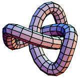
ADVANCES IN BALANCING-RELATED MODEL REDUCTION FOR CIRCUIT SIMULATION

Peter Benner

Mathematik in Industrie und Technik
Fakultät für Mathematik
Technische Universität Chemnitz



Scientific Computing in Electrical Engineering
Helsinki University of Technology
Espoo, Finland
September 28 – October 3, 2008

Support and Thanks

SyreNe
System reduction for IC design in nano-electronics

BMBF (Ministry of Education and Research, Germany) research network.

Partners:
TU Berlin, TU Braunschweig, TU Chemnitz,
U Hamburg. FhG-ITWM Kaiserslautern.
Infineon Technologies AG,
NEC Europe Ltd., Qimonda AG.

SyreNe
System Reduction for Nanoscale IC Design

Support and Thanks

O-MOORE-NICE!
Operational model order reduction for nanoscale IC electronics

EU support via Marie Curie Host Fellowships for the Transfer of Knowledge (ToK) Industry-Academia Partnership Scheme.

Partners:
TU Chemnitz, TU Eindhoven, U Antwerpen.
NXP Semiconductors.

MOORE NICE !

Support and Thanks

Cooperation partners:

- MiIT working group in Chemnitz: Ulrike Baur, Lihong Feng, Jens Saak, André Schneider, Michael Striebel, Mohammed-Sahadet Hossain, . . .
- Former and current students: Patrick Kürschner, René Günzel, Tobias Rothaug,
- Heike Faßbender,
- Tatjana Stykel,
- Enrique Quintana-Ortí and the group at Universitat Jaume I, Castellón,
- Qimonda AG/TITAN group: Uwe Feldmann, Georg Denk, . . .
- NXP Semiconductors: Jan ter Maten, Wil Schilders, . . .
- . . .

Several talks at SCEE 2008 related to the topic of this presentation.

Overview

Introduction

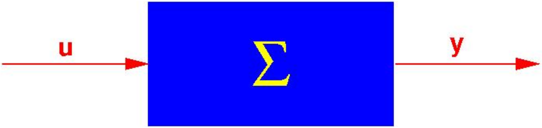
Dynamical Systems/DAEs

$$\Sigma : \left\{ \begin{array}{rcl} E\dot{x}(t) & = & f(t, x(t), u(t)), \quad x(t_0) = x_0, \\ y(t) & = & g(t, x(t), u(t)), \end{array} \right.$$

with

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

$E \in \mathbb{R}^{n \times n}$ singular $\rightsquigarrow$ differential-algebraic equations (DAEs) (DAEs), otherwise ordinary differential equations (ODEs).

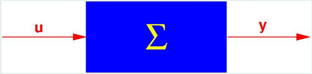


Model Reduction for Dynamical Systems

Original System

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

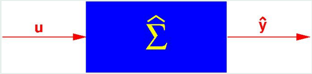
- states $x(t) \in \mathbb{R}^n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $y(t) \in \mathbb{R}^p$.



Reduced-Order System

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

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



Goal:

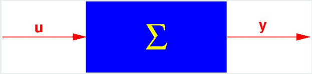
$\|y - \hat{y}\| <$ tolerance $\cdot \|u\|$ for all admissible input signals.

Model Reduction for Dynamical Systems

Original System

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

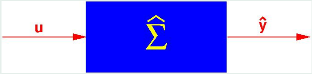
- states $x(t) \in \mathbb{R}^n$,
- inputs $u(t) \in \mathbb{R}^m$,
- outputs $y(t) \in \mathbb{R}^p$.



Reduced-Order System

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

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



Goal:

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

Model Reduction for Linear Systems

Linear Descriptor Systems

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

Linear Systems in Frequency Domain

Application of Laplace transformation ($x(t) \mapsto x(s)$, $\dot{x}(t) \mapsto sx(s)$) to linear descriptor system with $x(0) = 0$:

$$sEx(s) = Ax(s) + Bu(s), \quad y(s) = Bx(s) + Du(s),$$

yields I/O-relation in frequency domain:

$$y(s) = \Big(\underbrace{C(sE - A)^{-1}B + D}_{=:G(s)}\Big)u(s)$$

G is the transfer function of Σ.

Model Reduction for Linear Systems

Linear Descriptor Systems

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

Linear Systems in Frequency Domain

Application of Laplace transformation $(x(t) \mapsto x(s),\ \dot{x}(t) \mapsto sx(s))$ to linear descriptor system with $x(0)=0$:

$$sEx(s) = Ax(s) + Bu(s), \quad y(s) = Bx(s) + Du(s),$$

yields I/O-relation in frequency domain:

$$y(s) = \Big(\underbrace{C(sE-A)^{-1}B + D}_{=:G(s)} \Big) u(s)$$

G is the transfer function of Σ.

Model Reduction for Linear Descriptor Systems

Problem

Approximate the dynamical system

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

by reduced-order system

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

of order $r \ll n$, such that

$$\|y - \hat{y}\| = \|Gu - \hat{G}u\| \le \|G - \hat{G}\|\|u\| < \text{tolerance} \cdot \|u\|.$$

$\Longrightarrow$ Approximation problem: $\min_{\text{order}\,(\hat{G})\le r} \|G - \hat{G}\|$.

Model Reduction for Linear Descriptor Systems

Problem

Approximate the dynamical system

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

by reduced-order system

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

of order $r \ll n$, such that

$$\|y - \hat{y}\| = \|Gu - \hat{G}u\| \leq \|G - \hat{G}\|\|u\| < \text{tolerance} \cdot \|u\|.$$

$\Longrightarrow$ Approximation problem: $\min_{\text{order}\,(\hat{G}) \leq r} \|G - \hat{G}\|$.

Circuit Simulation

(Electronic) circuit simulation

- utilizes mathematical models to replicate the behavior of an actual electronic device or circuit.
- Simulating a circuit's behavior before actually building it greatly improves efficiency and provides insights into the behavior of electronics circuit designs.
- In particular, for integrated circuits,
  - the tooling (photomasks) is expensive,
  - breadboards are impractical,
  - probing the behavior of internal signals is extremely difficult.

Therefore almost all IC design relies heavily on simulation.

quoted from http://en.wikipedia.org/wiki/Circuit_simulation

Circuit Simulation

(Electronic) circuit simulation

- utilizes mathematical models to replicate the behavior of an actual electronic device or circuit.
- Simulating a circuit's behavior before actually building it greatly improves efficiency and provides insights into the behavior of electronics circuit designs.
- In particular, for integrated circuits,
  - the tooling (photomasks) is expensive,
  - breadboards are impractical,
  - probing the behavior of internal signals is extremely difficult.

Therefore almost all IC design relies heavily on simulation.

quoted from `http://en.wikipedia.org/wiki/Circuit_simulation`

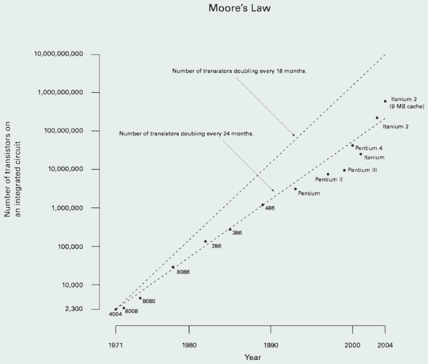
Circuit Simulation

The need for model reduction techniques

Progressive miniaturization

- Verification of VLSI/ULSI chip design requires high number of simulations for different input signals.

- **Moore's Law (1965/75)** states that the number of on-chip transistors doubles each 24 months.



Source: http://en.wikipedia.org/wiki/Image:Moores_law.svg

Circuit Simulation

The need for model reduction techniques

Progressive miniaturization

- Verification of VLSI/ULSI chip design requires high number of simulations for different input signals.
- **Moore's Law (1965/75)** $\rightsquigarrow$ steady increase of describing equations, i.e., network topology (Kirchhoff's laws) and characteristic element/semiconductor equations.
- Increase in packing density and multilayer technology requires modeling of interconncet to ensure that thermic/electro-magnetic effects do not disturb signal transmission.

Intel 4004 (1971)	Intel Core 2 Extreme (quad-core) (2007)
1 layer, 10μ technology	9 layers, $45nm$ technology
2,300 transistors	$> 8,200,000$ transistors
64 kHz clock speed	> 3 GHz clock speed.

Circuit Simulation

The need for model reduction techniques

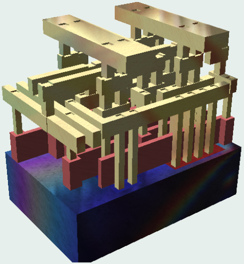
Progressive miniaturization

- Verification of VLSI/ULSI chip design requires high number of simulations for different input signals.
- **Moore's Law (1965/75)** $\rightsquigarrow$ steady increase of describing equations, i.e., network topology (Kirchhoff's laws) and characteristic element/semiconductor equations.
- Increase in packing density and multilayer technology requires modeling of interconncet to ensure that thermic/electro-magnetic effects do not disturb signal transmission.

Circuit Simulation
The need for model reduction techniques

Progressive miniaturization

- Verification of VLSI/ULSI chip design requires high number of simulations for different input signals.
- **Moore's Law (1965/75)** ⇝ steady increase of describing equations, i.e., network topology (Kirchhoff's laws) and characteristic element/semiconductor equations.
- Multilayer technology



Circuit Simulation

The need for model reduction techniques

Progressive miniaturization

- Verification of VLSI/ULSI chip design requires high number of simulations for different input signals.

$\rightsquigarrow$ Clear need for model reduction techniques in order to facilitate or even enable circuit simulation for current and future VLSI design.

Circuit Simulation

The need for model reduction techniques

Progressive miniaturization

- Verification of VLSI/ULSI chip design requires high number of simulations for different input signals.

⇝ Clear need for model reduction techniques in order to facilitate or even enable circuit simulation for current and future VLSI design.

Here: linear systems, they occur in micro electronics through modified nodal analysis (MNA) for RLC networks. e.g., when

- decoupling large linear subcircuits,
- modeling transmission lines,
- modeling pin packages in VLSI chips,
- modeling circuit elements described by Maxwell's equation using partial element equivalent circuits (PEEC).

Goals of model reduction

- Automatic generation of compact models.
- Satisfy desired error tolerance for all admissible input signals, i.e., want

$$\|y - \hat{y}\| < \text{tolerance} \cdot \|u\| \qquad \forall u \in L_2(\mathbb{R}, \mathbb{R}^m).$$

$\Longrightarrow$ Need computable error bound/estimate!

- Preserve physical properties:
  - stability (poles of G in $\mathbb{C}^-$, i.e., $\Lambda(A) \subset \mathbb{C}^-$),
  - minimum phase (zeroes of G in $\mathbb{C}^-$),
  - passivity

Goals of model reduction

- Automatic generation of compact models.
- Satisfy desired error tolerance for all admissible input signals, i.e., want

$$\|y - \hat{y}\| < \text{tolerance} \cdot \|u\| \qquad \forall u \in L_2(\mathbb{R}, \mathbb{R}^m).$$

$\Longrightarrow$ Need computable error bound/estimate!

- Preserve physical properties:
  - stability (poles of G in $\mathbb{C}^-$, i.e., $\Lambda(A) \subset \mathbb{C}^-$),
  - minimum phase (zeroes of G in $\mathbb{C}^-$),
  - passivity

Goals of model reduction

- Automatic generation of compact models.
- Satisfy desired error tolerance for all admissible input signals, i.e., want

$$\|y - \hat{y}\| < \text{tolerance} \cdot \|u\| \qquad \forall u \in L_2(\mathbb{R}, \mathbb{R}^m).$$

$\Longrightarrow$ Need computable error bound/estimate!

- Preserve physical properties:
  - stability (poles of G in $\mathbb{C}^-$, i.e., $\Lambda(A) \subset \mathbb{C}^-$),
  - minimum phase (zeroes of G in $\mathbb{C}^-$),
  - passivity

Goals of model reduction

- Automatic generation of compact models.
- Satisfy desired error tolerance for all admissible input signals, i.e., want

$$\|y - \hat{y}\| < \text{tolerance} \cdot \|u\| \qquad \forall u \in L_2(\mathbb{R}, \mathbb{R}^m).$$

$\Longrightarrow$ Need computable error bound/estimate!

- Preserve physical properties:
  - stability (poles of G in $\mathbb{C}^-$, i.e., $\Lambda(A) \subset \mathbb{C}^-$),
  - minimum phase (zeroes of G in $\mathbb{C}^-$),
  - passivity

Goals of model reduction

- Automatic generation of compact models.
- Satisfy desired error tolerance for all admissible input signals, i.e., want

$$\|y - \hat{y}\| < \text{tolerance} \cdot \|u\| \qquad \forall u \in L_2(\mathbb{R}, \mathbb{R}^m).$$

$\Longrightarrow$ Need computable error bound/estimate!

- Preserve physical properties:
  - stability (poles of G in $\mathbb{C}^-$, i.e., $\Lambda(A) \subset \mathbb{C}^-$),
  - minimum phase (zeroes of G in $\mathbb{C}^-$),
  - passivity

Goals of model reduction

- Automatic generation of compact models.
- Satisfy desired error tolerance for all admissible input signals, i.e., want

$$\|y - \hat{y}\| < \text{tolerance} \cdot \|u\| \qquad \forall u \in L_2(\mathbb{R}, \mathbb{R}^m).$$

$\Longrightarrow$ Need computable error bound/estimate!

- Preserve physical properties:
  - stability (poles of G in $\mathbb{C}^-$, i.e., $\Lambda(A) \subset \mathbb{C}^-$),
  - minimum phase (zeroes of G in $\mathbb{C}^-$),
  - passivity:

$$\int_{-\infty}^{t} u(\tau)^T y(\tau)\, d\tau \geq 0 \quad \forall t \in \mathbb{R}, \quad \forall u \in L_2(\mathbb{R}, \mathbb{R}^m).$$

("system does not generate energy").

Goals of model reduction

- Automatic generation of compact models.
- Satisfy desired error tolerance for all admissible input signals, i.e., want

$$\|y-\hat{y}\| < \text{tolerance} \cdot \|u\| \qquad \forall u \in L_2(\mathbb{R}, \mathbb{R}^m).$$

$\Longrightarrow$ Need computable error bound/estimate!

- Preserve physical properties:
  - stability (poles of G in $\mathbb{C}^-$, i.e., $\Lambda(A) \subset \mathbb{C}^-$),
  - minimum phase (zeroes of G in $\mathbb{C}^-$),
  - passivity

A variety of methods for linear model reduction exist (e.g., moment-matching, rational interpolation, . . .), here we only consider system-theoretic methods which have advantageous theoretical properties, but are often considered not applicable for really large-scale problems.

Balanced Truncation

Motivation: best approximation using SVD

Theorem: (Schmidt-Mirsky/Eckart-Young)

Best rank-r approximation to $X \in \mathbb{R}^{n_x \times n_y}$ w.r.t. spectral norm:

$$\widehat{X} = \sum\nolimits_{j=1}^{r} \sigma_j u_j v_j^T,$$

where $X = U\Sigma V^T$ is the singular value decomposition (SVD) of X, where $U = [u_1, \ldots]$, $V = [v_1, \ldots]$, $\Sigma = \operatorname{diag}(\sigma_1, \ldots)$.

The approximation error is $\|X - \widehat{X}\|_2 = \sigma_{r+1}$.

Balanced Truncation

Motivation: best approximation using SVD

Theorem: (Schmidt-Mirsky/Eckart-Young)

Best rank-r approximation to $X \in \mathbb{R}^{n_x \times n_y}$ w.r.t. spectral norm:

$$\widehat{X} = \sum_{j=1}^{r} \sigma_j u_j v_j^T,$$

where $X = U \Sigma V^T$ is the singular value decomposition (SVD) of X, where $U = [u_1, \ldots]$, $V = [v_1, \ldots]$, $\Sigma = \operatorname{diag}(\sigma_1, \ldots)$.

The approximation error is $\|X - \widehat{X}\|_2 = \sigma_{r+1}$.

Idea for dimension reduction

Instead of X save $u_1, \ldots, u_r$, $\sigma_1 v_1, \ldots, \sigma_r v_r$.

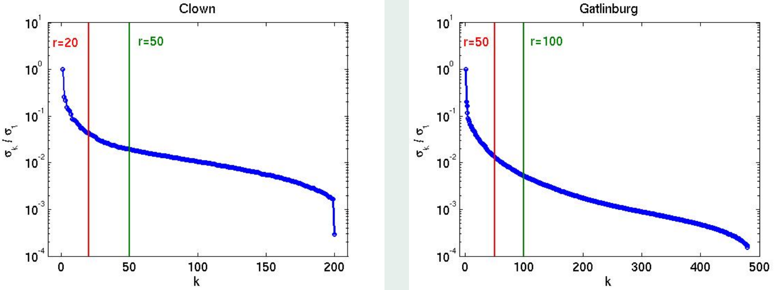
$\Rightarrow$ memory $= r \times (n_x + n_y)$ instead of $n_x \times n_y$.

Balanced Truncation

Motivation: best approximation using SVD

Data compression via SVD works, if the singular values decay (exponentially).

Singular Value Decay



Balanced Truncation

The Basic Ideas (E nonsingular)

Idea:

- A system Σ, realized by (A, B, C, D, E), is called balanced, if solutions P, Q of the Lyapunov equations

$$APE^T + EPA^T + BB^T \;=\; 0, \quad A^TQE + E^TQA + C^TC \;=\; 0,$$

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

- $\{\sigma_1, \ldots, \sigma_n\}$ are the Hankel singular values (HSVs) of Σ.
- Compute balanced realization of the system via system equivalence transformation ($S, T \in \mathbb{R}^{n\times n}$ nonsingular)

$$\mathcal{T} : (A, B, C, D, E) \mapsto (TAS, TB, CS, D, TES)$$

$$= \left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \begin{bmatrix} C_1 & C_2 \end{bmatrix}, D, \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix} \right)$$

- Truncation $\rightsquigarrow (\hat{A}, \hat{B}, \hat{C}, \hat{D}, \hat{E}) = (A_{11}, B_1, C_1, D, E_{11})$.

Balanced Truncation

The Basic Ideas (E nonsingular)

Idea:

- A system Σ, realized by (A, B, C, D, E), is called balanced, if solutions P, Q of the Lyapunov equations

$$APE^T + EPA^T + BB^T = 0, \quad A^TQE + E^TQA + C^TC = 0,$$

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

- $\{\sigma_1, \ldots, \sigma_n\}$ are the Hankel singular values (HSVs) of Σ.
- Compute balanced realization of the system via system equivalence transformation ($S, T \in \mathbb{R}^{n\times n}$ nonsingular)

$$\mathcal{T} : (A, B, C, D, E) \mapsto (TAS, TB, CS, D, TES)$$

$$= \left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \begin{bmatrix} C_1 & C_2 \end{bmatrix}, D, \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix} \right)$$

- Truncation $\rightsquigarrow (\hat{A}, \hat{B}, \hat{C}, \hat{D}, \hat{E}) = (A_{11}, B_1, C_1, D, E_{11})$.

Balanced Truncation

The Basic Ideas (E nonsingular)

Idea:

- A system Σ, realized by (A, B, C, D, E), is called balanced, if solutions P, Q of the Lyapunov equations

$$APE^T + EPA^T + BB^T = 0, \quad A^TQE + E^TQA + C^TC = 0,$$

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

- $\{\sigma_1, \ldots, \sigma_n\}$ are the Hankel singular values (HSVs) of Σ.
- Compute balanced realization of the system via system equivalence transformation ($S, T \in \mathbb{R}^{n \times n}$ nonsingular)

$$\mathcal{T} : (A, B, C, D, E) \mapsto (TAS, TB, CS, D, TES)$$

$$= \left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \begin{bmatrix} C_1 & C_2 \end{bmatrix}, D, \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix} \right)$$

- Truncation $\rightsquigarrow (\hat{A}, \hat{B}, \hat{C}, \hat{D}, \hat{E}) = (A_{11}, B_1, C_1, D, E_{11})$.

Balanced Truncation

The Basic Ideas (E nonsingular)

Idea:

- A system Σ, realized by (A, B, C, D, E), is called balanced, if solutions P, Q of the Lyapunov equations

$$APE^T + EPA^T + BB^T \;=\; 0, \quad A^TQE + E^TQA + C^TC \;=\; 0,$$

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

- $\{\sigma_1, \ldots, \sigma_n\}$ are the Hankel singular values (HSVs) of Σ.
- Compute balanced realization of the system via system equivalence transformation ($S, T \in \mathbb{R}^{n\times n}$ nonsingular)

$$\mathcal{T} : (A, B, C, D, E) \mapsto (TAS, TB, CS, D, TES)$$

$$= \left(\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \begin{bmatrix} C_1 & C_2 \end{bmatrix}, D, \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix} \right)$$

- Truncation $\rightsquigarrow$ $(\hat{A}, \hat{B}, \hat{C}, \hat{D}, \hat{E}) = (A_{11}, B_1, C_1, D, E_{11})$.

Balanced Truncation

The Basic Ideas (E nonsingular)

Motivation:

HSV are system invariants: they are preserved under $\mathcal{T}$ and determine the energy transfer given by the Hankel map

$$\mathcal{H} : L_2(-\infty, 0) \mapsto L_2(0, \infty) : u_- \mapsto y_+.$$

Balanced Truncation

The Basic Ideas (E nonsingular)

Motivation:

HSV are system invariants: they are preserved under $\mathcal{T}$ and determine the energy transfer given by the Hankel map

$$\mathcal{H} : L_2(-\infty, 0) \mapsto L_2(0, \infty) : u_- \mapsto y_+.$$

In balanced coordinates ... energy transfer from u_- to y_+:

$$E := \sup_{\substack{u \in L_2(-\infty,0] \\ x(0)=x_0}} \frac{\int\limits_0^\infty y(t)^T y(t)\, dt}{\int\limits_{-\infty}^0 u(t)^T u(t)\, dt} = \frac{1}{\|x_0\|_2} \sum_{j=1}^n \sigma_j^2 x_{0,j}^2$$

Balanced Truncation

The Basic Ideas (E nonsingular)

Motivation:

HSV are system invariants: they are preserved under $\mathcal{T}$ and determine the energy transfer given by the Hankel map

$$\mathcal{H} : L_2(-\infty, 0) \mapsto L_2(0, \infty) : u_- \mapsto y_+.$$

In balanced coordinates …energy transfer from u_- to y_+:

$$E := \sup_{\substack{u \in L_2(-\infty,0] \\ x(0)=x_0}} \frac{\int\limits_0^\infty y(t)^T y(t)\, dt}{\int\limits_{-\infty}^0 u(t)^T u(t)\, dt} = \frac{1}{\|x_0\|_2} \sum_{j=1}^n \sigma_j^2 x_{0,j}^2$$

$\Longrightarrow$ Truncate states corresponding to “small” HSVs

$\Longrightarrow$ complete analogy to best approximation via SVD!

Implementation: SR Method

1. Compute Cholesky factors of the solutions of the Lyapunov equations,
$$P = S^T S, \quad E^T Q E = R^T R.$$
2. Compute SVD
$$S R^T = [\, U_1,\ U_2 \,] \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}.$$
3. Set
$$W = (RE^{-1})^T V_1 \Sigma_1^{-1/2}, \qquad V = S^T U_1 \Sigma_1^{-1/2}.$$
4. Reduced model is $(W^T AV, W^T B, CV, D, \underbrace{W^T EV}_{=I_r})$.

Remark: Low-rank (rectangular) approximations of S, R can be computed directly using several methods, e.g. sign function [B./Quintana-Ortí] and methods for large, sparse problems (⇝ later) .

Balanced Truncation

The Basic Ideas (E nonsingular)

Implementation: SR Method

1. Compute Cholesky factors of the solutions of the Lyapunov equations,
$$P = S^T S, \quad E^T Q E = R^T R.$$
2. Compute SVD
$$SR^T = [\, U_1,\ U_2 \,] \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}.$$
3. Set
$$W = (RE^{-1})^T V_1 \Sigma_1^{-1/2}, \qquad V = S^T U_1 \Sigma_1^{-1/2}.$$
4. Reduced model is $(W^T AV, W^T B, CV, D, \underbrace{W^T EV}_{=I_r})$.

Remark: Low-rank (rectangular) approximations of S, R can be computed directly using several methods, e.g. sign function [B./Quintana-Ortí] and methods for large, sparse problems (⇝ later) ⇒ small-size SVD.

Balanced Truncation

The Basic Ideas (E nonsingular)

Implementation: SR Method

1. Compute Cholesky factors of the solutions of the Lyapunov equations,
$$P = S^T S, \quad E^T Q E = R^T R.$$
2. Compute SVD
$$S R^T = [\, U_1, \; U_2 \,] \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}.$$
3. Set
$$W = (R E^{-1})^T V_1 \Sigma_1^{-1/2}, \qquad V = S^T U_1 \Sigma_1^{-1/2}.$$
4. Reduced model is $(W^T A V, W^T B, C V, D, \underbrace{W^T E V}_{=I_r})$.

Remark: Reduced-order model with $E \neq I_r$ can be computed using balancing-free SR method [Safonov/Chiang '89, Stykel '04].

Balancing-Related Model Reduction

Assuming $E = I_n$ for simplicity.

Basic Principle of Balanced Truncation

Given positive semidefinite matrices $P = S^T S$, $Q = R^T R$, compute balancing state-space transformation so that

$$P = Q = \mathrm{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma, \quad \sigma_1 \geq \ldots \geq \sigma_n \geq 0,$$

and truncate corresponding realization at size r with $\sigma_r > \sigma_{r+1}$.

Balancing-Related Model Reduction

Assuming $E = I_n$ for simplicity.

Basic Principle of Balanced Truncation

Given positive semidefinite matrices $P = S^T S$, $Q = R^T R$, compute balancing state-space transformation so that

$$P = Q = \mathrm{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma, \quad \sigma_1 \geq \ldots \geq \sigma_n \geq 0,$$

and truncate corresponding realization at size r with $\sigma_r > \sigma_{r+1}$.

Classical Balanced Truncation (BT) Mullis/Roberts '76, Moore '81

- $P =$ controllability Gramian of system given by (A, B, C, D).
- $Q =$ observability Gramian of system given by (A, B, C, D).
- P, Q solve dual Lyapunov equations

$$AP + PA^T + BB^T = 0, \qquad A^T Q + QA + C^T C = 0.$$

- Need stability of A!

Balancing-Related Model Reduction

Assuming $E = I_n$ for simplicity.

Basic Principle of Balanced Truncation

Given positive semidefinite matrices $P = S^T S$, $Q = R^T R$, compute balancing state-space transformation so that

$$P = Q = \mathrm{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma, \quad \sigma_1 \geq \ldots \geq \sigma_n \geq 0,$$

and truncate corresponding realization at size r with $\sigma_r > \sigma_{r+1}$.

LQG Balanced Truncation (LQGBT)

Jonckheere/Silverman '83

- $P/Q =$ controllability/observability Gramian of closed-loop system based on LQG compensator.
- P, Q solve dual algebraic Riccati equations (AREs)

$$\begin{aligned} 0 &= AP + PA^T - PC^T CP + B^T B, \\ 0 &= A^T Q + QA - QBB^T Q + C^T C. \end{aligned}$$

- Can be applied to unstable systems!

Balancing-Related Model Reduction

Assuming $E = I_n$ for simplicity.

Basic Principle of Balanced Truncation

Given positive semidefinite matrices $P = S^T S$, $Q = R^T R$, compute balancing state-space transformation so that

$$P = Q = \mathrm{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma, \quad \sigma_1 \geq \ldots \geq \sigma_n \geq 0,$$

and truncate corresponding realization at size r with $\sigma_r > \sigma_{r+1}$.

Balanced Stochastic Truncation (BST)

Desai/Pal '84, Green '88

- $P =$ controllability Gramian of system given by (A, B, C, D), i.e., solution of Lyapunov equation $AP + PA^T + BB^T = 0$.
- $Q =$ observability Gramian of right spectral factor of power spectrum of system given by (A, B, C, D), i.e., solution of ARE

$$\hat{A}^T Q + Q\hat{A} + QB_W(DD^T)^{-1}B_W^T Q + C^T(DD^T)^{-1}C = 0,$$

where $\hat{A} := A - B_W(DD^T)^{-1}C$, $B_W := BD^T + PC^T$.

Balancing-Related Model Reduction

Assuming $E = I_n$ for simplicity.

Basic Principle of Balanced Truncation

Given positive semidefinite matrices $P = S^T S$, $Q = R^T R$, compute balancing state-space transformation so that

$$P = Q = \mathrm{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma, \quad \sigma_1 \geq \ldots \geq \sigma_n \geq 0,$$

and truncate corresponding realization at size r with $\sigma_r > \sigma_{r+1}$.

Positive-Real Balanced Truncation (PRBT) GREEN '88

- Based on positive-real equations, related to positive real (Kalman-Yakubovich-Popov-Anderson) lemma.
- P, Q solve dual AREs

$$\begin{aligned} 0 &= \bar{A}P + P\bar{A}^T + PC^T\bar{R}^{-1}CP + B\bar{R}^{-1}B^T, \\ 0 &= \bar{A}^TQ + Q\bar{A} + QB\bar{R}^{-1}B^TQ + C^T\bar{R}^{-1}C, \end{aligned}$$

where $\bar{R} = D + D^T$, $\bar{A} = A - B\bar{R}^{-1}C$.

Balancing-Related Model Reduction

Assuming $E = I_n$ for simplicity.

Basic Principle of Balanced Truncation

Given positive semidefinite matrices $P = S^T S$, $Q = R^T R$, compute balancing state-space transformation so that

$$P = Q = \operatorname{diag}(\sigma_1, \ldots, \sigma_n) = \Sigma, \quad \sigma_1 \geq \ldots \geq \sigma_n \geq 0,$$

and truncate corresponding realization at size r with $\sigma_r > \sigma_{r+1}$.

Other Balancing-Based Methods

- Bounded-real balanced truncation (BRBT) – based on bounded real lemma [Opdenacker/Jonckheere '88];
- H_∞ balanced truncation (HinfBT) – closed-loop balancing based on H_∞ compensator [Mustafa/Glover '91].

Both approaches require solution of dual AREs.

- Frequency-weighted versions of the above approaches.

Balancing-Related Model Reduction

Properties

- Guaranteed preservation of physical properties like
  - stability (all),
  - passivity (PRBT), ⇝ cf. Oral CS 3C (T. Stykel) Thu, 16:00h,
  - minimum phase (BST).
- Computable error bounds, e.g.,

$$\begin{aligned}
\text{BT:} \quad \|G-\hat{G}\|_\infty &\leq 2\sum_{j=r+1}^{n}\sigma_j^{BT},\\
\text{LQGBT:} \quad \|G-\hat{G}\|_\infty &\leq \; 2\sum_{j=r+1}^{n}\frac{\sigma_j^{LQG}}{\sqrt{1+(\sigma_j^{LQG})^2}},\\
\text{BST:} \quad \|G-\hat{G}\|_\infty &\leq \left(\prod_{j=r+1}^{n}\frac{1+\sigma_j^{BST}}{1-\sigma_j^{BST}}-1\right)\|G\|_\infty,
\end{aligned}$$

- Can be combined with singular perturbation approximation for steady-state performance.
- Computations can be modularized.

Application to Descriptor Systems

Now: E singular.

Often, one finds statements that BT can be based on the generalized Lyapunov equations

$$APE^T + EPA^T + BB^T \;=\; 0, \quad A^TQE + E^TQA + C^TC \;=\; 0, \quad (1)$$

e.g.,

[1] J. Phillips, L.M. Silveira.
Poor Man's TBR: A Simple Model Reduction Scheme.
Proc. DATE 2004, Vol. 2.

[2] J.R. Phillips, L.M. Silveira.
Poor man's TBR: A simple model reduction scheme.
IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., 24(1):43–55, 2005.

Application to Descriptor Systems

Now: E singular.

Often, one finds statements that BT can be based on the generalized Lyapunov equations

$$APE^T + EPA^T + BB^T \;=\; 0, \quad A^TQE + E^TQA + C^TC \;=\; 0, \quad (1)$$

e.g.,

1. J. Phillips, L.M. Silveira.
   Poor Man's TBR: A Simple Model Reduction Scheme.
   Proc. DATE 2004, Vol. 2.
2. J.R. Phillips, L.M. Silveira.
   Poor man's TBR: A simple model reduction scheme.
   IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., 24(1):43–55, 2005.

This is wrong in general! — (1) may or may not have solutions, no matter whether the associated system is asymptotically stable or not!

Thus, model reduction algorithms should not be based on (1)!

Application to Descriptor Systems

- Fully developed theory and numerical algorithms for balanced truncation [STYKEL '02–'08], based on projected generalized Lyapunov equations

$$\begin{aligned} APE^T + EPA^T + \mathcal{P}_\ell BB^T\mathcal{P}_\ell^T &= 0, \quad P = \mathcal{P}_r P, \\ A^T QE + E^T QA + \mathcal{P}_r^T C^T C\mathcal{P}_r &= 0, \quad Q = Q\mathcal{P}_\ell, \end{aligned}$$

  where $\mathcal{P}_r, \mathcal{P}_\ell$ are the spectral projectors onto the right and left deflating subspaces of $\lambda E - A$ corresponding to the finite eigenvalues.

- Theory and algorithms are based implicitly on Weierstraß canonical form

$$\lambda E - A = T \begin{bmatrix} \lambda I_{n_f} - J^0 & 0 \\ 0 & \lambda N - J^\infty \end{bmatrix} S^{-1},$$

  where

  - J^0 contains finite eigenvalues,
  - $N \in \mathbb{R}^{n_\infty \times n_\infty}$ is nilpotent of index ν ($N^\nu = 0,\ N^{\nu-1} \neq 0$), and
  - $n_\infty = n - n_f$ is the number of infinite eigenvalues.

Application to Descriptor Systems

- Fully developed theory and numerical algorithms for balanced truncation [Stykel '02–'08], based on projected generalized Lyapunov equations

$$\begin{array}{rcll} APE^T + EPA^T + \mathcal{P}_\ell BB^T\mathcal{P}_\ell^T & = & 0, & P = \mathcal{P}_r P, \\ A^TQE + E^TQA + \mathcal{P}_r^T C^T C\mathcal{P}_r & = & 0, & Q = Q\mathcal{P}_\ell, \end{array}$$

  where $\mathcal{P}_r, \mathcal{P}_\ell$ are the spectral projectors onto the right and left deflating subspaces of $\lambda E - A$ corresponding to the finite eigenvalues.

- Theory and algorithms are based implicitly on Weierstraß canonical form

$$\lambda E - A = T \begin{bmatrix} \lambda I_{n_f} - J^0 & 0 \\ 0 & \lambda N - J^\infty \end{bmatrix} S^{-1},$$

  where

  - J^0 contains finite eigenvalues,
  - $N \in \mathbb{R}^{n_\infty \times n_\infty}$ is nilpotent of index ν ($N^\nu = 0,\ N^{\nu-1} \neq 0$), and
  - $n_\infty = n - n_f$ is the number of infinite eigenvalues.

Application to Descriptor Systems

- Here: use algorithm mathematically equivalent to [STYKEL '02–'08] based on explicit decomposition

  $$G(s) \;=\; G_f(s) + G_\infty(s),$$

  where $G_f(s)$, $G_\infty(s)$ correspond to finite, infinite poles, resp.
- lock-diagonalization of $\lambda E - A$:

  $$\lambda\hat{E} - \hat{A} := U(\lambda E - A)V^{-1} = \lambda \begin{bmatrix} E_0 & 0 \\ 0 & E_\infty \end{bmatrix} - \begin{bmatrix} A_f & 0 \\ 0 & A_\infty \end{bmatrix},$$

  and setting

  $$\hat{B} := UB =: \begin{bmatrix} B_f \\ B_\infty \end{bmatrix}, \quad \hat{C} := CV^{-1} =: [\, C_f \;\, C_\infty \,], \quad \hat{D} := D.$$

Application to Descriptor Systems

- Here: use algorithm mathematically equivalent to [Stykel '02–'08] based on explicit decomposition

$$G(s) \;=\; G_f(s) + G_\infty(s),$$

where $G_f(s), G_\infty(s)$ correspond to finite, infinite poles, resp.

- This is achieved by computing block-diagonalization of $\lambda E - A$:

$$\lambda\hat{E} - \hat{A} := U(\lambda E - A)V^{-1} = \lambda \begin{bmatrix} E_0 & 0 \\ 0 & E_\infty \end{bmatrix} - \begin{bmatrix} A_f & 0 \\ 0 & A_\infty \end{bmatrix},$$

and setting

$$\hat{B} := UB =: \begin{bmatrix} B_f \\ B_\infty \end{bmatrix}, \quad \hat{C} := CV^{-1} =: [\, C_f \;\, C_\infty \,], \quad \hat{D} := D.$$

Application to Descriptor Systems

- Block-diagonalization of $\lambda E - A$:

$$\lambda \hat{E} - \hat{A} := U(\lambda E - A)V^{-1} = \lambda \begin{bmatrix} E_0 & 0 \\ 0 & E_\infty \end{bmatrix} - \begin{bmatrix} A_f & 0 \\ 0 & A_\infty \end{bmatrix},$$

and setting

$$\hat{B} := UB =: \begin{bmatrix} B_f \\ B_\infty \end{bmatrix}, \quad \hat{C} := CV^{-1} =: [\, C_f \; C_\infty \,], \quad \hat{D} := D.$$

Then

$$\begin{aligned} G(s) &= C(sE - A)^{-1}B + D = \hat{C}(s\hat{E} - \hat{A})^{-1}\hat{B} + \hat{D} \\ &= \begin{bmatrix} C_f & C_\infty \end{bmatrix} \begin{bmatrix} sE_f - A_f & \\ & sE_\infty - A_\infty \end{bmatrix}^{-1} \begin{bmatrix} B_f \\ B_\infty \end{bmatrix} + D \\ &= \underbrace{C_f(sE_f - A_f)^{-1}B_f}_{=:G_f(s)} + \underbrace{C_\infty(sE_\infty - A_\infty)^{-1}B_\infty + D}_{:=G_\infty(s)}. \end{aligned}$$

$\rightsquigarrow$ apply BT to G_f $\rightsquigarrow$ $\hat{G}_f$, compute minimal realization of G_∞.

Application to Descriptor Systems

Implementation: block diagonalization

1. **(Block-triangular form)** Use disk function method (quadratically convergent, matrix multiplication rich, inverse-free algorithm) to obtain Q, Z orthogonal such that
$$Q^T(\lambda E - A)Z = \lambda \begin{bmatrix} E_f & W_E \\ 0 & E_\infty \end{bmatrix} - \begin{bmatrix} A_f & W_A \\ 0 & A_\infty \end{bmatrix}.$$
2. **(Block-diagonal form)** Solve generalized Sylvester equation
$$A_f Y + Z A_\infty + W_A = 0, \quad E_f Y + Z E_\infty + W_E = 0.$$

Application to Descriptor Systems

Implementation: block diagonalization

1. **(Block-triangular form)** Use disk function method (quadratically convergent, matrix multiplication rich, inverse-free algorithm) to obtain Q, Z orthogonal such that

$$Q^T(\lambda E - A)Z = \lambda \begin{bmatrix} E_f & W_E \\ 0 & E_\infty \end{bmatrix} - \begin{bmatrix} A_f & W_A \\ 0 & A_\infty \end{bmatrix}.$$

2. **(Block-diagonal form)** Solve generalized Sylvester equation

$$A_f Y + Z A_\infty + W_A = 0, \quad E_f Y + Z E_\infty + W_E = 0.$$

Then

$$\begin{aligned} \lambda \hat{E} - \hat{A} \quad &:= \quad \begin{bmatrix} I & Z \\ 0 & I \end{bmatrix} Q^T(\lambda E - A)Z \begin{bmatrix} I & Y \\ 0 & I \end{bmatrix} \\ &= \quad \lambda \begin{bmatrix} E_f & 0 \\ 0 & E_\infty \end{bmatrix} - \begin{bmatrix} A_f & 0 \\ 0 & A_\infty \end{bmatrix}. \end{aligned}$$

Application to Descriptor Systems

Implementation: block diagonalization

1. **(Block-triangular form)** Use disk function method (quadratically convergent, matrix multiplication rich, inverse-free algorithm) to obtain Q, Z orthogonal such that

$$Q^T(\lambda E - A)Z = \lambda \begin{bmatrix} E_f & W_E \\ 0 & E_\infty \end{bmatrix} - \begin{bmatrix} A_f & W_A \\ 0 & A_\infty \end{bmatrix}.$$

2. **(Block-diagonal form)** Solve generalized Sylvester equation

$$A_f Y + Z A_\infty + W_A = 0, \quad E_f Y + Z E_\infty + W_E = 0.$$

Simplification for index $\nu = 1$:

$$\left.\begin{array}{l} E_\infty = 0 \\ A_\infty \text{ nonsingular} \end{array}\right\} \Rightarrow \left\{\begin{array}{lcl} Y & = & -E_f^{-1} W_E, \\ Z & = & -(W_A - A_f Y)A_\infty^{-1}, \end{array}\right.$$

otherwise use appropriate solver for Sylvester equations, e.g., from SLICOT, see `www.slicot.org`.

Application to Descriptor Systems

Implementation: solution of Lyapunov equations

Solve

$$A_f P E_f^T + E_f P A_f^T + B_f B_f^T = 0, \quad A_f^T Q E_f + E_f^T Q A_f + C_f^T C_f = 0,$$

via dual gen. Newton it. for sign function [B./Claver/Quintana-Ortí '97]:

```
$A_0 \leftarrow A, \quad S_0 \leftarrow B, \quad R_0 \leftarrow C$
for $j = 0, 1, 2, \ldots$
    $A_{j+1} \leftarrow \frac{1}{\sqrt{2c_j}} \left( A_j + c_j^2 E_f A_j^{-1} E_f \right),$
    $S_{j+1} \leftarrow$ full-rank factor of $\frac{1}{\sqrt{2c_j}} \left[ S_j \quad c_j E_f A_j^{-1} S_j \right]$
    $R_{j+1} \leftarrow$ full-rank factor of $\frac{1}{\sqrt{2c_j}} \begin{bmatrix} R_j \\ c_j R_j A_j^{-1} E_f \end{bmatrix}$
```

$$\text{Set} \quad S := \frac{1}{\sqrt{2}} E_f^{-1} \lim_{j\to\infty} S_j = \text{ factor of } P,$$

$$R := \frac{1}{\sqrt{2}} \lim_{j\to\infty} R_j = \text{ factor of } E_f^T Q E_f.$$

Application to Descriptor Systems

Implementation: solution of Lyapunov equations

Solve

$$A_f P E_f^T + E_f P A_f^T + B_f B_f^T = 0, \quad A_f^T Q E_f + E_f^T Q A_f + C_f^T C_f = 0,$$

via dual gen. Newton it. for sign function [B./Claver/Quintana-Ortí '97]:

```
$A_0 \leftarrow A, \quad S_0 \leftarrow B, \quad R_0 \leftarrow C$
for $j = 0, 1, 2, \ldots$
    $A_{j+1} \leftarrow \frac{1}{\sqrt{2c_j}} \left( A_j + c_j^2 E_f A_j^{-1} E_f \right),$
    $S_{j+1} \leftarrow$ full-rank factor of $\frac{1}{\sqrt{2c_j}} \left[ S_j \quad c_j E_f A_j^{-1} S_j \right]$
    $R_{j+1} \leftarrow$ full-rank factor of $\frac{1}{\sqrt{2c_j}} \begin{bmatrix} R_j \\ c_j R_j A_j^{-1} E_f \end{bmatrix}$
```

Note: Full-rank factors are computed using rank-revealing LQ/QR factorization (RRLQ/RRQR) with respect to tolerance τ for rank determination, without accumulation of Q.

Application to Descriptor Systems

Implementation: minimal realization of $G_\infty(s)$

If index $\nu = 1$:

$$G_\infty(s) \equiv \hat{D} := D - C_\infty A_\infty^{-1} B_\infty .$$

Application to Descriptor Systems

Implementation: minimal realization of $G_\infty(s)$

If index $\nu = 1$:

$$G_\infty(s) \equiv \hat{D} := D - C_\infty A_\infty^{-1} B_\infty.$$

If no feed-through term allowed in simulation software:

$$\begin{aligned} \hat{\mathcal{A}} &:= \begin{bmatrix} \hat{A} & \\ & -I_m \end{bmatrix}, & \hat{\mathcal{B}} &:= \begin{bmatrix} \hat{B} \\ I_m \end{bmatrix}, \\ \hat{\mathcal{C}} &:= \begin{bmatrix} \hat{C} & \hat{D} \end{bmatrix}, & \hat{\mathcal{E}} &:= \begin{bmatrix} \hat{E} & \\ & 0_m \end{bmatrix}. \end{aligned}$$

$\Longrightarrow$

$$\begin{aligned} \hat{G}(s) &:= \hat{G}_f(s) + G_\infty(s) = \hat{C}(s\hat{E} - \hat{A})^{-1}\hat{B} + \hat{D} \\ &= \hat{\mathcal{C}}(s\hat{\mathcal{E}} - \hat{\mathcal{A}})^{-1}\hat{\mathcal{B}} \end{aligned}$$

Application to Descriptor Systems

Implementation: minimal realization of $G_\infty(s)$

If index $\nu > 1$: the McMillan degree $\hat{n}_\infty$ of $G_\infty(s)$ satisfies [Stykel '02/'04]

$$\hat{n}_\infty \leq \min\{\nu m, \nu p, n_\infty\}.$$

Corresponding minimal realization can be computed by applying balanced truncation with "zero" threshold for polynomial part $G_\infty(s)$; for details see [Stykel '02/'04].

Application to Descriptor Systems

Implementation: minimal realization of $G_\infty(s)$

If index $\nu > 1$: the McMillan degree $\hat{n}_\infty$ of $G_\infty(s)$ satisfies [Stykel '02/'04]

$$\hat{n}_\infty \leq \min\{\nu m, \nu p, n_\infty\}.$$

Corresponding minimal realization can be computed by applying balanced truncation with "zero" threshold for polynomial part $G_\infty(s)$; for details see [Stykel '02/'04].

In any case, $\hat{G}_\infty(s) = G_\infty(s)$ and thus,

$$G(s) - \hat{G}(s) = G_f(s) - \hat{G}_f(s),$$

therefore

$$\|G - \hat{G}\|_\infty \leq 2 \sum_{j=r+1}^{n_f} \sigma_j^f.$$

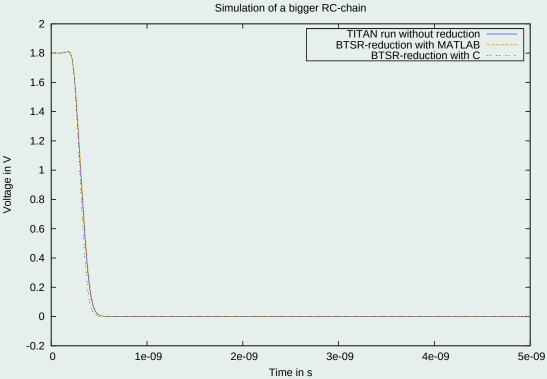
Application to Descriptor Systems

Numerical results

Descriptor systems BT algorithm was implemented in circuit simulator TITAN (Qimonda AG, Diploma thesis R. Günzel, 2008).

Example 1: small nonlinear circuit

297 resistors, 268 capacitors, 4 voltage sources, 8 MOSFETs.
Linear subcircuit of order $n = 297$ extracted, reduced to order $r = 31$.
TITAN simulation results:

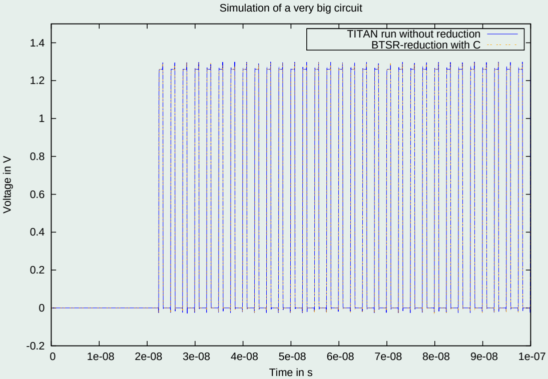


Application to Descriptor Systems
Numerical results

Descriptor systems BT algorithm was implemented in circuit simulator TITAN (Qimonda AG, Diploma thesis R. Günzel, 2008).

Example 2: industrial circuit

14,677 resistors, 15,404 capacitors, 14 voltage sources, 4,800 MOSFETs.
14 linear subcircuit of varying order extracted and reduced.
TITAN simulation results:



Application to Large-Scale, Sparse Systems

General misconception: complexity of BT $\mathcal{O}(n^3)$ – true for several implementations (e.g., MATLAB, SLICOT)!

Application to Large-Scale, Sparse Systems

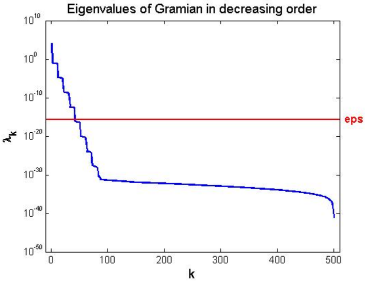
General misconception: complexity of BT $\mathcal{O}(n^3)$ – true for several implementations (e.g., MATLAB, SLICOT)!

Algorithmic ideas from numerical linear algebra (since $\sim$ 1997):

- Instead of Gramians P, Q or Cholesky factors thereof compute $S, R \in \mathbb{R}^{n \times k}$, $k \ll n$, such that

$$P \approx SS^T, \quad Q \approx RR^T.$$

- Compute S, R with problem-specific Lyapunov/Riccati solvers of "low" complexity directly.



Application to Large-Scale, Sparse Systems

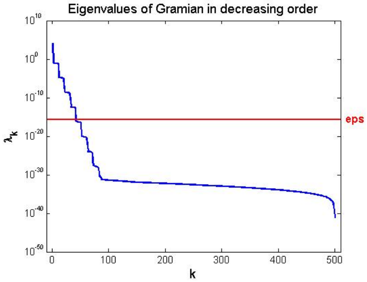
General misconception: complexity of BT $\mathcal{O}(n^3)$ – true for several implementations (e.g., MATLAB, SLICOT)!

Algorithmic ideas from numerical linear algebra (since $\sim$ 1997):

- Instead of Gramians P, Q or Cholesky factors thereof compute $S, R \in \mathbb{R}^{n\times k}$, $k \ll n$, such that

$$P \approx SS^T, \quad Q \approx RR^T.$$

- Compute S, R with problem-specific Lyapunov/Riccati solvers of "low" complexity directly.



$\rightsquigarrow$ need solver for large-scale matrix equations which computes S, R directly!

Application to Large-Scale, Sparse Systems

ADI Method for Lyapunov Equations

- For $A \in \mathbb{R}^{n\times n}$ stable, $B \in \mathbb{R}^{n\times m}$ $(w \ll n)$, consider Lyapunov equation
$$AX + XA^T = -BB^T.$$
- ADI Iteration: [Wachspress '88]
$$\begin{aligned}(A + p_k I)X_{(j-1)/2} &= -BB^T - X_{k-1}(A^T - p_k I)\\ (A + \overline{p_k} I){X_k}^T &= -BB^T - X_{(j-1)/2}(A^T - \overline{p_k} I)\end{aligned}$$
with parameters $p_k \in \mathbb{C}^-$ and $p_{k+1} = \overline{p_k}$ if $p_k \notin \mathbb{R}$.
- For $X_0 = 0$ and proper choice of p_k: $\lim\limits_{k\to\infty} X_k = X$ superlinear.
- Re-formulation using $X_k = Y_k Y_k^T$ yields iteration for Y_k...

Application to Large-Scale, Sparse Systems
ADI Method for Lyapunov Equations

- For $A \in \mathbb{R}^{n \times n}$ stable, $B \in \mathbb{R}^{n \times m}$ $(w \ll n)$, consider Lyapunov equation
$$AX + XA^T = -BB^T.$$
- ADI Iteration: [Wachspress '88]
$$\begin{aligned} (A + p_k I)X_{(j-1)/2} &= -BB^T - X_{k-1}(A^T - p_k I) \\ (A + \overline{p_k} I){X_k}^T &= -BB^T - X_{(j-1)/2}(A^T - \overline{p_k} I) \end{aligned}$$
with parameters $p_k \in \mathbb{C}^-$ and $p_{k+1} = \overline{p_k}$ if $p_k \notin \mathbb{R}$.
- For $X_0 = 0$ and proper choice of p_k: $\lim\limits_{k \to \infty} X_k = X$ superlinear.
- Re-formulation using $X_k = Y_k Y_k^T$ yields iteration for Y_k...

Application to Large-Scale, Sparse Systems

ADI Method for Lyapunov Equations

- For $A \in \mathbb{R}^{n\times n}$ stable, $B \in \mathbb{R}^{n\times m}$ $(w \ll n)$, consider Lyapunov equation
$$AX + XA^T = -BB^T.$$
- ADI Iteration: [Wachspress '88]
$$\begin{array}{rcl} (A + p_k I)X_{(j-1)/2} & = & -BB^T - X_{k-1}(A^T - p_k I) \\ (A + \overline{p_k} I){X_k}^T & = & -BB^T - X_{(j-1)/2}(A^T - \overline{p_k} I) \end{array}$$
with parameters $p_k \in \mathbb{C}^-$ and $p_{k+1} = \overline{p_k}$ if $p_k \notin \mathbb{R}$.
- For $X_0 = 0$ and proper choice of p_k: $\lim_{k\to\infty} X_k = X$ superlinear.
- Re-formulation using $X_k = Y_k Y_k^T$ yields iteration for Y_k...

Application to Large-Scale, Sparse Systems

ADI Method for Lyapunov Equations

- For $A \in \mathbb{R}^{n \times n}$ stable, $B \in \mathbb{R}^{n \times m}$ $(w \ll n)$, consider Lyapunov equation
$$AX + XA^T = -BB^T.$$
- ADI Iteration: [Wachspress '88]
$$\begin{aligned}(A + p_k I)X_{(j-1)/2} &= -BB^T - X_{k-1}(A^T - p_k I)\\ (A + \overline{p_k} I)X_k{}^T &= -BB^T - X_{(j-1)/2}(A^T - \overline{p_k} I)\end{aligned}$$
with parameters $p_k \in \mathbb{C}^-$ and $p_{k+1} = \overline{p_k}$ if $p_k \notin \mathbb{R}$.
- For $X_0 = 0$ and proper choice of p_k: $\lim\limits_{k \to \infty} X_k = X$ superlinear.
- Re-formulation using $X_k = Y_k Y_k^T$ yields iteration for Y_k...

Factored ADI Iteration

Lyapunov equation $AX + XA^T = -BB^T$.

Setting $X_k = Y_k Y_k^T$, some algebraic manipulations $\Longrightarrow$

Algorithm [Penzl '97, Li/White '02, B./Li/Penzl '99/'08]

```
$V_1 \;\leftarrow\; \sqrt{-2\mathrm{Re}\,(p_1)}(A + p_1 I)^{-1}B, \qquad Y_1 \;\leftarrow\; V_1$
FOR $j = 2, 3, \ldots$
    $V_k \leftarrow \sqrt{\frac{\mathrm{Re}\,(p_k)}{\mathrm{Re}\,(p_{k-1})}}\left(V_{k-1} - (p_k + \overline{p_{k-1}})(A + p_k I)^{-1}V_{k-1}\right),$
    $Y_k \leftarrow \mathtt{rrqr}\left(\left[\begin{array}{cc} Y_{k-1} & V_k \end{array}\right]\right)$    % column compression
```

At convergence, $Y_{k_{\max}} Y_{k_{\max}}^T \approx X$, where

$$\mathrm{range}\,(Y_{k_{\max}}) = \mathrm{range}\left(\left[\begin{array}{ccc} V_1 & \ldots & V_{k_{\max}} \end{array}\right]\right), \quad V_k = \Big[\;\Big] \in \mathbb{C}^{n\times m}.$$

Factored ADI Iteration

Lyapunov equation $AX + XA^T = -BB^T$.

Setting $X_k = Y_k Y_k^T$, some algebraic manipulations $\Longrightarrow$

Algorithm [Penzl '97, Li/White '02, B./Li/Penzl '99/'08]

```
$V_1 \leftarrow \sqrt{-2\mathrm{Re}\,(p_1)}(A + p_1 I)^{-1} B,$        $Y_1 \leftarrow V_1$
FOR $j = 2, 3, \ldots$
    $V_k \leftarrow \sqrt{\frac{\mathrm{Re}\,(p_k)}{\mathrm{Re}\,(p_{k-1})}}\left(V_{k-1} - (p_k + \overline{p_{k-1}})(A + p_k I)^{-1} V_{k-1}\right),$
    $Y_k \leftarrow \mathtt{rrqr}\left(\left[\begin{array}{cc} Y_{k-1} & V_k \end{array}\right]\right)$        % column compression
```

At convergence, $Y_{k_{\max}} Y_{k_{\max}}^T \approx X$, where

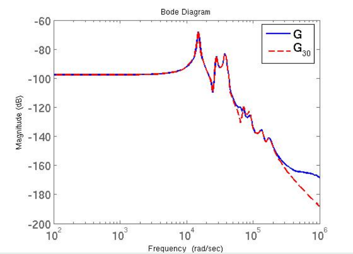
$$\mathrm{range}\,(Y_{k_{\max}}) = \mathrm{range}\left(\left[\begin{array}{ccc} V_1 & \ldots & V_{k_{\max}} \end{array}\right]\right), \quad V_k = \boxed{\vphantom{\Big|}} \in \mathbb{C}^{n \times m}.$$

Application to Large-Scale, Sparse Systems

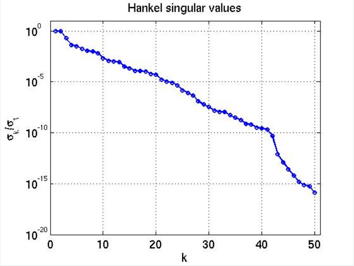
Numerical example: butterfly gyro

- FEM discretization of MEMS device (micro gyroscope) $\rightsquigarrow$ $n = 34,722,\ m = 1,\ p = 12.$
- Reduced model computed using BT with low-rank ADI for Lyapunov equations, $r = 30.$

Frequency Response Analysis



Hankel Singular Values



Application to Large-Scale, Sparse Systems

Recent developments for large-scale Lyapunov equations

- Improved ADI parameter selection strategies for non-real spectra [B./Mena/Saak '06/'08, Sabino '06, Truhar/Li/Tomljanović '08].
- Competitive Krylov subspace method based on projection onto

$$\mathcal{K}(A, B, k) \cup \mathcal{K}(A^{-1}, A^{-1}B, k) = \mathrm{span}\{b, A^{-1}b, Ab, A^{-2}b, A2b, \ldots\},$$

  called K-PIK [Simoncini '06/'07].
- Hybrid ADI + Galerkin projection [B./li/Truhar '08].
- Under development: Hybrid K-PIK + ADI, dominant poles as ADI parameters (with J. Saak, J. Rommes).
- Low-rank ADI for descriptor systems:
  - [Stykel '06/'08] for projected generalized Lyapunov equations;
  - [Heinkenschloss/Sorensen/Sun '06/'08] for Oseen-type systems;
  - [Freitas/Martins/Rommes '08, B. '07/'08] for different semi-explicit index-1 systems.

Application to Large-Scale, Sparse Systems

Recent developments for large-scale Lyapunov equations

- Improved ADI parameter selection strategies for non-real spectra [B./Mena/Saak '06/'08, Sabino '06, Truhar/Li/Tomljanović '08].
- Competitive Krylov subspace method based on projection onto

  $$\mathcal{K}(A, B, k) \cup \mathcal{K}(A^{-1}, A^{-1}B, k) = \mathrm{span}\{b, A^{-1}b, Ab, A^{-2}b, A2b, \ldots\},$$

  called K-PIK [Simoncini '06/'07].
- Hybrid ADI + Galerkin projection [B./li/Truhar '08].
- Under development: Hybrid K-PIK + ADI, dominant poles as ADI parameters (with J. Saak, J. Rommes).
- Low-rank ADI for descriptor systems:
  - [Stykel '06/'08] for projected generalized Lyapunov equations;
  - [Heinkenschloss/Sorensen/Sun '06/'08] for Oseen-type systems;
  - [Freitas/Martins/Rommes '08, B. '07/'08] for different semi-explicit index-1 systems.

Application to Large-Scale, Sparse Systems

Recent developments for large-scale Lyapunov equations

- Improved ADI parameter selection strategies for non-real spectra [B./Mena/Saak '06/'08, Sabino '06, Truhar/Li/Tomljanović '08].
- Competitive Krylov subspace method based on projection onto

  $$\mathcal{K}(A, B, k) \cup \mathcal{K}(A^{-1}, A^{-1}B, k) = \mathrm{span}\{b, A^{-1}b, Ab, A^{-2}b, A2b, \ldots\},$$

  called K-PIK [Simoncini '06/'07].
- Hybrid ADI + Galerkin projection [B./li/Truhar '08].
- Under development: Hybrid K-PIK + ADI, dominant poles as ADI parameters (with J. Saak, J. Rommes).
- Low-rank ADI for descriptor systems:
  - [Stykel '06/'08] for projected generalized Lyapunov equations;
  - [Heinkenschloss/Sorensen/Sun '06/'08] for Oseen-type systems;
  - [Freitas/Martins/Rommes '08, B. '07/'08] for different semi-explicit index-1 systems.

Application to Large-Scale, Sparse Systems

Recent developments for large-scale Lyapunov equations

- Improved ADI parameter selection strategies for non-real spectra [B./Mena/Saak '06/'08, Sabino '06, Truhar/Li/Tomljanović '08].
- Competitive Krylov subspace method based on projection onto

  $$\mathcal{K}(A, B, k) \cup \mathcal{K}(A^{-1}, A^{-1}B, k) = \operatorname{span}\{b, A^{-1}b, Ab, A^{-2}b, A2b, \ldots\},$$

  called K-PIK [Simoncini '06/'07].
- Hybrid ADI + Galerkin projection [B./li/Truhar '08].
- Under development: Hybrid K-PIK + ADI, dominant poles as ADI parameters (with J. Saak, J. Rommes).
- Low-rank ADI for descriptor systems:
  - [Stykel '06/'08] for projected generalized Lyapunov equations;
  - [Heinkenschloss/Sorensen/Sun '06/'08] for Oseen-type systems;
  - [Freitas/Martins/Rommes '08, B. '07/'08] for different semi-explicit index-1 systems.

Application to Large-Scale, Sparse Systems

Recent developments for large-scale Lyapunov equations

- Improved ADI parameter selection strategies for non-real spectra [B./Mena/Saak '06/'08, Sabino '06, Truhar/Li/Tomljanović '08].
- Competitive Krylov subspace method based on projection onto

  $$\mathcal{K}(A, B, k) \cup \mathcal{K}(A^{-1}, A^{-1}B, k) = \mathrm{span}\{b, A^{-1}b, Ab, A^{-2}b, A2b, \ldots\},$$

  called K-PIK [Simoncini '06/'07].
- Hybrid ADI + Galerkin projection [B./li/Truhar '08].
- Under development: Hybrid K-PIK + ADI, dominant poles as ADI parameters (with J. Saak, J. Rommes).
- Low-rank ADI for descriptor systems:
  - [Stykel '06/'08] for projected generalized Lyapunov equations;
  - [Heinkenschloss/Sorensen/Sun '06/'08] for Oseen-type systems;
  - [Freitas/Martins/Rommes '08, B. '07/'08] for different semi-explicit index-1 systems.

Application to Large-Scale, Sparse Systems

Recent developments for large-scale AREs

Recall: balancing-related model reduction methods like positive-real balancing ($\rightsquigarrow$ passivity-preserving) require solution of algebraic Riccati equations (AREs) of the form

$$W + A^T XE + E^T XA + E^T XGXE = 0. \tag{1}$$

- Various algorithms for dense matrices; e.g., implementation of PRBT for $E \neq I_n$ nonsingular [B./Quintana-Ortí×2 '04] based on sign function method.
- For large, sparse matrices: use Newton's method $\rightsquigarrow$ Newton step = solution of Lyapunov equation $\rightsquigarrow$ use low-rank ADI, obtain approximate solution in low-rank format [B./Li/Penzl '99/'00].

Application to Large-Scale, Sparse Systems

Recent developments for large-scale AREs

Recall: balancing-related model reduction methods like positive-real balancing ($\rightsquigarrow$ passivity-preserving) require solution of algebraic Riccati equations (AREs) of the form

$$W + A^T XE + E^T XA + E^T XGXE = 0. \qquad (1)$$

- Various algorithms for dense matrices; e.g., implementation of PRBT for $E \neq I_n$ nonsingular [B./Quintana-Ortí×2 '04] based on sign function method.
- For large, sparse matrices: use Newton's method $\rightsquigarrow$ Newton step $=$ solution of Lyapunov equation $\rightsquigarrow$ use low-rank ADI, obtain approximate solution in low-rank format [B./Li/Penzl '99/'00].

Application to Large-Scale, Sparse Systems

Recent developments for large-scale AREs

Recall: balancing-related model reduction methods like positive-real balancing ($\rightsquigarrow$ passivity-preserving) require solution of algebraic Riccati equations (AREs) of the form

$$W + A^T XE + E^T XA + E^T XGXE = 0. \tag{1}$$

Generalization to descriptor systems: Do not use (1)!

Theorem [B./Stykel '08]

Consider the projected ARE

$$\mathcal{P}_r^T Q \mathcal{P}_r + A^T XE + E^T XA + E^T XGXE = 0, \quad X = \mathcal{P}_\ell^T X \mathcal{P}_\ell. \tag{2}$$

with $G = G^T \geq 0$ and $Q = Q^T \geq 0$ and
$\mathcal{P}_r/\mathcal{P}_\ell$: projectors onto right/left defl. subspaces of $\lambda E - A$ wrt finite e-values.

If (E, A, G) is stabilizable and (E, A, Q) is detectable, then (2) has a unique stabilizing solution.

Algorithms based on Newton's method using various Lyapunov solvers for dense or large-scale problems [B./Stykel '08].

Application to Large-Scale, Sparse Systems

Recent developments for large-scale AREs

Recall: balancing-related model reduction methods like positive-real balancing ($\rightsquigarrow$ passivity-preserving) require solution of algebraic Riccati equations (AREs) of the form

$$W + A^T XE + E^T XA + E^T XGXE = 0. \qquad (1)$$

Generalization to descriptor systems: Do not use (1)!

Theorem [B./Stykel '08]

Consider the projected ARE

$$\mathcal{P}_r^T Q \mathcal{P}_r + A^T XE + E^T XA + E^T XGXE = 0, \quad X = \mathcal{P}_\ell^T X \mathcal{P}_\ell. \qquad (2)$$

with $G = G^T \geq 0$ and $Q = Q^T \geq 0$ and
$\mathcal{P}_r / \mathcal{P}_\ell$: projectors onto right/left defl. subspaces of $\lambda E - A$ wrt finite e-values.

If (E, A, G) is stabilizable and (E, A, Q) is detectable, then (2) has a unique stabilizing solution.

Algorithms based on Newton's method using various Lyapunov solvers for dense or large-scale problems [B./Stykel '08].

Approximate BT

- There is no exact BT.

Approximate BT

- There is no exact BT. Why?

Approximate BT

- There is no exact BT. Why?
- All computational methods for BT require solution of dual Lyapunov (or Riccati) equations, for simplicity consider

$$AX + XA^T + BB^T = 0, \quad A \in \mathbb{R}^{n \times n},\ B \in \mathbb{R}^{n \times m}.$$

Approximate BT

- There is no exact BT. Why?
- All computational methods for BT require solution of dual Lyapunov (or Riccati) equations, for simplicity consider

$$AX + XA^T + BB^T = 0, \quad A \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times m}.$$

- There is no direct or numerically backward stable method with complexity $\leq \mathcal{O}(n^3)$ to solve Lyapunov equations!

Approximate BT

- There is no exact BT. Why?
- All computational methods for BT require solution of dual Lyapunov (or Riccati) equations, for simplicity consider

$$AX + XA^T + BB^T = 0, \quad A \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times m}.$$

- There is no direct or numerically backward stable method with complexity $\leq \mathcal{O}(n^3)$ to solve Lyapunov equations!
  Bartels-Stewart/Hammerling algorithms are considered to be numerically backward stable.

  This is only true for triangular A: otherwise, the QR algorithm is used to triangularize A, but this algorithm solves an eigenvalue problem that may be ill-conditioned even if the solution of the Lyapunov equation is well-conditioned!!

  Also note: the QR algorithm is iterative whenever $n > 4$!

  Sign function solvers for Lyapunov equations may be more accurate than Bartels-Stewart/Hammarling, even though they are not numerically stable!

Approximate BT

- There is no exact BT. Why?
- All computational methods for BT require solution of dual Lyapunov (or Riccati) equations, for simplicity consider

$$AX + XA^T + BB^T = 0, \quad A \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times m}.$$

- There is no direct or numerically backward stable method with complexity $\leq \mathcal{O}(n^3)$ to solve Lyapunov equations!
- Current solvers for large-scale, sparse Lyapunov equations (ADI, cyclic Smith, K-PIK; complexity $\mathcal{O}(m \cdot nnz)$) may or may not compute a solution that is as accurate as solutions obtained with $\mathcal{O}(n^3)$ solver.

Structure Preservation

RLC network equations

System structure often encountered in circuit simulation, e.g., in RC(L) networks w/o voltage sources:

$$E = \begin{bmatrix} E_1 & 0 \\ 0 & E_2 \end{bmatrix}, \quad A = \begin{bmatrix} -A_1 & -A_2^T \\ A_2 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix} = C^T,$$

where $A_1, E_1 \geq 0$, $E_2 > 0$.

Note: $G(s)$ symmetric, multiplication of 2nd block row by -1 yields $E = E^T$, $A = A^T$

$\Rightarrow$ *Gramians coincide, $P = Q$*

$\Rightarrow$ *BT needs only one Lyapunov equation, $W \equiv V$*

$\Rightarrow$ *BT preserves stability and passivity.*

Structure Preservation

Split-congruence BT (scBT)

[KERNS/YANG '98]: split-congruence transformations

$$(\hat{E}, \hat{A}, \hat{B}) = (\mathcal{V}^T E \mathcal{V}, \mathcal{V}^T A \mathcal{V}, \mathcal{V}^T B), \text{ where } \mathcal{V} = \begin{bmatrix} V_1 & \\ & V_2 \end{bmatrix}, \qquad (3)$$

preserve stability, passivity, and reciprocity, i.e., reduced-order transfer function has the form

$$\hat{G}(s) = \hat{B}_1^T (s\hat{E}_1 + \hat{A}_1 + \frac{1}{s} \hat{A}_2^T \hat{E}_2^{-1} \hat{A}_2) \hat{B}_1,$$

cf. SPRIM papers [FREUND '04/'06].

Structure Preservation

Split-congruence BT (scBT)

[KERNS/YANG '98]: split-congruence transformations

$$(\hat{E}, \hat{A}, \hat{B}) = (\mathcal{V}^T E \mathcal{V}, \mathcal{V}^T A \mathcal{V}, \mathcal{V}^T B), \text{ where } \mathcal{V} = \begin{bmatrix} V_1 & \\ & V_2 \end{bmatrix}, \qquad (3)$$

preserve stability, passivity, and reciprocity, i.e., reduced-order transfer function has the form

$$\hat{G}(s) = \hat{B}_1^T (s\hat{E}_1 + \hat{A}_1 + \frac{1}{s}\hat{A}_2^T \hat{E}_2^{-1} \hat{A}_2) \hat{B}_1,$$

cf. SPRIM papers [FREUND '04/'06].

Reciprocity preserved ⇝ reduced-order model can be synthesized as circuit (e.g., [REIS '08]).

Structure Preservation

Split-congruence BT (scBT)

[KERNS/YANG '98]: split-congruence transformations

$$(\hat{E}, \hat{A}, \hat{B}) = (\mathcal{V}^T E \mathcal{V}, \mathcal{V}^T A \mathcal{V}, \mathcal{V}^T B), \text{ where } \mathcal{V} = \begin{bmatrix} V_1 & \\ & V_2 \end{bmatrix}, \qquad (3)$$

preserve stability, passivity, and reciprocity, i.e., reduced-order transfer function has the form

$$\hat{G}(s) = \hat{B}_1^T (s\hat{E}_1 + \hat{A}_1 + \frac{1}{s}\hat{A}_2^T \hat{E}_2^{-1} \hat{A}_2) \hat{B}_1,$$

cf. SPRIM papers [FREUND '04/'06].

(Very) basic idea: let $V = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \in \mathbb{R}^{n \times r}$ be projection matrix computed by BT, then use V_1, V_2 as in (3).

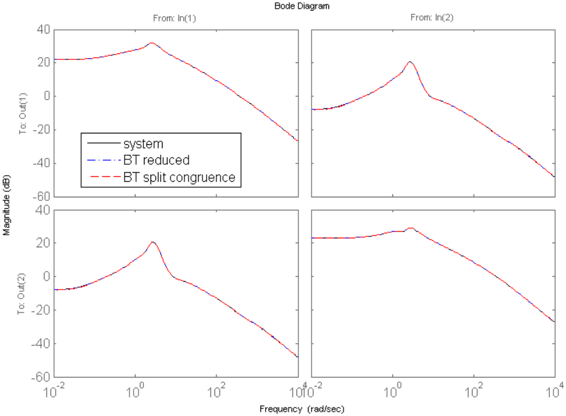
Note: $\operatorname{range}(V) \subset \operatorname{range}(\mathcal{V})$.

Note: theoretical properties of scBT not clear yet.

Structure Preservation

Numerical example

- Random system, $n = 150, m = 2$
- reduced-order, tolerance 10^{-2} $\rightsquigarrow$ $r = 34,\ \delta = 8.6 \cdot 10^{-3}$.



Note: larger error for $\omega \to 0$, error bound does not hold!

Sparsity of Reduced-Order Systems

- BT is often criticized for producing dense reduced-order models.
- Note: this is also true for almost all recent moment-matching methods, e.g. PRIMA, rational interpolation/Krylov, SPRIM.
- Mostly, reduced-order models are used when solving linear systems of equations
  - $(\jmath\omega\hat{E} - \hat{A})x = b$ in frequency-domain analysis,
  - $(\hat{E} - h_k\hat{A})x_{k+1} = \hat{E}x_k + \ldots$ in implicit integration schemes (e.g., transient analysis).

The cost for solving the linear systems may not benefit from smaller order, if efficient sparse direct solver for full-size sparse system matrices is available.

Sparsity of Reduced-Order Systems

- BT is often criticized for producing dense reduced-order models.
- Note: this is also true for almost all recent moment-matching methods, e.g. PRIMA, rational interpolation/Krylov, SPRIM.
- Mostly, reduced-order models are used when solving linear systems of equations
  - $(\jmath\omega\hat{E} - \hat{A})x = b$ in frequency-domain analysis,
  - $(\hat{E} - h_k\hat{A})x_{k+1} = \hat{E}x_k + \ldots$ in implicit integration schemes (e.g., transient analysis).

  The cost for solving the linear systems may not benefit from smaller order, if efficient sparse direct solver for full-size sparse system matrices is available.

Sparsity of Reduced-Order Systems

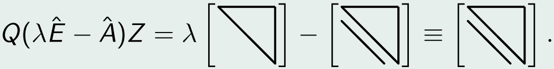
- BT is often criticized for producing dense reduced-order models.
- Note: this is also true for almost all recent moment-matching methods, e.g. PRIMA, rational interpolation/Krylov, SPRIM.
- Mostly, reduced-order models are used when solving linear systems of equations
  - $(\jmath\omega\hat{E} - \hat{A})x = b$ in frequency-domain analysis,
  - $(\hat{E} - h_k\hat{A})x_{k+1} = \hat{E}x_k + \ldots$ in implicit integration schemes (e.g., transient analysis).

The cost for solving the linear systems may not benefit from smaller order, if efficient sparse direct solver for full-size sparse system matrices is available.

Sparsity of Reduced-Order Systems

An easy improvement

Significant reduction can be achieved by transforming $(\hat{A}, \hat{E})$ to Hessenberg-triangular form using QZ algorithm, i.e., compute orthogonal Q, Z such that



$$Q(\lambda \hat{E} - \hat{A})Z = \lambda \qquad - \qquad \equiv \qquad .$$

New reduced-order system: $(Q\hat{E}Z, Q\hat{A}Z, Q\hat{B}, \hat{C}Z)$, linear systems of equations

$$\begin{array}{rcl} (\jmath\omega \hat{E} - \hat{A})x & = & b, \\ (\hat{E} - h_k \hat{A})x_{k+1} & = & \hat{E}x_k + \ldots, \quad \text{etc.} \end{array}$$

have Hessenberg form and can thus be solved using $r-1$ Givens rotations only! (Needs Hessenberg solver inside simulator.)

For symmetric systems, further reduction can be achieved.

Systems with a Large Number of Terminals

$m, p = \mathcal{O}(n)$

- Efficient BT implementations are based on assumption $n \gg m, p$.
- For on-chip clock distribution networks, power grids, wide buses, this assumption is not justified; here, $m, p = \mathcal{O}(n)$, e.g., $m = p = \frac{n}{2}, \frac{n}{4}$.
- Cure: BT can easily be combined with SVDMOR [Feldmann/Liu '04]: for $G(s) = C(sE - A)^{-1}B$, let

$$\begin{aligned} G(s_0) &= C(s_0E - A)^{-1}B = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \\ &\approx U_1\Sigma_1V_1^T \quad \text{(rank-}k\text{ approximation)}, \end{aligned}$$

so that $\|G(s_0) - U_1\Sigma_1V_1^T\|_2 = \sigma_{k+1}$.

Now define $\tilde{B} := BV_1$, $\tilde{C} := U_1^TC$, then

$$G(s) \approx U_1\underbrace{\tilde{B}(sE - A)^{-1}\tilde{C}}_{=:\tilde{G}(s)}V_1^T,$$

and apply BT to $\tilde{G}(s) \rightsquigarrow \hat{G}(s)$: $G(s) \approx U_1\hat{G}(s)V_1^T$.

Systems with a Large Number of Terminals

$m, p = \mathcal{O}(n)$

- Efficient BT implementations are based on assumption $n \gg m, p$.
- For on-chip clock distribution networks, power grids, wide buses, this assumption is not justified; here, $m, p = \mathcal{O}(n)$, e.g., $m = p = \frac{n}{2}, \frac{n}{4}$.
- Cure: BT can easily be combined with SVDMOR [Feldmann/Liu '04]: for $G(s) = C(sE - A)^{-1}B$, let

$$\begin{aligned} G(s_0) &= C(s_0E - A)^{-1}B = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \\ &\approx U_1 \Sigma_1 V_1^T \quad \text{(rank-}k\text{ approximation)}, \end{aligned}$$

so that $\| G(s_0) - U_1 \Sigma_1 V_1^T \|_2 = \sigma_{k+1}$.

Now define $\tilde{B} := BV_1$, $\tilde{C} := U_1^T C$, then

$$G(s) \approx U_1 \underbrace{\tilde{B}(sE - A)^{-1}\tilde{C}}_{=:\tilde{G}(s)} V_1^T,$$

and apply BT to $\tilde{G}(s) \rightsquigarrow \hat{G}(s)$: $G(s) \approx U_1 \hat{G}(s) V_1^T$.

Systems with a Large Number of Terminals

$m,p=\mathcal{O}(n)$

- Efficient BT implementations are based on assumption $n \gg m,p$.
- For on-chip clock distribution networks, power grids, wide buses, this assumption is not justified; here, $m,p=\mathcal{O}(n)$, e.g., $m=p=\frac{n}{2},\frac{n}{4}$.
- Cure: BT can easily be combined with SVDMOR [Feldmann/Liu '04]: for $G(s)=C(sE-A)^{-1}B$, let

$$\begin{aligned} G(s_0) &= C(s_0E-A)^{-1}B = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \\ &\approx U_1\Sigma_1V_1^T \quad \text{(rank-}k\text{ approximation)}, \end{aligned}$$

so that $\|G(s_0)-U_1\Sigma_1V_1^T\|_2=\sigma_{k+1}$.

Now define $\tilde{B}:=BV_1$, $\tilde{C}:=U_1^TC$, then

$$G(s)\approx U_1\underbrace{\tilde{B}(sE-A)^{-1}\tilde{C}}_{=:\tilde{G}(s)}V_1^T,$$

and apply BT to $\tilde{G}(s) \rightsquigarrow \hat{G}(s)$: $G(s)\approx U_1\hat{G}(s)V_1^T$.

Systems with a Large Number of Terminals

$m, p = \mathcal{O}(n)$

- Efficient BT implementations are based on assumption $n \gg m, p$.
- For on-chip clock distribution networks, power grids, wide buses, this assumption is not justified; here, $m, p = \mathcal{O}(n)$, e.g., $m = p = \frac{n}{2}, \frac{n}{4}$.
- Cure: BT can easily be combined with SVDMOR [Feldmann/Liu '04]: for $G(s) = C(sE - A)^{-1}B$, let

$$\begin{aligned} G(s_0) &= C(s_0E - A)^{-1}B = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \\ &\approx U_1 \Sigma_1 V_1^T \quad \text{(rank-}k\text{ approximation)}, \end{aligned}$$

so that $\|G(s_0) - U_1\Sigma_1 V_1^T\|_2 = \sigma_{k+1}$.
Now define $\tilde{B} := BV_1, \quad \tilde{C} := U_1^T C,$ then

$$G(s) \approx U_1 \underbrace{\tilde{B}(sE - A)^{-1}\tilde{C}}_{=:\tilde{G}(s)} V_1^T,$$

and apply BT to $\tilde{G}(s) \rightsquigarrow \hat{G}(s)$: $G(s) \approx U_1 \hat{G}(s) V_1^T$.

Systems with a Large Number of Terminals

$m, p = \mathcal{O}(n)$

- Efficient BT implementations are based on assumption $n \gg m, p$.
- For on-chip clock distribution networks, power grids, wide buses, this assumption is not justified; here, $m, p = \mathcal{O}(n)$, e.g., $m = p = \frac{n}{2}, \frac{n}{4}$.
- Cure: BT can easily be combined with SVDMOR [Feldmann/Liu '04]: for $G(s) = C(sE - A)^{-1}B$, let

$$\begin{aligned} G(s_0) &= C(s_0E - A)^{-1}B = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \\ &\approx U_1 \Sigma_1 V_1^T \quad \text{(rank-}k\text{ approximation)}, \end{aligned}$$

so that $\|G(s_0) - U_1\Sigma_1 V_1^T\|_2 = \sigma_{k+1}$.
Now define $\tilde{B} := BV_1, \quad \tilde{C} := U_1^T C,$ then

$$G(s) \approx U_1 \underbrace{\tilde{B}(sE - A)^{-1}\tilde{C}}_{=:\tilde{G}(s)} V_1^T,$$

and apply BT to $\tilde{G}(s) \rightsquigarrow \hat{G}(s)$: $G(s) \approx U_1 \hat{G}(s) V_1^T$.

Use truncated SVD → cf. Oral CS 1C (A. Schneider), Mon 16:00h.

Systems with a Large Number of Terminals

$m, p = \mathcal{O}(n)$

- Efficient BT implementations are based on assumption $n \gg m, p$.
- For on-chip clock distribution networks, power grids, wide buses, this assumption is not justified; here, $m, p = \mathcal{O}(n)$, e.g., $m = p = \frac{n}{2}, \frac{n}{4}$.
- Cure: BT can easily be combined with SVDMOR [Feldmann/Liu '04]: for $G(s) = C(sE - A)^{-1}B$, let

$$\begin{aligned} G(s_0) &= C(s_0E - A)^{-1}B = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \\ &\approx U_1 \Sigma_1 V_1^T \quad \text{(rank-}k\text{ approximation)}, \end{aligned}$$

so that $\|G(s_0) - U_1 \Sigma_1 V_1^T\|_2 = \sigma_{k+1}$.
Now define $\tilde{B} := BV_1, \quad \tilde{C} := U_1^T C,$ then

$$G(s) \approx U_1 \underbrace{\tilde{B}(sE - A)^{-1}\tilde{C}}_{=:\tilde{G}(s)} V_1^T,$$

and apply BT to $\tilde{G}(s) \rightsquigarrow \hat{G}(s)$: $G(s) \approx U_1 \hat{G}(s) V_1^T$.

Alternative for medium-size m: superposition of reduced-order SIMO models using Padé-type approximation [Feng/B./Rudnyi '08].

Conclusions

- BT preferred MOR technique in control theory, so far less popular in circuit simulation.
- Limitations of balancing-related model reduction methods w.r.t. descriptor systems, large-scale systems, unstable systems fade away bit by bit.
- Viable alternative to moment-matching/Padé approximation/rational interpolation methods in many situations; computational complexity is usually higher, but in the same complexity class $\mathcal{O}(nnz \times r)$.
- Modern implementations of BT are essentially of the same computational complexity as approximations like frequency-domain POD [WILLCOX/PERAIRE '02] (aka Poor Man's TBR [PHILLIPS/SILVEIRA '04/'05] $\sim$ rational interpolation [GRIMME '97,...]), but are closer to satisfy theoretical properties of BT.
- Split-congruence BT preserves reciprocity; thus, allows circuit synthesis approach of [REIS '08] to derive MNA equations/netlist.
- Reduced-order models can be made more sparse to allow faster simulation (if integrator is adapted).

Conclusions

Software

MATLAB:

- Lyapack/M.E.S.S. (Matrix Equations Sparse Solvers),
- MORLAB (dense, pre-β . . .)

F77/C:

- PLiCMR (dense),
- SpaRed (sparse).

Available from

http://www.tu-chemnitz.de/mathematik/industrie_technik/software

More to come . . .

SyreNe
System Reduction for Nanoscale IC Design

MOORE
NICE !

References

1 A.C. Antoulas.
Lectures on the Approximation of Large-Scale Dynamical Systems.
SIAM Publications, Philadelphia, PA, 2005.

2 P. Benner, E.S. Quintana-Ortí, and G. Quintana-Ortí.
State-space truncation methods for parallel model reduction of large-scale systems.
Parallel Comput., 29:1701–1722, 2003.

3 P. Benner.
Numerical linear algebra for model reduction in control and simulation.
GAMM Mitt., 29(2):275–296, 2006.

4 P. Benner, R. Freund, D. Sorensen, and A. Varga (editors).
Special issue on *Order Reduction of Large-Scale Systems*.
Linear Algebra Appl., Vol. 415, June 2006.

5 P. Benner, V. Mehrmann, and D. Sorensen (editors).
Dimension Reduction of Large-Scale Systems.
Lecture Notes in Computational Science and Engineering, Vol. 45,
Springer-Verlag, Berlin/Heidelberg, Germany, 2005.

6 M. Günther, U. Feldmann, and E.J.W. ter Maten.
Modelling and Discretization of Circuit Problems.
In W.H.A. Schilders and E.J.W. ter Maten (editors), Numerical Methods in Electromagnetics, vol. 15 of *Handbook of Numerical Analysis*, pp. 523–659, Elsevier, 2005.

7 G. Obinata and B.D.O. Anderson.
Model Reduction for Control System Design.
Springer-Verlag, London, UK, 2001.

8 W.H.A. Schilders, H.A. van der Vorst, and J. Rommes (editors).
Model Order Reduction: Theory, Research Aspects and Applications.
Springer-Verlag, Berlin, Heidelberg, 2008.

9 S.X.-D. Tan and L. He.
Advanced Model Order Reduction Techniques in VLSI Design.
Cambridge University Press, New York, 2007.