_i(t)A_i x(t) + Bu(t), \\ & & A, A_i \in \mathbb{R}^{n\times n},\ H \in \mathbb{R}^{n\times n^2},\ B \in \mathbb{R}^{n\times m}, \\ y(t) &= g(t,x,u) &= Cx(t) + Du(t), \qquad C \in \mathbb{R}^{q\times n},\ D \in \mathbb{R}^{q\times m}. \end{aligned}$$

Written in control-affine form:

$$\begin{aligned} \mathcal{A}(x) &:= Ax + H\left(x\otimes x\right), \qquad & \mathcal{B}(x) &:= \left[A_1,\ldots,A_m\right]\left(I_m \otimes x\right) + B \\ \mathcal{C}(x) &:= Cx, & \mathcal{D}(x) &:= D. \end{aligned}$$

Control-Affine (Autonomous) Systems

$$\begin{array}{rclcll} \dot{x}(t) & = & f(t,x,u) & = & \mathcal{A}(x(t)) + \mathcal{B}(x(t))u(t), & \mathcal{A}:\mathbb{R}^n \to \mathbb{R}^n,\ \mathcal{B}:\mathbb{R}^n \to \mathbb{R}^{n\times m}, \\ y(t) & = & g(t,x,u) & = & \mathcal{C}(x(t)) + \mathcal{D}(x(t))u(t), & \mathcal{C}:\mathbb{R}^n \to \mathbb{R}^q,\ \mathcal{D}:\mathbb{R}^n \to \mathbb{R}^{q\times m}. \end{array}$$

Quadratic-Bilinear (QB) Systems

$$\begin{array}{rclcl} \dot{x}(t) & = & f(t,x,u) & = & Ax(t) + H\left(x(t)\otimes x(t)\right) + \sum_{i=1}^{m} u_i(t)A_i x(t) + Bu(t), \\ & & & & \qquad A, A_i \in \mathbb{R}^{n\times n},\ H \in \mathbb{R}^{n\times n^2},\ B \in \mathbb{R}^{n\times m}, \\ y(t) & = & g(t,x,u) & = & Cx(t) + Du(t), \qquad C \in \mathbb{R}^{q\times n},\ D \in \mathbb{R}^{q\times m}. \end{array}$$

Written in control-affine form:

$$\begin{array}{rclrcl} \mathcal{A}(x) & := & Ax + H\left(x\otimes x\right), & \mathcal{B}(x) & := & \left[A_1,\ldots,A_m\right]\left(I_m \otimes x\right) + B \\ \mathcal{C}(x) & := & Cx, & \mathcal{D}(x) & := & D. \end{array}$$

Here, we focus on linear and polynomial systems.

Original System

$$\Sigma : \begin{cases} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t). \end{cases}$$

- states $x(t) \in \mathbb{R}^n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $y(t) \in \mathbb{R}^p$.

Reduced-Order Model (ROM)

$$\widehat{\Sigma} : \begin{cases} \dot{\hat{x}}(t) &= \widehat{A}\hat{x}(t) + \widehat{B}u(t), \\ \hat{y}(t) &= \widehat{C}\hat{x}(t) + \widehat{D}u(t). \end{cases}$$

- states $\hat{x}(t) \in \mathbb{R}^r$, $r \ll n$
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $\hat{y}(t) \in \mathbb{R}^p$.

Goals:

$\|y - \hat{y}\| <$ tolerance $\cdot \|u\|$ for all admissible input signals.

Secondary goal: reconstruct approximation of x from $\hat{x}$.

Linear Systems in Frequency Domain

Application of Laplace transform $(x(t) \mapsto x(s),\ \dot{x}(t) \mapsto sX(s) - x(0))$ to LTI system

$$\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t)$$

with $x(0) = 0$ yields:

$$sX(s) = AX(s) + BU(s), \quad Y(s) = CX(s) + DU(s),$$

Linear Systems in Frequency Domain

Application of Laplace transform $(x(t) \mapsto x(s),\ \dot{x}(t) \mapsto sX(s) - x(0))$ to LTI system

$$\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t)$$

with $x(0) = 0$ yields:

$$sX(s) = AX(s) + BU(s), \quad Y(s) = CX(s) + DU(s),$$

$\Longrightarrow$ I/O-relation in frequency domain:

$$Y(s) = \Big(\underbrace{C(sI_n - A)^{-1}B + D}_{=:G(s)} \Big) U(s).$$

$G(s)$ is the **transfer function** of Σ.

Linear Systems in Frequency Domain

Application of Laplace transform $(x(t) \mapsto x(s),\ \dot{x}(t) \mapsto sX(s) - x(0))$ to LTI system

$$\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t) + Du(t)$$

with $x(0) = 0$ yields:

$$sX(s) = AX(s) + BU(s), \quad Y(s) = CX(s) + DU(s),$$

$\Longrightarrow$ I/O-relation in frequency domain:

$$Y(s) = \Big(\underbrace{C(sI_n - A)^{-1}B + D}_{=:G(s)}\Big)U(s).$$

$G(s)$ is the **transfer function** of Σ.

Model reduction in frequency domain: Fast evaluation of mapping $U \to Y$.

Formulating model reduction in frequency domain

Approximate the time domain dynamical system

$$\begin{array}{rcll} \dot{x} & = & Ax + Bu, & A \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times m}, \\ y & = & Cx + Du, & C \in \mathbb{R}^{p\times n},\ D \in \mathbb{R}^{p\times m}, \end{array}$$

by reduced-order system

$$\begin{array}{rcll} \dot{\hat{x}} & = & \hat{A}\hat{x} + \hat{B}u, & \hat{A} \in \mathbb{R}^{r\times r},\ \hat{B} \in \mathbb{R}^{r\times m}, \\ \hat{y} & = & \hat{C}\hat{x} + \hat{D}u, & \hat{C} \in \mathbb{R}^{p\times r},\ \hat{D} \in \mathbb{R}^{p\times m} \end{array}$$

of order $r \ll n$, such that

$$\begin{aligned} \|y - \hat{y}\| \simeq \left\|Y - \hat{Y}\right\| = \left\|GU - \hat{G}U\right\| & \\ \leq \left\|G - \hat{G}\right\| \cdot \|U\| \simeq \left\|G - \hat{G}\right\| \cdot \|u\| & \\ \leq \text{tolerance} \cdot \|u\| . & \end{aligned}$$

Outline

Balanced Truncation for Linear Systems

Basic concept

- System $\Sigma : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t), \end{cases}$ with A stable, i.e., $\Lambda(A) \subset \mathbb{C}^-$,

  is balanced, if system Gramians, i.e., solutions P, Q of the Lyapunov equations

  $$AP + PA^T + BB^T = 0, \qquad A^TQ + QA + C^TC = 0,$$

  satisfy: $P = Q = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$ with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$.

Basic concept

- System $\Sigma:\ \begin{cases}\dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t),\end{cases}$ with A stable, i.e., $\Lambda(A) \subset \mathbb{C}^-$,

  is balanced, if system Gramians, i.e., solutions P, Q of the Lyapunov equations

  $$AP + PA^T + BB^T = 0, \qquad A^TQ + QA + C^TC = 0,$$

  satisfy: $P = Q = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$ with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$.
- $\{\sigma_1, \ldots, \sigma_n\}$ are the Hankel singular values (HSVs) of Σ.

Basic concept

- System $\Sigma : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t), \end{cases}$ with A stable, i.e., $\Lambda(A) \subset \mathbb{C}^-$,

  is balanced, if system Gramians, i.e., solutions P, Q of the Lyapunov equations

  $$AP + PA^T + BB^T = 0, \qquad A^TQ + QA + C^TC = 0,$$

  satisfy: $P = Q = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$ with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$.
- $\{\sigma_1, \ldots, \sigma_n\}$ are the Hankel singular values (HSVs) of Σ.
- Compute balanced realization (needs P, Q!) of the system via state-space transformation

$$\begin{aligned} \mathcal{T} : (A, B, C) &\mapsto (TAT^{-1}, TB, CT^{-1}) \\ &= \left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \begin{bmatrix} C_1 & C_2 \end{bmatrix} \right). \end{aligned}$$

Balanced Truncation for Linear Systems

Basic concept

- System $\Sigma : \begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t), \end{cases}$ with A stable, i.e., $\Lambda(A) \subset \mathbb{C}^-$,

  is balanced, if system Gramians, i.e., solutions P, Q of the Lyapunov equations

  $$AP + PA^T + BB^T = 0, \qquad A^TQ + QA + C^TC = 0,$$

  satisfy: $P = Q = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$ with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$.
- $\{\sigma_1, \ldots, \sigma_n\}$ are the Hankel singular values (HSVs) of Σ.
- Compute balanced realization (needs P, Q!) of the system via state-space transformation

  $$\begin{aligned} \mathcal{T} : (A, B, C) &\mapsto (TAT^{-1}, TB, CT^{-1}) \\ &= \left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \begin{bmatrix} C_1 & C_2 \end{bmatrix} \right). \end{aligned}$$

- Truncation $\rightsquigarrow (\hat{A}, \hat{B}, \hat{C}) = (A_{11}, B_1, C_1)$.
  Note: in efficient algorithms, truncation is achieved via projection:

  $$(\hat{A}, \hat{B}, \hat{C}) = (W^TAV, W^TB, CV), \quad \text{where} \quad W^TV = I_r.$$

Properties

- Reduced-order model is stable with HSVs $\sigma_1, \ldots, \sigma_r$.

Properties

- Reduced-order model is stable with HSVs $\sigma_1, \ldots, \sigma_r$.
- Adaptive choice of r via computable error bound:

$$\|y - \hat{y}\|_2 \leq \|G - \hat{G}\|_{\mathcal{H}_\infty} \|u\|_2 \leq \left(2 \sum\nolimits_{k=r+1}^{n} \sigma_k\right) \|u\|_2 \,,$$

where $\|G\|_{\mathcal{H}_\infty} := \sup_{u \in \mathcal{L}_2 \setminus \{0\}} \frac{\|Gu\|_2}{\|u\|_2} = \sup_{\omega \in \mathbb{R}} \sigma_{\max}(G(\jmath\omega))$.

Properties

- Reduced-order model is stable with HSVs $\sigma_1, \ldots, \sigma_r$.
- Adaptive choice of r via computable error bound:

$$\|y - \hat{y}\|_2 \leq \|G - \hat{G}\|_{\mathcal{H}_\infty} \|u\|_2 \leq \left(2 \sum\nolimits_{k=r+1}^{n} \sigma_k\right) \|u\|_2 \,,$$

where $\|G\|_{\mathcal{H}_\infty} := \sup_{u \in \mathcal{L}_2 \setminus \{0\}} \frac{\|Gu\|_2}{\|u\|_2} = \sup_{\omega \in \mathbb{R}} \sigma_{\max}(G(\jmath\omega))$.

Practical implementation

- Rather than solving Lyapunov equations for P, Q (n^2 unknowns!), find $S, R \in \mathbb{R}^{n \times s}$ with $s \ll n$ such that $P \approx SS^T$, $Q \approx RR^T$.
- Reduced-order model directly obtained via small-scale ($s \times s$) SVD of $R^T S$!

[a] http://www.mpi-magdeburg.mpg.de/projects/morlab
[b] https://www.mpi-magdeburg.mpg.de/projects/mess, full MATLAB integration in progress.

Balanced Truncation for Linear Systems

Properties

- Reduced-order model is stable with HSVs $\sigma_1, \ldots, \sigma_r$.
- Adaptive choice of r via computable error bound:

$$\|y - \hat{y}\|_2 \leq \|G - \hat{G}\|_{\mathcal{H}_\infty} \|u\|_2 \leq \left(2 \sum\nolimits_{k=r+1}^{n} \sigma_k\right) \|u\|_2 \,,$$

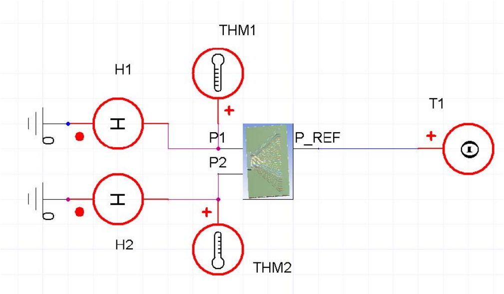
where $\|G\|_{\mathcal{H}_\infty} := \sup_{u \in \mathcal{L}_2 \setminus \{0\}} \frac{\|Gu\|_2}{\|u\|_2} = \sup_{\omega \in \mathbb{R}} \sigma_{\max}(G(\jmath\omega))$.

Practical implementation

- Rather than solving Lyapunov equations for P, Q (n^2 unknowns!), find $S, R \in \mathbb{R}^{n \times s}$ with $s \ll n$ such that $P \approx SS^T$, $Q \approx RR^T$.
- Reduced-order model directly obtained via small-scale $(s \times s)$ SVD of $R^T S$!
- Two software packages:
  - MORLAB[a] (**M**odel **O**rder **R**eduction **LAB**oratory), based on spectral projection methods ($\rightsquigarrow$ small to medium size problems, up to $n \sim 5,000$.)
  - M-M.E.S.S.[b] provides solvers for large-scale matrix equations with sparse/low-rank coefficients and basic MOR functionality; no $\mathcal{O}(n^3)$ or $\mathcal{O}(n^2)$ computations necessary!

[a] http://www.mpi-magdeburg.mpg.de/projects/morlab
[b] https://www.mpi-magdeburg.mpg.de/projects/mess, full MATLAB integration in progress.

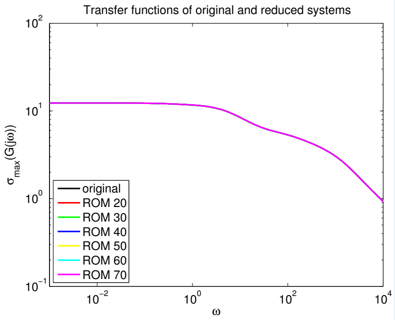
- SIMPLORER® test circuit with 2 transistors.



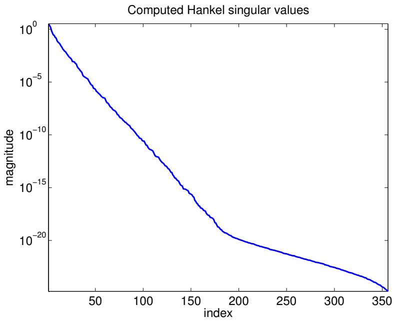
- Conservative thermic sub-system in SIMPLORER:
  voltage ⇝ temperature, current ⇝ heat flow.
- Original model: $n = 270.593$, $m = q = 2 \Rightarrow$
  Computing times (on Intel Xeon dualcore 3GHz, 1 Thread):
  - Main computational cost for set-up data $\approx 22min$.
  - Computation of reduced models from set-up data: 44–49sec. ($r = 20$–70).
  - Bode plot (MATLAB on Intel Core i7, 2,67GHz, 12GB):
    7.5h for original system , $<$ 1min for reduced system.

- Original model: $n = 270.593$, $m = q = 2 \Rightarrow$
  Computing times (on Intel Xeon dualcore 3GHz, 1 Thread):
  - Main computational cost for set-up data $\approx 22min$.
  - Computation of reduced models from set-up data: 44–49sec. ($r = 20$–70).
  - Bode plot (MATLAB on Intel Core i7, 2,67GHz, 12GB):
    7.5h for original system , $<$ 1min for reduced system.

Bode Plot (Amplitude)

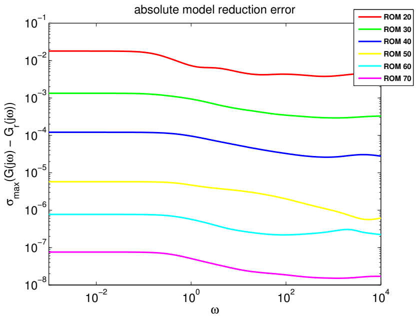


Hankel Singular Values

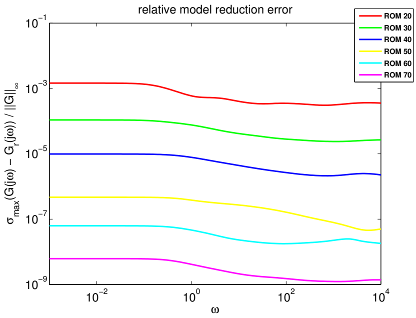


- Original model: $n = 270.593$, $m = q = 2 \Rightarrow$
  Computing times (on Intel Xeon dualcore 3GHz, 1 Thread):
  - Main computational cost for set-up data $\approx 22min$.
  - Computation of reduced models from set-up data: 44–49sec. ($r = 20$–70).
  - Bode plot (MATLAB on Intel Core i7, 2,67GHz, 12GB):
    7.5h for original system , $<$ 1min for reduced system.

Absolute Error

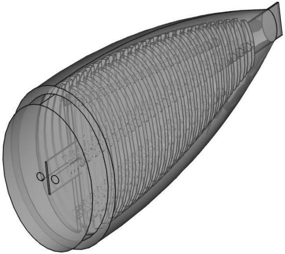


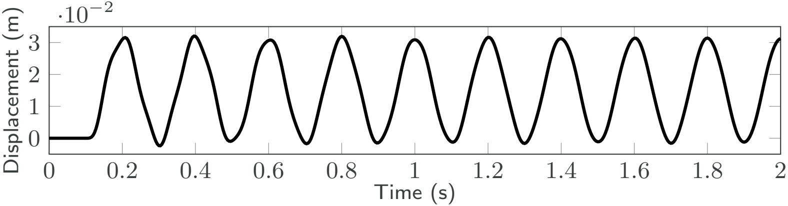
Relative Error

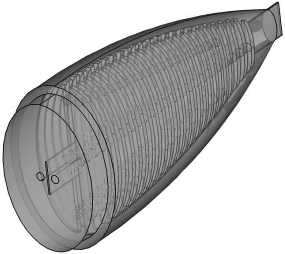


Original system:

- mechanical second-order structure
- $n = 779{,}232$, $m = 1$, $p = 3$
- test simulation time $\approx 4\,\text{h}$





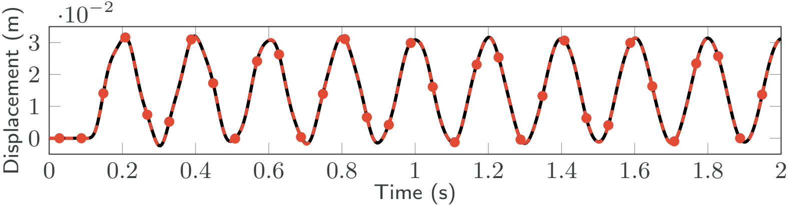


Original system:

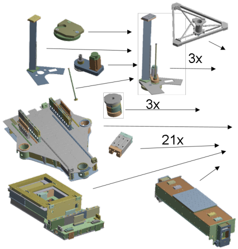
- mechanical second-order structure
- $n = 779,232$, $m = 1$, $p = 3$
- test simulation time $\approx 4\,\mathrm{h}$

Second-Order Balanced Truncation:

- two-step method with MORLAB backend
- $r = 1$, wall time $\approx 21\,\mathrm{h}$
- test simulation time $\approx 10\,\mathrm{msec}$ (speed up $\approx 1.4 \cdot 10^6$)



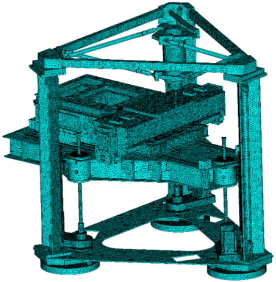
50 subassemblies



CAD model

$\overset{FEM}{\rightsquigarrow}$

FE-Model: **1.2M DOFs**



Industrial challenges for virtual twin:

- **non-homogeneous initial conditions** (IC) — two approaches: augment input with IC ("BTX0") or use superposition ("2phase"),
- subsystem reduction ("output coupled") vs. holistic reduction ("FE-coupled").

Technische Universität Dresden · RWTH Aachen University · Technische Universität Chemnitz · Sonderforschungsbereich Transregio 96

Numerical Examples

MAX: Results considering an inhomogeneous initial condition $x_0 \neq 0$

FE-coupled

method	red. order tol 10^{-3}	t_{red}
2phase	196	6.5h
BTX0	174	4.5h

output-coupled

method	red. order tol 10^{-3}	t_{red}
2phase	3,005	2h
BTX0	2,515	1.8h

Numerical Examples

MAX: Results considering an inhomogeneous initial condition $x_0 \neq 0$

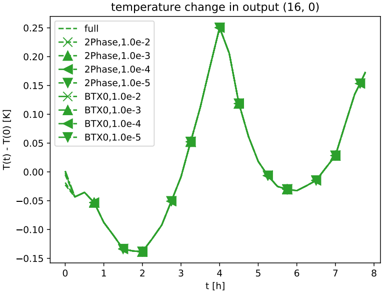
FE-coupled

method	red. order tol 10^{-3}	t_{red}
2phase	196	6.5h
BTX0	174	4.5h

output-coupled

method	red. order tol 10^{-3}	t_{red}
2phase	3,005	2h
BTX0	2,515	1.8h

$\rightarrow$ Required storage for reduced matrices just 1MB!



Vettermann, J., Sauerzapf, S., Naumann, A., Beitelschmidt, M., Herzog, R., Benner, P., Saak, J. (2021): Model order reduction methods for coupled machine tool models. MM Science Journal 2021:4652-4659.

Vortex shedding

- Often encountered in industrial applications: system of the form

$$E\dot{x}(t) = Ax(t) + Bu(t)$$

with singular E $\rightsquigarrow$ linear differential-algebraic (descriptor) system

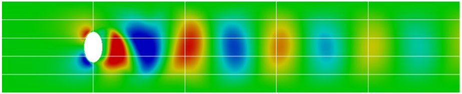
- Here: linearized Navier-Stokes equations for control design: index $= 2$, $n = 22,385$, $m = 2$, $p = 7$.

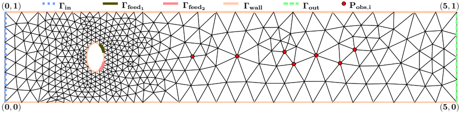
Balanced truncation approximation

- $r = 153$ for tolerance $\tau = 10^{-5}$
- hardware: Intel® Core™ i7-6700 with 16GB RAM
- software: M-M.E.S.S.-2.0.1 in MATLAB® R2018a
- wall time: $\approx$ 2min.

M.E.S.S. demo example:

```
bt_mor_DAE2('NSE',3,500)
```





Model by Heiko K. Weichelt

Vortex shedding

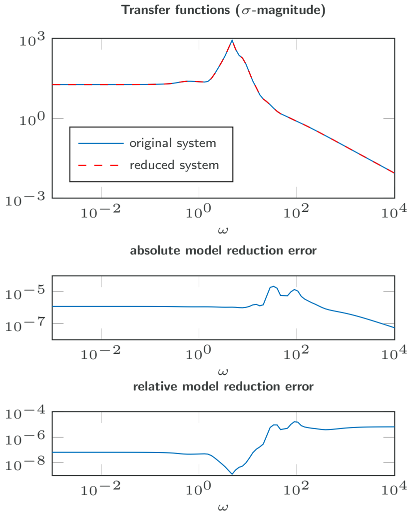
- Often encountered in industrial applications: system of the form

$$E\dot{x}(t) = Ax(t) + Bu(t)$$

with singular E $\rightsquigarrow$ linear differential-algebraic (descriptor) system
- Here: linearized Navier-Stokes equations for control design: index $= 2$, $n = 22,385$, $m = 2$, $p = 7$.

Balanced truncation approximation

- $r = 153$ for tolerance $\tau = 10^{-5}$
- hardware: Intel® Core™ i7-6700 with 16GB RAM
- software: M-M.E.S.S.-2.0.1 in MATLAB R2018a
- wall time:$\approx$ 2min.

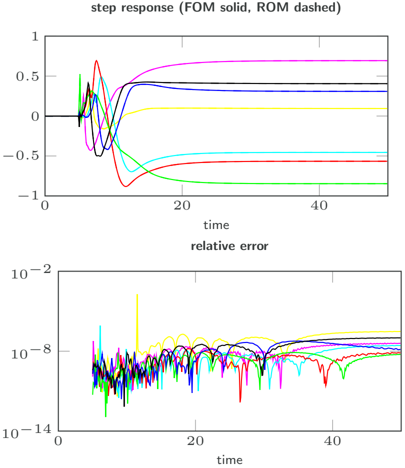


Vortex shedding

- Often encountered in industrial applications: system of the form
$$E\dot{x}(t) = Ax(t) + Bu(t)$$
with singular E $\rightsquigarrow$ linear differential-algebraic (descriptor) system
- Here: linearized Navier-Stokes equations for control design: index $= 2$, $n = 22,385$, $m = 2$, $p = 7$.

Balanced truncation approximation

- $r = 153$ for tolerance $\tau = 10^{-5}$
- hardware: Intel® Core™ i7-6700 with 16GB RAM
- software: M-M.E.S.S.-2.0.1 in MATLAB R2018a
- wall time:$\approx$ 2min.



- Nonlinear balancing based on energy functionals [SCHERPEN 1993, GRAY/MESKO 1996].

Definition [SCHERPEN 1993, GRAY/MESKO 1996]

The reachability energy functional $L_c(x_0)$, and observability energy functional $L_o(x_0)$ of a system are given as:

$$L_c(x_0) = \inf_{\substack{u \in L_2(-\infty,0] \\ x(-\infty)=0,\ x(0)=x_0}} \frac{1}{2}\int_{-\infty}^{0} \|u(t)\|^2 dt, \qquad L_o(x_0) = \frac{1}{2}\int_{0}^{\infty} \|y(t)\|^2 dt.$$

Disadvantage: energy functionals are the solutions of nonlinear Hamilton-Jacobi equations which are hardly solvable for large-scale systems.

Note: For linear (LTI) systems,

$$L_c(x_0) = \frac{1}{2} x_0^T P^{-1} x_0, \qquad L_o(x_0) = \frac{1}{2} x_0^T Q x_0,$$

where P, Q are the controllability and observability Gramians, respectively!

- Nonlinear balancing based on energy functionals [Scherpen 1993, Gray/Mesko 1996].

  Disadvantage: energy functionals are the solutions of nonlinear Hamilton-Jacobi equations which are hardly solvable for large-scale systems.

- Empirical Gramians/frequency-domain POD [Lall et al 1999, Willcox/Peraire 2002].

Example: controllability Gramian from time domain data (snapshots)

1. Define reachability Gramian of the system
$$P = \int_0^\infty x(t)x(t)^T dt, \quad \text{where } x(t) \text{ solves } \dot{x} = f(x, \delta),\ x(0) = x_0.$$
2. Use time-domain integrator to produce snapshots $x_k \approx x(t_k)$, $k = 1, \ldots, K$.
3. Approximate $P \approx \sum_{k=0}^K w_k x_k x_k^T$ with positive weights w_k.
4. Analogously for observability Gramian.
5. Compute balancing transformation and apply it to nonlinear system.

Disadvantage: Depends on chosen training input (e.g., $\delta(t_0)$) like other POD approaches.

For recent developments on empirical Gramians, see [Himpe 2018].

- Nonlinear balancing based on energy functionals [Scherpen 1993, Gray/Mesko 1996].

  Disadvantage: energy functionals are the solutions of nonlinear Hamilton-Jacobi equations which are hardly solvable for large-scale systems.

- Empirical Gramians/frequency-domain POD [Lall et al 1999, Willcox/Peraire 2002].

  Disadvantage: Depends on chosen training input (e.g., $\delta(t_0)$) like other POD approaches.

  For recent developments on empirical Gramians, see [Himpe 2018].

- $\rightsquigarrow$ **Goal:** computationally efficient and input-independent method!

W. S. Gray and J. P. Mesko. Controllability and observability functions for model reduction of nonlinear systems. In Proc. of the Conf. on Information Sci. and Sys., pp. 1244–1249, 1996.

C. Himpe. emgr — The empirical Gramian framework. Algorithms 11(7): 91, 2018. doi:10.3390/a11070091.

S. Lall, J. Marsden, and S. Glavaški. A subspace approach to balanced truncation for model reduction of nonlinear control systems. International Journal of Robust and Nonlinear Control, 12:519–535, 2002.

J. M. A. Scherpen. Balancing for nonlinear systems. Systems & Control Letters, 21:143–153, 1993.

K. Willcox and J. Peraire, Balanced model reduction via the proper orthogonal decomposition. AIAA Journal, 40:2323–2330, 2002.

- A possible solution is to obtain bounds for the energy functionals, instead of computing them exactly.

- A possible solution is to obtain bounds for the energy functionals, instead of computing them exactly.

- For bilinear systems, such local bounds were derived in [B./Damm 2011] using the solutions to the Lyapunov-plus-positive equations:

$$AP + PA^T + \sum_{i=1}^{m} A_i P A_i^T + BB^T = 0,$$
$$A^T Q + QA^T + \sum_{i=1}^{m} A_i^T Q A_i + C^T C = 0.$$

(Note: if their solutions exist, they define reachability and observability Gramians of BIBO stable bilinear system.)

- **Here, we aim at determining algebraic Gramians for polynomial systems, which**
  - **provide bounds for the energy functionals of polynomial systems,**
  - **generalize the Gramians of linear and bilinear systems, and**
  - **allow us to find the states that are hard to reach as well as hard to observe in an efficient and reliable way.**

Now, consider the class of **polynomial control (PC) Systems**:

$$\boxed{\begin{aligned}\dot{x}(t) &= Ax(t) + \sum_{j=2}^{n_p} H_j\left(\otimes^j x(t)\right) + \sum_{j=2}^{n_p}\sum_{k=1}^{m} N_j^k\left(\otimes^j x(t)\right)u_k(t) + Bu(t),\\ y(t) &= Cx(t), \quad x(0) = 0,\end{aligned}}$$

where

- n_p is the degree of the polynomial part of the system,
- $x(t) \in \mathbb{R}^n$, $\otimes^j x(t) = \underbrace{x(t) \otimes \cdots \otimes x(t)}_{j\text{-times}}$,
- $u(t) \in \mathbb{R}^m$, and $y(t) \in \mathbb{R}^p$, $n \gg m, p$.
- $A \in \mathbb{R}^{n\times n}$, $H_j, N_j^k \in \mathbb{R}^{n\times n^j}$, $B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{p\times n}$.
- Assumption: A is supposed to be Hurwitz $\Rightarrow$ local stability.

Now, consider the class of **polynomial control (PC) Systems**:

$$\boxed{\begin{aligned}\dot{x}(t) &= Ax(t) + \sum_{j=2}^{n_p} H_j \left(\otimes^j x(t)\right) + \sum_{j=2}^{n_p}\sum_{k=1}^{m} N_j^k \left(\otimes^j x(t)\right) u_k(t) + Bu(t),\\ y(t) &= Cx(t), \quad x(0) = 0,\end{aligned}}$$

where

- n_p is the degree of the polynomial part of the system,
- $x(t) \in \mathbb{R}^n$, $\otimes^j x(t) = \underbrace{x(t) \otimes \cdots \otimes x(t)}_{j\text{-times}}$,
- $u(t) \in \mathbb{R}^m$, and $y(t) \in \mathbb{R}^p$, $n \gg m, p$.
- $A \in \mathbb{R}^{n \times n}$, $H_j, N_j^k \in \mathbb{R}^{n \times n^j}$, $B \in \mathbb{R}^{n \times m}$ and $C \in \mathbb{R}^{p \times n}$.
- Assumption: A is supposed to be Hurwitz $\Rightarrow$ local stability.

Examples: FitzHugh-Nagumo and Chafee-Infante equations lead to cubic control systems; cubic-quintic Allen-Cahn equation to quintic control system.

- Consider input $\rightarrow$ state map of (for simplicity) quadratic-bilinear system ($n_P = 2$, $m = 1,\ N \equiv A_1$):

$$\dot{x}(t) = Ax(t) + Hx(t) \otimes x(t) + Nx(t)u(t) + Bu(t), \qquad x(0) = 0.$$

- Integration yields

$$x(t) = \int_0^t e^{A\sigma_1} Bu(t-\sigma_1)d\sigma_1 + \int_0^t e^{A\sigma_1} Nx(t-\sigma_1)u(t-\sigma_1)d\sigma_1 + \int_0^t e^{A\sigma_1} Hx(t-\sigma_1) \otimes x(t-\sigma_1)d\sigma_1$$

[Rugh 1981]

- Consider input $\rightarrow$ state map of (for simplicity) quadratic-bilinear system ($n_P = 2$, $m = 1,\ N \equiv A_1$):

$$\dot{x}(t) = Ax(t) + Hx(t) \otimes x(t) + Nx(t)u(t) + Bu(t), \qquad x(0) = 0.$$

- Integration yields

$$\begin{aligned} x(t) &= \int_0^t e^{A\sigma_1} Bu(t-\sigma_1)d\sigma_1 + \int_0^t e^{A\sigma_1} Nx(t-\sigma_1)u(t-\sigma_1)d\sigma_1 \\ &\qquad + \int_0^t e^{A\sigma_1} Hx(t-\sigma_1) \otimes x(t-\sigma_1)d\sigma_1 \\ &= \int_0^t e^{A\sigma_1} Bu(t-\sigma_1)d\sigma_1 + \int_0^t \int_0^{t-\sigma_1} e^{A\sigma_1} N e^{A\sigma_2} Bu(t-\sigma_1)u(t-\sigma_1-\sigma_2)d\sigma_1 d\sigma_2 \\ &\quad + \int_0^t \int_0^{t-\sigma_1} \int_0^{t-\sigma_1} e^{A\sigma_1} H(e^{A\sigma_2}B \otimes e^{A\sigma_3}B)u(t-\sigma_1-\sigma_2)u(t-\sigma_1-\sigma_3)d\sigma_1 d\sigma_2 d\sigma_3 + \ldots \end{aligned}$$

[Rugh 1981]

- Consider input $\rightarrow$ state map of (for simplicity) quadratic-bilinear system ($n_P = 2$, $m = 1,\ N \equiv A_1$):

$$\dot{x}(t) = Ax(t) + Hx(t) \otimes x(t) + Nx(t)u(t) + Bu(t), \qquad x(0) = 0.$$

- Integration yields

$$\begin{aligned}
x(t) &= \int_0^t e^{A\sigma_1} Bu(t-\sigma_1)d\sigma_1 + \int_0^t e^{A\sigma_1} Nx(t-\sigma_1)u(t-\sigma_1)d\sigma_1 \\
&\qquad + \int_0^t e^{A\sigma_1} Hx(t-\sigma_1) \otimes x(t-\sigma_1)d\sigma_1 \\
&= \int_0^t e^{A\sigma_1} Bu(t-\sigma_1)d\sigma_1 + \int_0^t \int_0^{t-\sigma_1} e^{A\sigma_1} N e^{A\sigma_2} Bu(t-\sigma_1)u(t-\sigma_1-\sigma_2)d\sigma_1 d\sigma_2 \\
&\quad + \int_0^t \int_0^{t-\sigma_1} \int_0^{t-\sigma_1} e^{A\sigma_1} H(e^{A\sigma_2} B \otimes e^{A\sigma_3} B)u(t-\sigma_1-\sigma_2)u(t-\sigma_1-\sigma_3)d\sigma_1 d\sigma_2 d\sigma_3 + \ldots
\end{aligned}$$

- By iteratively inserting expressions for $x(t - \bullet)$, we obtain the **Volterra series expansion** for quadratic-bilinear (and, more general, polynomial) systems. [Rugh 1981]

Expanding the response of the PC system into a Volterra series representation and using the idea of iterated linear systems, we define the reachability Gramian as

$$\boxed{P = \sum_{k=1}^{\infty} \int_0^{\infty} \cdots \int_0^{\infty} \bar{P}_k(t_1, \ldots, t_k) \bar{P}_k(t_1, \ldots, t_k)^T dt_1 \ldots dt_k,}$$

where

$$\bar{P}_1(t_1) = e^{At_1} B, \qquad \bar{P}_2(t_1, t_2) = \sum_{k=1}^{m} e^{At_1} N_1^k e^{At_2} B,$$

$$\bar{P}_3(t_1, t_2, t_3) = e^{At_1} H_2 e^{At_2} B \otimes e^{At_3} B, \quad \ldots$$

are the kernels of the Volterra series expansion of the system output.

Expanding the response of the PC system into a Volterra series representation and using the idea of iterated linear systems, we define the reachability Gramian as

$$\boxed{P = \sum_{k=1}^{\infty} \int_0^{\infty} \cdots \int_0^{\infty} \bar{P}_k(t_1, \ldots, t_k) \bar{P}_k(t_1, \ldots, t_k)^T dt_1 \ldots dt_k,}$$

where

$$\bar{P}_1(t_1) = e^{At_1} B, \qquad \bar{P}_2(t_1, t_2) = \sum_{k=1}^{m} e^{At_1} N_1^k e^{At_2} B,$$
$$\bar{P}_3(t_1, t_2, t_3) = e^{At_1} H_2 e^{At_2} B \otimes e^{At_3} B, \quad \ldots$$

are the kernels of the Volterra series expansion of the system output.

Theorem [B./Goyal/Pontes Duff 2018]

The reachability Gramian P of a PC system solves the polynomial Lyapunov equation

$$AP + PA^T + BB^T + \sum_{j=2}^{n_p} H_j \left(\otimes^j P\right) H_j^T + \sum_{j=2}^{n_p} \sum_{k=1}^{m} N_j^k \left(\otimes^j P\right) \left(N_j^k\right)^T = 0.$$

The Observability Gramian is defined as follows:

- First, we write the adjoint system as [Fujimoto et al. 2002]

$$
\boxed{
\begin{aligned}
\dot{x}(t) &= Ax(t) + \sum_{j=2}^{n_p} H_j x_j^{\otimes}(t) + \sum_{j=1}^{n_p}\sum_{k=1}^{m} N_j^k x_j^{\otimes}(t) u_k(t) + Bu(t),\\
\dot{x}_d(t) &= -A^T x_d(t) - \sum_{j=2}^{n_p} H_j^{(2)} x_{d,j}^{\otimes}(t) - \sum_{j=1}^{n_p}\sum_{k=1}^{m} \left(N_j^{k,(2)}\right) x_{d,j}^{\otimes}(t) u_{d,k}(t) - C^T u_d(t), \quad x_d(\infty) = 0,\\
y_d(t) &= B^T x_d(t).
\end{aligned}
}
$$

The Observability Gramian is defined as follows:

- First, we write the adjoint system as [Fujimoto et al. 2002]

$$\boxed{\begin{aligned}
\dot{x}(t) &= Ax(t) + \sum_{j=2}^{n_p} H_j x_j^{\otimes}(t) + \sum_{j=1}^{n_p}\sum_{k=1}^{m} N_j^k x_j^{\otimes}(t) u_k(t) + Bu(t),\\
\dot{x}_d(t) &= -A^T x_d(t) - \sum_{j=2}^{n_p} H_j^{(2)} x_{d,j}^{\otimes}(t) - \sum_{j=1}^{n_p}\sum_{k=1}^{m} \left(N_j^{k,(2)}\right) x_{d,j}^{\otimes}(t) u_{d,k}(t) - C^T u_d(t), \quad x_d(\infty) = 0,\\
y_d(t) &= B^T x_d(t).
\end{aligned}}$$

- Then, by taking the kernel of Volterra series, one has

Theorem [B./Goyal/Pontes Duff 2018]

Let P be the reachability Gramian. Then, the observability Gramian Q of the PC system solves the polynomial Lyapunov equation

$$A^T Q + QA + C^T C + \sum_{j=2}^{n_p} H_j^{(2)} \left(\otimes^{j-1} P \otimes Q\right) \left(H_j^{(2)}\right)^T + \sum_{j=2}^{n_p}\sum_{k=1}^{m} N_j^{k,(2)} \left(\otimes^{j-1} P \otimes Q\right) \left(N_j^{k,(2)}\right)^T = 0.$$

Truncated Gramians

- Polynomial Lyapunov equations are very expensive to solve, efficient algorithms have not yet been developed.

- We thus propose truncated Gramians that only involve a finite number of kernels and can be computed using the methods in MORLAB or M-M.E.S.S.:

$$\boxed{P_{\mathcal{T}} = \sum_{k=1}^{n_p+1} \int_0^\infty \cdots \int_0^\infty \bar{P}_k(t_1,\ldots,t_k)\bar{P}_k(t_1,\ldots,t_k)^T dt_1 \ldots dt_k.}$$

Truncated Gramians

The truncated reachability Gramian solves

$$AP_{\mathcal{T}} + P_{\mathcal{T}}A^T + BB^T + \sum_{j=2}^{n_p} H_j \otimes^j P_l H_j^T + \sum_{j=2}^{n_p}\sum_{k=1}^{m} N_j^k \otimes^j P_l \left(N_j^k\right)^T = 0.$$

where $AP_l + P_lA^T + BB^T = 0$

- Advantage: Only need to solve a finite number of (linear) Lyapunov equations.

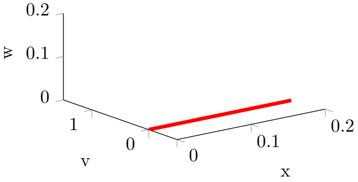
$$\epsilon v_t(x,t) = \epsilon^2 v_{xx}(x,t) + f(v(x,t)) - w(x,t) + q,$$
$$w_t(x,t) = hv(x,t) - \gamma w(x,t) + q,$$

with a nonlinear function

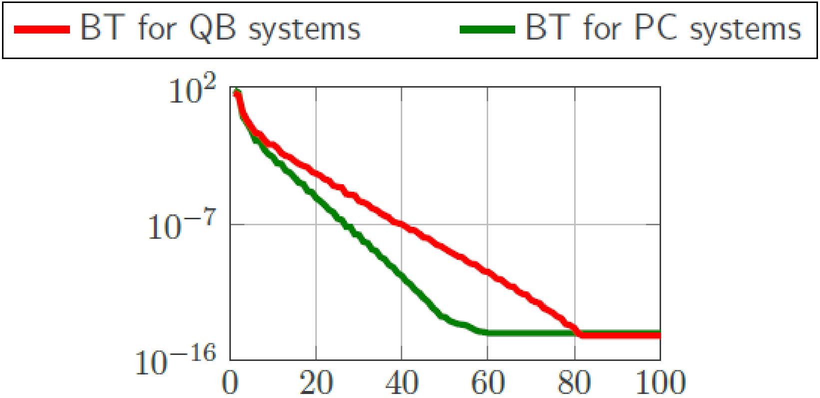
$$f(v(x,t)) = v(v-0.1)(1-v),$$

$\epsilon = 0.015,\ h = 0.5,\ \gamma = 2,\ q = 0.05,\ L = 0.2,$
and boundary conditions:

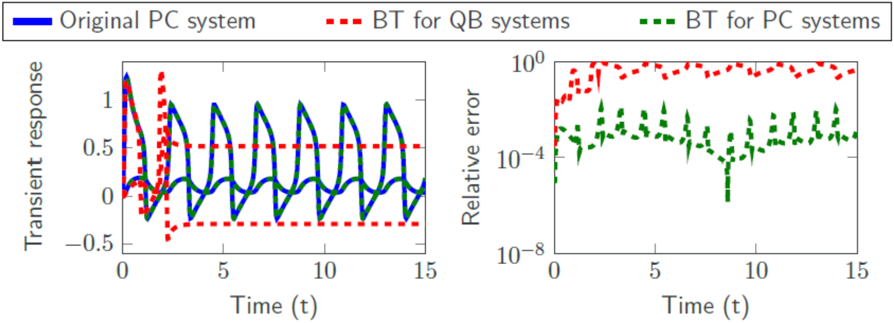
$$v_x(0,t) = i_0(t), \quad v_x(L,t) = 0, \quad t \geq 0,$$



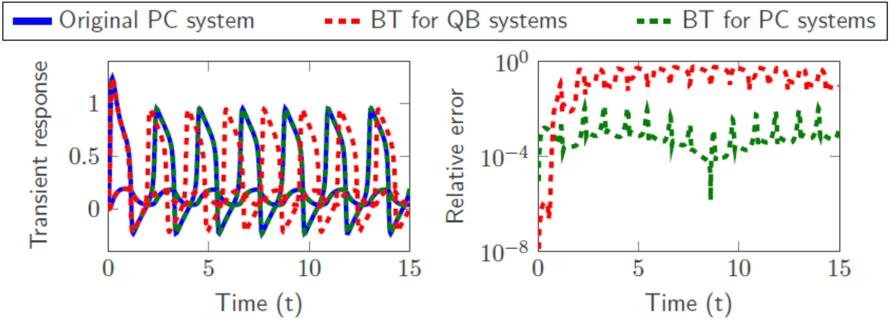
- Spatial discretization leads to PC system with cubic nonlinearity of order $n_{pc} = 600$.
- Lifting: $z := v^2 \ \Rightarrow\ f(v,z) = -vz + 1.1z - 0.1v, \quad z_t = 2vv_t = \ldots \ \rightsquigarrow$
- lifted quadratic-bilinear (QB) system of order $n_{qb} = 900$.
- Outputs of interest $v(0,t), w(0,t)$ are the responses at the left boundary.
- We compare balanced truncation for PC and QB systems.



- Decay of singular values for PC systems is faster $\Rightarrow$ smaller reduced-order model!



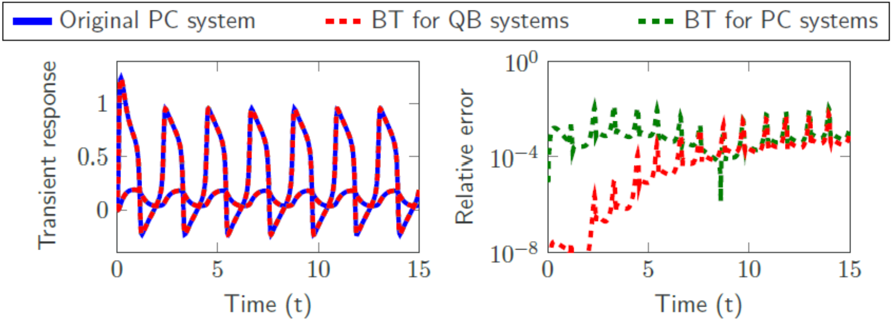
- Original PC system of order 600. Original QB system of order 900.
- Reduced PC system of order 10. Reduced QB system of order 10.



- Original PC system of order 600. Original QB system of order 900.
- Reduced PC system of order 10. Reduced QB system of order 30.

Numerical Example

The FitzHugh-Nagumo Model: Time-domain Simulation



- Original PC system of order 600. Original QB system of order 900.
- Reduced PC system of order 10. Reduced QB system of order 43.

Conclusions

BT for linear systems:

- Method of choice for model order reduction in optimal and feedback control.
- Computational efficiency enhanced through advanced techniques from numerical linear algebra so that problems of industrial scale can be reduced.
- Numerous technical details necessary for use in digital and virtual twins.
- Robust and efficient implementations available in numerous software packages like SLICOT, M.E.S.S., MORLAB, pyMOR, ...

Conclusions

BT for linear systems:

- Method of choice for model order reduction in optimal and feedback control.
- Computational efficiency enhanced through advanced techniques from numerical linear algebra so that problems of industrial scale can be reduced.
- Numerous technical details necessary for use in digital and virtual twins.
- Robust and efficient implementations available in numerous software packages like SLICOT, M.E.S.S., MORLAB, pyMOR, . . .

BT for nonlinear systems:

- BT extended to bilinear, quadratic-bilinear, and polynomial systems.
- Not in this talk: local Lyapunov stability is preserved.
- As of yet, only weak motivation by local bounds of energy functionals.
- No error bounds in terms of "Hankel" singular values.
- Computationally efficient (as compared to nonlinear balancing), and input independent.
- **To do:**
  - improve efficiency of Lyapunov solvers with many right-hand sides further;
  - error bound;
  - conditions for existence of new PC Gramians;
  - extension to descriptor systems.

Own References (Linear BT and Software)

P. Benner, E. S. Quintana-Ortí, and G. Quintana-Ortí.
Balanced truncation model reduction of large-scale dense systems on parallel computers.
Mathematical and Computer Modelling of Dynamical Systems, 6:383–405, 2000.

P. Benner.
Solving large-scale control problems.
IEEE Control Systems Magazine, 24:44–59, 2004.

P. Benner.
Numerical linear algebra for model reduction in control and simulation.
GAMM-Mitteilungen, 29:275–296, 2006.

P. Benner and J. Saak.
Efficient balancing-based MOR for large-scale second-order systems.
Mathematical and Computer Modelling of Dynamical Systems, 17:123–143, 2011.

C. Nowakowski, P. Kürschner, P. Eberhard and P. Benner.
Model reduction of an elastic crankshaft for elastic multibody simulations.
Zeitschrift für Angewandte Mathematik und Mechanik, 93:198–216, 2013.

P. Benner, P. Kürschner, and J. Saak.
An improved numerical method for balanced truncation for symmetric second order systems.
Mathematical and Computer Modelling of Dynamical Systems, 19:593–615, 2013.

P. Benner, J. Saak, and M. M. Uddin.
Balancing based model reduction for structured index-2 unstable descriptor systems with application to flow control.
Numerical Algebra, Control and Optimization, 6:1–20, 2016.

P. Benner, J. Saak, and M. M. Uddin.
Structure preserving model order reduction of large sparse second-order index-1 systems and application to a mechatronics model.
Mathematical and Computer Modelling of Dynamical Systems, 22:509–523, 2016.

P. Benner and T. Stykel.
Model order reduction for differential-algebraic equations: A survey.
In Surveys in Differential-Algebraic Equations IV, A. Ilchmann and T. Reis, eds., Differential-Algebraic Equations Forum, Springer International Publishing, Cham, pp. 107–160, 2017.

J. Saak, D. Siebelts, and S. W. R. Werner.
A comparison of second-order model order reduction methods for an artificial fishtail.
at-Automatisierungstechnik, 67:648–667, 2019.

P. Benner and S. W. R. Werner.
MORLAB – A model order reduction framework in MATLAB and Octave.
In Mathematical Software – ICMS 2020, A. M. Bigatti, J. Carette, J. H. Davenport, M. Joswig, and T. de Wolff, eds., vol. 12097 of Lecture Notes in Comput. Sci., Springer International Publishing, Cham, pp. 432–441, 2020.

P. Benner, M. Köhler, and J. Saak.
Matrix equations, sparse solvers: M-M.E.S.S.-2.0.1 – philosophy, features and application for (parametric) model order reduction.
In Model Reduction of Complex Dynamical Systems, P. Benner, T. Breiten, H. Faßbender, M. Hinze, T. Stykel, and R. Zimmermann, eds., vol. 171 of International Series of Numerical Mathematics, Birkhäuser, Cham, pp. 369–392, 2021.

J. Vettermann, S. Sauerzapf, A. Naumann, M. Beitelschmidt, R. Herzog, P. Benner, and J. Saak.
Model order reduction methods for coupled machine tool models.
MM Science Journal, pp. 4652–4659, 2021.

Own References (Nonlinear BT)

P. Benner and T. Damm.
Lyapunov Equations, Energy Functionals, and Model Order Reduction of Bilinear and Stochastic Systems.
SIAM Journal on Control and Optimiztion, 49(2):686–711, 2011.

P. Benner and T. Breiten.
Low Rank Methods for a Class of Generalized Lyapunov Equations and Related Issues.
Numerische Mathematik, 124(3):441–470, 2013.

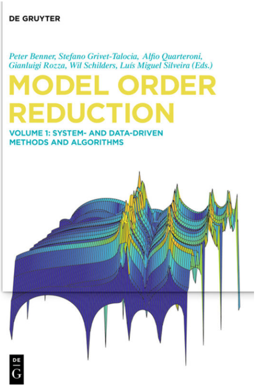
P. Benner and T. Breiten.
Two-Sided Projection Methods for Nonlinear Model Order Reduction.
SIAM Journal on Scientific Computing, 37(2):B239—B260, 2015.

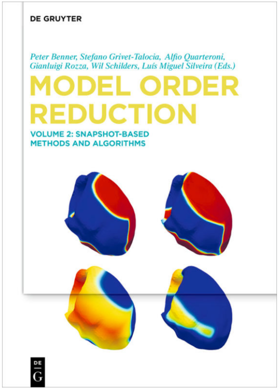
P. Benner, T. Damm, M. Redmann, and Y. Rocio Rodriguez Cruz.
Positive Operators and Stable Truncation.
Linear Algebra and its Applications, 498:74–87, 2016.

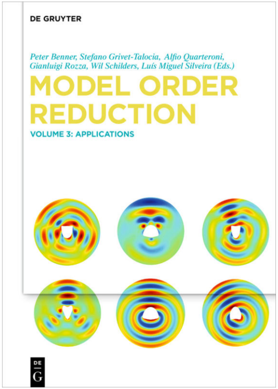
P. Benner, P. Goyal, and M. Redmann.
Truncated Gramians for Bilinear Systems and their Advantages in Model Order Reduction.
In P. Benner, M. Ohlberger, T. Patera, G. Rozza, K. Urban (Eds.), Model Reduction of Parametrized Systems, MS & A — Modeling, Simulation and Applications, Vol. 17, pp. 285–300. Springer International Publishing, Cham, 2017.

P. Benner and P. Goyal.
Balanced Truncation Model Order Reduction for Quadratic-Bilinear Control Systems.
arXiv Preprint arXiv:1705.00160, April 2017.

P. Benner, P. Goyal, and I. Pontes Duff.
Approximate Balanced Truncation for Polynomial Control Systems.
Coming soon(er or later).







- Edited by Peter Benner, Stefano Grivet-Talocia, Alfio Quarteroni, Gianluigi Rozza, Wil Schilders, and Luís Miguel Silveira,
- contains 30 tutorial chapters on modern model reduction techniques, methods, applications, and software,
- published by DeGruyter in 2021, ebook is fully Open Access!