Model Reduction for Complex Dynamical Systems
December 2, 2010

Nonlinear Model Reduction via Quadratic Bilinear Control Systems

Peter Benner Tobias Breiten

Max Planck Institute for Dynamics of Complex Technical Systems
Computational Methods in Systems and Control Theory

NETWORK THEORY

MAX PLANCK INSTITUTE FOR DYNAMICS OF COMPLEX TECHNICAL SYSTEMS MAGDEBURG

Outline

Motivation

Given a large-scale state-nonlinear control system of the form

$$\boxed{\Sigma : \begin{cases} \dot{x}(t) = f(x(t)) + bu(t), \\ y(t) = cx(t), \quad x(0) = x_0, \end{cases}}$$

with $f : \mathbb{R}^n \to \mathbb{R}^n$ nonlinear and $b, c^T \in \mathbb{R}^n$, $x \in \mathbb{R}^n$, $u, y \in \mathbb{R}$.

Motivation

Given a large-scale state-nonlinear control system of the form

$$\boxed{\Sigma : \begin{cases} \dot{x}(t) = f(x(t)) + bu(t), \\ y(t) = cx(t), \quad x(0) = x_0, \end{cases}}$$

with $f : \mathbb{R}^n \to \mathbb{R}^n$ nonlinear and $b, c^T \in \mathbb{R}^n$, $x \in \mathbb{R}^n$, $u, y \in \mathbb{R}$.

- Optimization, control and simulation cannot be done efficiently!

Motivation

Given a large-scale state-nonlinear control system of the form

$$\Sigma : \begin{cases} \dot{x}(t) = f(x(t)) + bu(t), \\ y(t) = cx(t), \quad x(0) = x_0, \end{cases}$$

with $f : \mathbb{R}^n \to \mathbb{R}^n$ nonlinear and $b, c^T \in \mathbb{R}^n$, $x \in \mathbb{R}^n$, $u, y \in \mathbb{R}$.

- Optimization, control and simulation cannot be done efficiently!

MOR

$$\hat{\Sigma} : \begin{cases} \dot{\hat{x}}(t) = \hat{f}(\hat{x}(t)) + \hat{b}u(t), \\ \hat{y}(t) = \hat{c}\hat{x}(t), \quad \hat{x}(0) = \hat{x}_0, \end{cases}$$

with $\hat{f} : \mathbb{R}^{\hat{n}} \to \mathbb{R}^{\hat{n}}$ and $\hat{b}, \hat{c}^T \in \mathbb{R}^{\hat{n}}$, $x \in \mathbb{R}^{\hat{n}}$, $u \in \mathbb{R}$ and

Motivation

Given a large-scale state-nonlinear control system of the form

$$\Sigma : \begin{cases} \dot{x}(t) = f(x(t)) + bu(t), \\ y(t) = cx(t), \quad x(0) = x_0, \end{cases}$$

with $f : \mathbb{R}^n \to \mathbb{R}^n$ nonlinear and $b, c^T \in \mathbb{R}^n$, $x \in \mathbb{R}^n$, $u, y \in \mathbb{R}$.

- Optimization, control and simulation cannot be done efficiently!

MOR

$$\hat{\Sigma} : \begin{cases} \dot{\hat{x}}(t) = \hat{f}(\hat{x}(t)) + \hat{b}u(t), \\ \hat{y}(t) = \hat{c}\hat{x}(t), \quad \hat{x}(0) = \hat{x}_0, \end{cases}$$

with $\hat{f} : \mathbb{R}^{\hat{n}} \to \mathbb{R}^{\hat{n}}$ and $\hat{b}, \hat{c}^T \in \mathbb{R}^{\hat{n}}$, $x \in \mathbb{R}^{\hat{n}}$, $u \in \mathbb{R}$ and $\hat{y} \approx y \in \mathbb{R}$, $\hat{n} \ll n$.

Common Reduction Techniques

Proper Orthogonal Decomposition (POD)

- Take computed or experimental 'snapshots' of full model: $[x(t_1), x(t_2), \ldots, x(t_N)] =: X$,
- perform SVD of snapshot matrix: $X = VSW^T \approx V_{\hat{n}} S_{\hat{n}} W_{\hat{n}}^T$.
- Reduction by POD-Galerkin projection: $\dot{\hat{x}} = V_{\hat{n}}^T f(V_{\hat{n}} \hat{x}) + V_{\hat{n}}^T Bu$.
- Requires evaluation of f
  $\rightsquigarrow$ discrete empirical interpolation [Sorensen/Chaturantabut '09].
- Input dependency due to 'snapshots'!

Common Reduction Techniques

Proper Orthogonal Decomposition (POD)

- Take computed or experimental 'snapshots' of full model: $[x(t_1), x(t_2), \ldots, x(t_N)] =: X$,
- perform SVD of snapshot matrix: $X = VSW^T \approx V_{\hat{n}} S_{\hat{n}} W_{\hat{n}}^T$.
- Reduction by POD-Galerkin projection: $\dot{\hat{x}} = V_{\hat{n}}^T f(V_{\hat{n}} \hat{x}) + V_{\hat{n}}^T Bu$.
- Requires evaluation of f
  $\rightsquigarrow$ discrete empirical interpolation [Sorensen/Chaturantabut '09].
- Input dependency due to 'snapshots'!

Trajectory Piecewise Linear (TPWL)

- Linearize f along trajectory,
- reduce resulting linear systems,
- construct reduced model by weighting sum of linear systems.
- Requires simulation of original model and several linear reduction steps, many heuristics.

State-Space Representation of QBDAEs

We will consider quadratic-bilinear SISO systems of the form

$$E\,\dot{x} = A_1\,x + A_2\,x \otimes x + N\,xu + b\,u$$
$$y = c\,x$$

where $E, A_1, N \in \mathbb{R}^{n\times n}$, $A_2 \in \mathbb{R}^{n\times n^2}$ (Hessian tensor), $b, c^T \in \mathbb{R}^n$.

- A large class of nonlinear control-affine systems can be transformed into the above type of control system.
- The transformation is exact, but a slight increase of the state dimension has to be accepted.
- Input-output behavior can be characterized by generalized transfer functions $\rightsquigarrow$ enables us to use Krylov-based reduction techniques.

Transformation via McCormick Relaxation

Theorem [Gu'09]

Assume that the state equation of a nonlinear system Σ is given by

$$\dot{x} = a_0 x + a_1 g_1(x) + \ldots + a_k g_k(x) + Bu,$$

where $g_i(x) : \mathbb{R}^n \to \mathbb{R}^n$ are compositions of uni-variable rational, exponential, logarithmic, trigonometric or root functions, respectively. Then, by iteratively taking derivatives and adding algebraic equations, respectively, Σ can be transformed into a system of quadratic-bilinear DAEs of dimension $N > n$.

Transformation via McCormick Relaxation

Theorem [Gu'09]

Assume that the state equation of a nonlinear system Σ is given by

$$\dot{x} = a_0 x + a_1 g_1(x) + \ldots + a_k g_k(x) + Bu,$$

where $g_i(x) : \mathbb{R}^n \to \mathbb{R}^n$ are compositions of uni-variable rational, exponential, logarithmic, trigonometric or root functions, respectively. Then, by iteratively taking derivatives and adding algebraic equations, respectively, Σ can be transformed into a system of quadratic-bilinear DAEs of dimension $N > n$.

Example

- $\dot{x}_1 = \exp(-x_2) \cdot \sqrt{x_1^2 + 1}, \quad \dot{x}_2 = -x_2 + u.$

Transformation via McCormick Relaxation

Theorem [Gu'09]

Assume that the state equation of a nonlinear system Σ is given by

$$\dot{x} = a_0 x + a_1 g_1(x) + \ldots + a_k g_k(x) + Bu,$$

where $g_i(x) : \mathbb{R}^n \to \mathbb{R}^n$ are compositions of uni-variable rational, exponential, logarithmic, trigonometric or root functions, respectively. Then, by iteratively taking derivatives and adding algebraic equations, respectively, Σ can be transformed into a system of quadratic-bilinear DAEs of dimension $N > n$.

Example

- $\dot{x}_1 = \exp(-x_2) \cdot \sqrt{x_1^2 + 1}, \quad \dot{x}_2 = -x_2 + u.$
- $z_1 := \exp(-x_2),$

Transformation via McCormick Relaxation

Theorem [Gu'09]

Assume that the state equation of a nonlinear system Σ is given by

$$\dot{x} = a_0 x + a_1 g_1(x) + \ldots + a_k g_k(x) + Bu,$$

where $g_i(x) : \mathbb{R}^n \to \mathbb{R}^n$ are compositions of uni-variable rational, exponential, logarithmic, trigonometric or root functions, respectively. Then, by iteratively taking derivatives and adding algebraic equations, respectively, Σ can be transformed into a system of quadratic-bilinear DAEs of dimension $N > n$.

Example

- $\dot{x}_1 = \exp(-x_2) \cdot \sqrt{x_1^2 + 1}, \quad \dot{x}_2 = -x_2 + u.$
- $z_1 := \exp(-x_2), \quad z_2 := \sqrt{x_1^2 + 1}.$

Transformation via McCormick Relaxation

Theorem [Gu'09]

Assume that the state equation of a nonlinear system Σ is given by

$$\dot{x} = a_0 x + a_1 g_1(x) + \ldots + a_k g_k(x) + Bu,$$

where $g_i(x) : \mathbb{R}^n \to \mathbb{R}^n$ are compositions of uni-variable rational, exponential, logarithmic, trigonometric or root functions, respectively. Then, by iteratively taking derivatives and adding algebraic equations, respectively, Σ can be transformed into a system of quadratic-bilinear DAEs of dimension $N > n$.

Example

- $\dot{x}_1 = \exp(-x_2) \cdot \sqrt{x_1^2 + 1}, \quad \dot{x}_2 = -x_2 + u.$
- $z_1 := \exp(-x_2), \quad z_2 := \sqrt{x_1^2 + 1}.$
- $\dot{x}_1 = z_1 \cdot z_2,$

Transformation via McCormick Relaxation

Theorem [Gu'09]

Assume that the state equation of a nonlinear system Σ is given by

$$\dot{x} = a_0 x + a_1 g_1(x) + \ldots + a_k g_k(x) + Bu,$$

where $g_i(x) : \mathbb{R}^n \to \mathbb{R}^n$ are compositions of uni-variable rational, exponential, logarithmic, trigonometric or root functions, respectively. Then, by iteratively taking derivatives and adding algebraic equations, respectively, Σ can be transformed into a system of quadratic-bilinear DAEs of dimension $N > n$.

Example

- $\dot{x}_1 = \exp(-x_2) \cdot \sqrt{x_1^2 + 1}, \quad \dot{x}_2 = -x_2 + u.$
- $z_1 := \exp(-x_2), \quad z_2 := \sqrt{x_1^2 + 1}.$
- $\dot{x}_1 = z_1 \cdot z_2, \quad \dot{x}_2 = -x_2 + u,$

Transformation via McCormick Relaxation

Theorem [Gu'09]

Assume that the state equation of a nonlinear system Σ is given by

$$\dot{x} = a_0 x + a_1 g_1(x) + \ldots + a_k g_k(x) + Bu,$$

where $g_i(x) : \mathbb{R}^n \to \mathbb{R}^n$ are compositions of uni-variable rational, exponential, logarithmic, trigonometric or root functions, respectively. Then, by iteratively taking derivatives and adding algebraic equations, respectively, Σ can be transformed into a system of quadratic-bilinear DAEs of dimension $N > n$.

Example

- $\dot{x}_1 = \exp(-x_2) \cdot \sqrt{x_1^2 + 1}, \quad \dot{x}_2 = -x_2 + u.$
- $z_1 := \exp(-x_2), \quad z_2 := \sqrt{x_1^2 + 1}.$
- $\dot{x}_1 = z_1 \cdot z_2, \quad \dot{x}_2 = -x_2 + u, \quad \dot{z}_1 = -z_1 \cdot (-x_2 + u),$

Transformation via McCormick Relaxation

Theorem [Gu'09]

Assume that the state equation of a nonlinear system Σ is given by

$$\dot{x} = a_0 x + a_1 g_1(x) + \ldots + a_k g_k(x) + Bu,$$

where $g_i(x) : \mathbb{R}^n \to \mathbb{R}^n$ are compositions of uni-variable rational, exponential, logarithmic, trigonometric or root functions, respectively. Then, by iteratively taking derivatives and adding algebraic equations, respectively, Σ can be transformed into a system of quadratic-bilinear DAEs of dimension $N > n$.

Example

- $\dot{x}_1 = \exp(-x_2) \cdot \sqrt{x_1^2 + 1}, \quad \dot{x}_2 = -x_2 + u.$
- $z_1 := \exp(-x_2), \quad z_2 := \sqrt{x_1^2 + 1}.$
- $\dot{x}_1 = z_1 \cdot z_2, \quad \dot{x}_2 = -x_2 + u, \quad \dot{z}_1 = -z_1 \cdot (-x_2 + u),$
  $\dot{z}_2 = \frac{2 \cdot x_1 \cdot z_1 \cdot z_2}{2 \cdot z_2} = x_1 \cdot z_1.$

Variational Analysis and Linear Subsystems

Analysis of nonlinear systems by variational equation approach:

Variational Analysis and Linear Subsystems

Analysis of nonlinear systems by variational equation approach:

- consider input of the form $\alpha u(t)$,

Variational Analysis and Linear Subsystems

Analysis of nonlinear systems by variational equation approach:

- consider input of the form $\alpha u(t)$,
- nonlinear system is assumed to be a series of homogeneous nonlinear subsystems, i.e. response should be of the form

$$x(t) = \alpha x_1(t) + \alpha^2 x_2(t) + \alpha^3 x_3(t) + \dots.$$

Variational Analysis and Linear Subsystems

Analysis of nonlinear systems by variational equation approach:

- consider input of the form $\alpha u(t)$,
- nonlinear system is assumed to be a series of homogeneous nonlinear subsystems, i.e. response should be of the form

$$x(t) = \alpha x_1(t) + \alpha^2 x_2(t) + \alpha^3 x_3(t) + \dots.$$

- comparison of terms $\alpha^i, i = 1, 2, \dots$ leads to series of systems

$$\begin{aligned} E\dot{x}_1 &= A_1 x_1 + Bu, \\ E\dot{x}_2 &= A_1 x_2 + A_2 x_1 \otimes x_1 + N x_1 u, \\ E\dot{x}_3 &= A_1 x_3 + A_2 \left(x_1 \otimes x_2 + x_2 \otimes x_1\right) + N x_2 u \\ &\vdots \end{aligned}$$

Variational Analysis and Linear Subsystems

Analysis of nonlinear systems by variational equation approach:

- consider input of the form $\alpha u(t)$,
- nonlinear system is assumed to be a series of homogeneous nonlinear subsystems, i.e. response should be of the form

$$x(t) = \alpha x_1(t) + \alpha^2 x_2(t) + \alpha^3 x_3(t) + \dots.$$

- comparison of terms $\alpha^i, i = 1, 2, \dots$ leads to series of systems

$$\begin{aligned} E\dot{x}_1 &= A_1 x_1 + Bu, \\ E\dot{x}_2 &= A_1 x_2 + A_2 x_1 \otimes x_1 + N x_1 u, \\ E\dot{x}_3 &= A_1 x_3 + A_2 \left(x_1 \otimes x_2 + x_2 \otimes x_1\right) + N x_2 u \\ &\vdots \end{aligned}$$

- although i-th subsystem is coupled nonlinearly to preceding systems, linear systems are obtained if terms x_j, $j < i$, are interpreted as pseudo-inputs.

Generalized Transfer Functions

In a similar way, a series of generalized symmetric transfer functions can be obtained via the growing exponential approach:

Generalized Transfer Functions

In a similar way, a series of generalized symmetric transfer functions can be obtained via the growing exponential approach:

$$H_1(s_1) = C\underbrace{(s_1E - A_1)^{-1}B}_{G_1(s_1)},$$

Generalized Transfer Functions

In a similar way, a series of generalized symmetric transfer functions can be obtained via the growing exponential approach:

$$H_1(s_1) = C\,\underbrace{(s_1E - A_1)^{-1}B}_{G_1(s_1)},$$

$$H_2(s_1, s_2) = \frac{1}{2!}C\left((s_1 + s_2)E - A_1\right)^{-1}\left[N\left(G_1(s_1) + G_1(s_2)\right)\right.$$
$$\left. + A_2\left(G_1(s_1) \otimes G_1(s_2) + G_1(s_2) \otimes G_1(s_1)\right)\right],$$

Generalized Transfer Functions

In a similar way, a series of generalized symmetric transfer functions can be obtained via the growing exponential approach:

$$H_1(s_1) = C\underbrace{(s_1E - A_1)^{-1}B}_{G_1(s_1)},$$

$$\begin{aligned} H_2(s_1,s_2) = \frac{1}{2!}C\left((s_1+s_2)E - A_1\right)^{-1}\big[&N\left(G_1(s_1)+G_1(s_2)\right) \\ &+A_2\left(G_1(s_1)\otimes G_1(s_2) + G_1(s_2)\otimes G_1(s_1)\right)\big], \end{aligned}$$

$$\begin{aligned} H_3(s_1,s_2,s_3) = \frac{1}{3!}C\left((s_1+s_2+s_3)E - A_1\right)^{-1} \\ \Big[N\big(G_2(s_1,s_2)+G_2(s_2,s_3)+G_2(s_1,s_3)\big) \\ + A_2\big(G_1(s_1)\otimes G_2(s_2,s_3) + G_1(s_2)\otimes G_2(s_1,s_3) \\ + G_1(s_3)\otimes G_2(s_1,s_3) + G_2(s_2,s_3)\otimes G_1(s_1) \\ + G_2(s_1,s_3)\otimes G_1(s_2) + G_2(s_1,s_2)\otimes G_1(s_3)\big)\Big]. \end{aligned}$$

Characterization via Multimoments

For simplicity, focus on the first two transfer functions. For $H_1(s_1)$, choosing σ and making use of the Neumann lemma leads to

$$H_1(s_1) = \sum_{i=0}^{\infty} C \underbrace{\left((A_1 - \sigma E)^{-1} E\right)^i (A_1 - \sigma E)^{-1} B \, (s_1 - \sigma)^i}_{m^i_{s_1,\sigma}}.$$

Characterization via Multimoments

For simplicity, focus on the first two transfer functions. For $H_1(s_1)$, choosing σ and making use of the Neumann lemma leads to

$$H_1(s_1) = \sum_{i=0}^{\infty} C \underbrace{\left((A_1 - \sigma E)^{-1} E\right)^i (A_1 - \sigma E)^{-1} B \, (s_1 - \sigma)^i}_{m^i_{s_1,\sigma}}.$$

Similarly, specifying an expansion point (τ, ξ) yields

$$H_2(s_1, s_2) = \frac{1}{2} \sum_{i=0}^{\infty} C \left((A_1 - (\tau + \xi)E)^{-1} E\right)^i (A_1 - (\tau + \xi)E)^{-1} \, (s_1 + s_2 - \tau - \xi)^i \cdot$$

$$\left[A_2 \left(\sum_{j=0}^{\infty} m^j_{s_1,\tau} \otimes \sum_{k=0}^{\infty} m^k_{s_2,\xi} + \sum_{k=0}^{\infty} m^k_{s_2,\xi} \otimes \sum_{j=0}^{\infty} m^j_{s_1,\tau} \right) + N \left(\sum_{p=0}^{\infty} m^p_{s_1,\tau} + \sum_{p=0}^{\infty} m^q_{s_2,\xi} \right) \right]$$

Constructing the Projection Matrix

For derivatives around $\sigma = \tau = \xi$ up to order $q - 1$, construct the Krylov spaces:

Constructing the Projection Matrix

For derivatives around $\sigma = \tau = \xi$ up to order $q-1$, construct the Krylov spaces:

$$U = \mathcal{K}_q\left((A_1 - \sigma E)^{-1}E, (A_1 - \sigma E)^{-1}B\right)$$

Constructing the Projection Matrix

For derivatives around $\sigma = \tau = \xi$ up to order $q-1$, construct the Krylov spaces:

$$U = \mathcal{K}_q\left((A_1 - \sigma E)^{-1}E, (A_1 - \sigma E)^{-1}B\right)$$

for $i = 1 : q$

$$W_i = \mathcal{K}_{q-i+1}\left((A_1 - 2\sigma E)^{-1}E, (A_1 - 2\sigma E)^{-1}NU_i\right),$$

Constructing the Projection Matrix

For derivatives around $\sigma = \tau = \xi$ up to order $q-1$, construct the Krylov spaces:

$$
\begin{aligned}
U =& \mathcal{K}_q\left((A_1 - \sigma E)^{-1}E, (A_1 - \sigma E)^{-1}B\right)\\
&\textbf{for } i = 1 : q\\
&\quad W_i = \mathcal{K}_{q-i+1}\left((A_1 - 2\sigma E)^{-1}E, (A_1 - 2\sigma E)^{-1}NU_i\right),\\
&\qquad \textbf{for } j = 1 : \min(q-i+1, i)\\
&\qquad\quad Z_i = \mathcal{K}_{q-i-j+2}\left((A_1 - 2\sigma E)^{-1}E,\right.\\
&\qquad\qquad\qquad \left.(A_1 - 2\sigma E)^{-1}A_2(U_i \otimes U_j + U_j \otimes U_i)\right),
\end{aligned}
$$

U_i denoting the i-*th* column of U.

Constructing the Projection Matrix

For derivatives around $\sigma = \tau = \xi$ up to order $q-1$, construct the Krylov spaces:

$$
\begin{aligned}
U =& \mathcal{K}_q\left((A_1 - \sigma E)^{-1}E, (A_1 - \sigma E)^{-1}B\right)\\
&\textbf{for } i = 1 : q\\
&\quad W_i = \mathcal{K}_{q-i+1}\left((A_1 - 2\sigma E)^{-1}E, (A_1 - 2\sigma E)^{-1}NU_i\right),\\
&\qquad \textbf{for } j = 1 : \min(q-i+1, i)\\
&\qquad\quad Z_i = \mathcal{K}_{q-i-j+2}\left((A_1 - 2\sigma E)^{-1}E,\right.\\
&\qquad\qquad\qquad \left.(A_1 - 2\sigma E)^{-1}A_2(U_i \otimes U_j + U_j \otimes U_i)\right),
\end{aligned}
$$

U_i denoting the i-*th* column of U. Set $V = \mathrm{orth}([U, W, Z])$ and construct $\hat{\Sigma}$ by the Galerkin-Projection $\mathcal{P} = VV^T$:

$$\hat{A}_1 = V^T A_1 V \in \mathbb{R}^{\hat{n}\times\hat{n}}, \quad \hat{A}_2 = V^T A_2 V \otimes V \in \mathbb{R}^{\hat{n}\times\hat{n}^2},$$

$$\hat{N} = V^T N V \in \mathbb{R}^{\hat{n}\times\hat{n}}, \quad \hat{B} = V^T B \in \mathbb{R}^{\hat{n}}, \quad \hat{C}^T = V^T C \in \mathbb{R}^{\hat{n}}.$$

The FitzHugh-Nagumo System

- FitzHugh-Nagumo system modeling a neuron

[Chaturantabut, Sorensen '09]

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

with $f(v) = v(v-0.1)(1-v)$ and initial and boundary conditions

$$\begin{aligned} v(x,0) &= 0, & w(x,0) &= 0, & x &\in [0,1], \\ v_x(0,t) &= -i_0(t), & v_x(1,t) &= 0, & t &\geq 0, \end{aligned}$$

where
$\epsilon = 0.015,\ h = 0.5,\ \gamma = 2,\ g = 0.05,\ i_0(t) = 5 \cdot 10^4 t^3 \exp(-15t)$

- original state dimension $n = 2 \cdot 400$, QBDAE dimension $N = 3 \cdot 400$, reduced QBDAE dimension $r = 26$, chosen expansion point $\sigma = 1$
- 3D phase space

Jet Diffusion Flame Model

Next, let us focus on a nonlinear PDE arising in jet diffusion flame models [Galbally, Willcox '09]

$$\frac{\partial w}{\partial t} + U \cdot \nabla w - \nabla(\kappa \nabla w) + f(w) = 0, \quad (x,t) \in (0,1) \times (0,T),$$

with Arrhenius type term $f(w) = Aw(c-w)e^{-\frac{E}{d-w}}$ and constant parameters U, A, E, c, d, κ. Again define initial and boundary conditions:

$$\begin{aligned} w(x,0) &= 0, \quad x \in [0,1], \\ w(0,t) &= u(t), \quad t \geq 0, \\ w(1,t) &= 0, \quad t \geq 0, \\ w_{center} &= \begin{bmatrix} \mathbf{0} & 1 & \mathbf{0} \end{bmatrix}. \end{aligned}$$

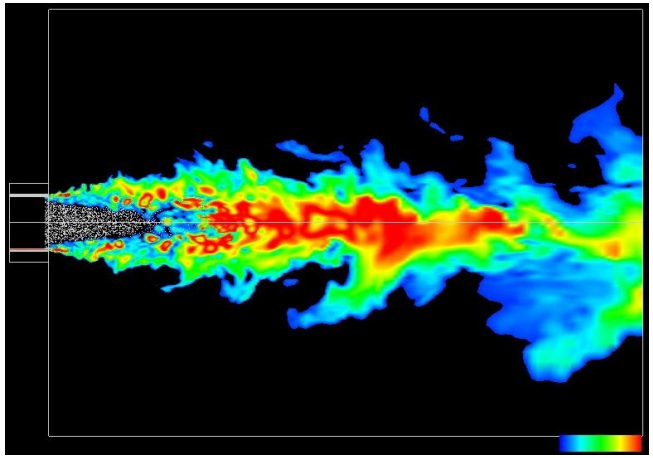
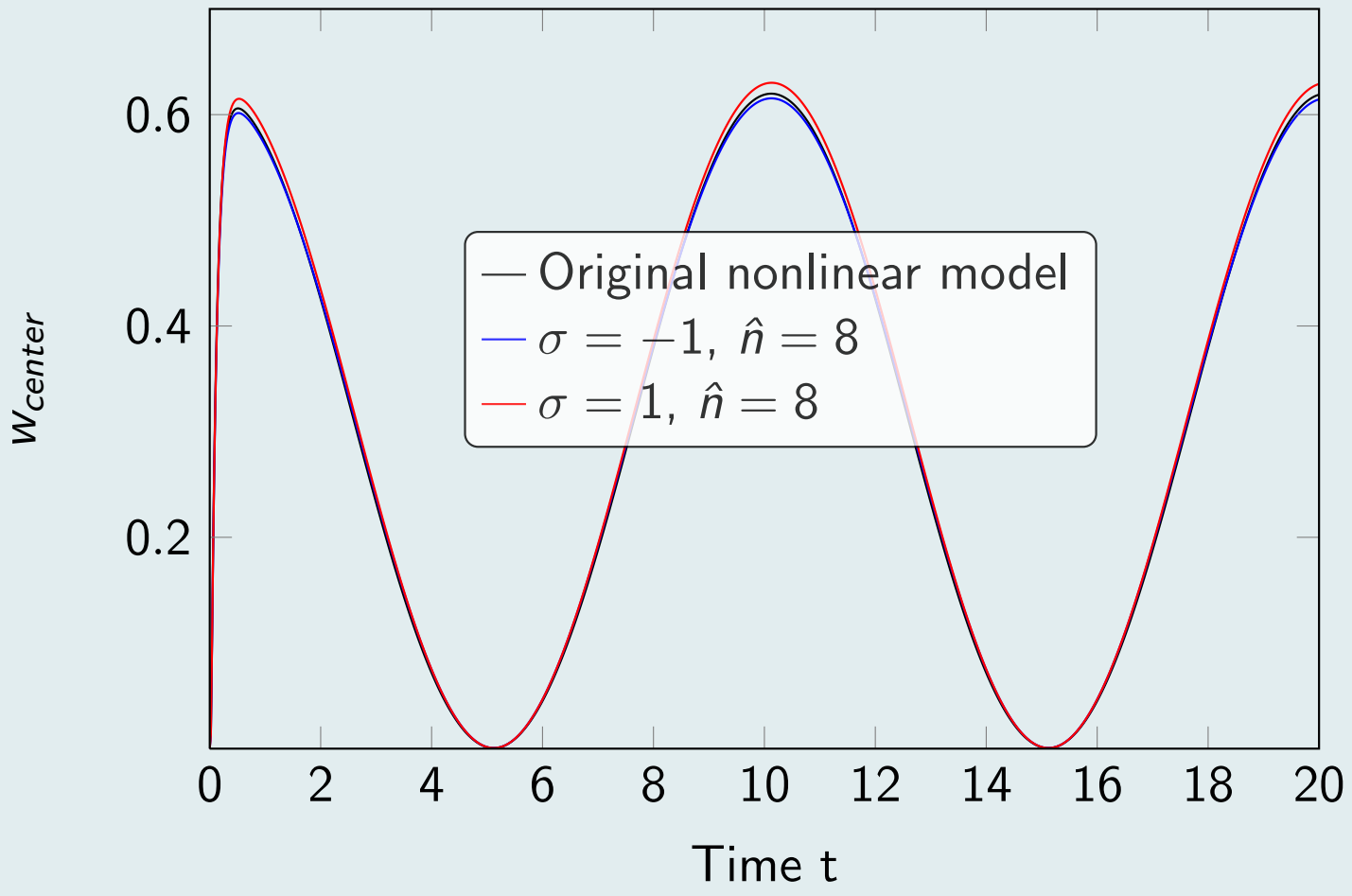


Figure: [Kurose]

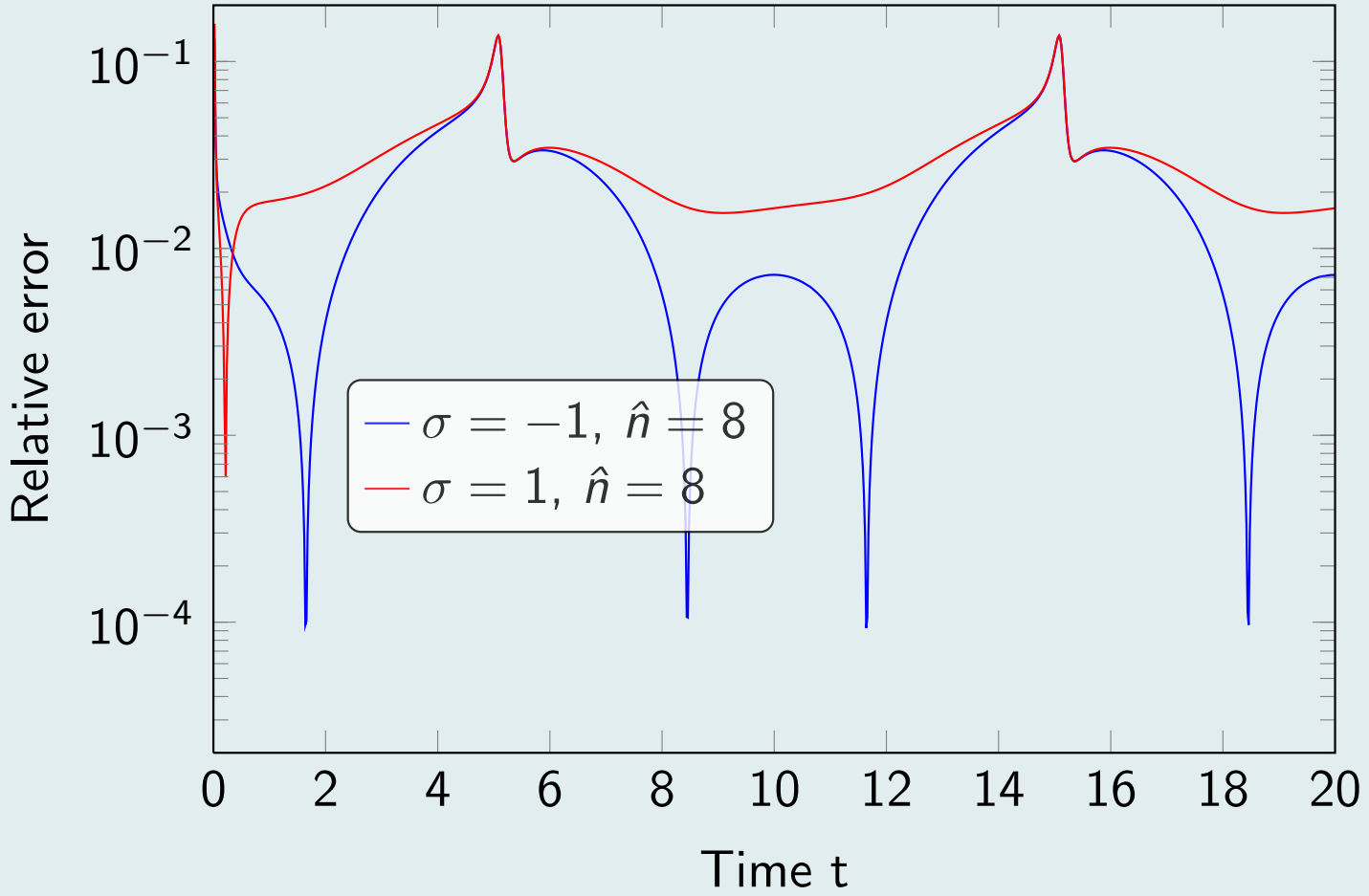
Jet Diffusion Flame Model

Transient responses for $k = 1500$ and $u(t) = \frac{1}{2}\cos(\frac{\pi t}{5} + 1)$



Jet Diffusion Flame Model

Relative errors for $k = 1500$ and $u(t) = \frac{1}{2}\cos(\frac{\pi t}{5} + 1)$



State-Space and Output Representation

Let us now focus on the special case of bilinear control systems:

$$\Sigma : \quad \begin{cases} \dot{x}(t) = Ax(t) + \sum_{i=1}^{m} N_i x(t) u(t) + Bu(t), \\ y(t) = Cx(t), \quad x(0) = x_0, \end{cases}$$

where $A, N_i \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$.

State-Space and Output Representation

Let us now focus on the special case of bilinear control systems:

$$\Sigma: \quad \begin{cases} \dot{x}(t) = Ax(t) + \sum_{i=1}^{m} N_i x(t)u(t) + Bu(t), \\ y(t) = Cx(t), \quad x(0) = x_0, \end{cases}$$

where $A, N_i \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times m},\ C \in \mathbb{R}^{p\times n}$.

Output Characterization (SISO): Volterra series

$$y(t) = \sum_{k=1}^{\infty} \int_0^t \int_0^{t_1} \dots \int_0^{t_{i-1}} h(t_1, \dots, t_k) u(t - t_1 - \dots - t_k) \cdots u(t - t_k) dt_i \cdots dt_1,$$

with kernels $h(t_1, \dots, t_k) = Ce^{At_k} N \cdots e^{At_2} N e^{At_1} B$.

State-Space and Output Representation

Let us now focus on the special case of bilinear control systems:

$$\Sigma : \quad \begin{cases} \dot{x}(t) = Ax(t) + \sum_{i=1}^{m} N_i x(t)u(t) + Bu(t), \\ y(t) = Cx(t), \quad x(0) = x_0, \end{cases}$$

where $A, N_i \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times m},\ C \in \mathbb{R}^{p\times n}$.

Output Characterization (SISO): Volterra series

$$y(t) = \sum_{k=1}^{\infty} \int_0^t \int_0^{t_1} \dots \int_0^{t_{i-1}} h(t_1, \dots, t_k) u(t - t_1 - \dots - t_k) \cdots u(t - t_k) dt_i \cdots dt_1,$$

with kernels $h(t_1, \dots, t_k) = Ce^{At_k} N \cdots e^{At_2} N e^{At_1} B$.
Multivariable Laplace-transform (SISO):

$$H_i(s_1, \dots, s_i) = C(s_i I - A)^{-1} N \cdots (s_2 I - A)^{-1} N (s_1 I - A)^{-1} B.$$

$\mathcal{H}_2$-Norm for Bilinear Systems

One possible generalization of the known linear $\mathcal{H}_2$-norm is given by:

$$||\Sigma||^2_{\mathcal{H}_2} := \operatorname{tr}\left(\sum_{k=1}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\frac{1}{(2\pi)^k}\,\overline{H_k(i\omega_1,\ldots,i\omega_k)}H_k^T(i\omega_1,\ldots,i\omega_k)\right),$$

where H_k denotes the k-th transfer function associated with the bilinear system.

$\mathcal{H}_2$-Norm for Bilinear Systems

One possible generalization of the known linear $\mathcal{H}_2$-norm is given by:

$$||\Sigma||_{\mathcal{H}_2}^2 := \operatorname{tr}\left(\sum_{k=1}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\frac{1}{(2\pi)^k}\,\overline{H_k(i\omega_1,\ldots,i\omega_k)}H_k^T(i\omega_1,\ldots,i\omega_k)\right),$$

where H_k denotes the k-th transfer function associated with the bilinear system.

It can be shown that we can alternatively compute

$$||\Sigma||_{\mathcal{H}_2}^2 = (\operatorname{vec}(I_p))^T\,(C\otimes C)\left(-A\otimes I - I\otimes A - \sum_{k=1}^{m} N_k\otimes N_k\right)^{-1}(B\otimes B)\operatorname{vec}(I_m).$$

$\mathcal{H}_2$-Norm for Bilinear Systems

One possible generalization of the known linear $\mathcal{H}_2$-norm is given by:

$$||\Sigma||^2_{\mathcal{H}_2} := \operatorname{tr}\left(\sum_{k=1}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\frac{1}{(2\pi)^k}\,\overline{H_k(i\omega_1,\ldots,i\omega_k)}H_k^T(i\omega_1,\ldots,i\omega_k)\right),$$

where H_k denotes the k-th transfer function associated with the bilinear system.

It can be shown that we can alternatively compute

$$||\Sigma||^2_{\mathcal{H}_2} = (\operatorname{vec}(I_p))^T\,(C\otimes C)\left(-A\otimes I - I\otimes A - \sum_{k=1}^{m} N_k\otimes N_k\right)^{-1}(B\otimes B)\operatorname{vec}(I_m).$$

In order to find an $\mathcal{H}_2$-optimal reduced system, we define the error system Σ^{err} as follows:

$$A^{err} = \begin{bmatrix} A & 0 \\ 0 & \hat{A}\end{bmatrix},\quad N_k^{err} = \begin{bmatrix} N_k & 0 \\ 0 & \hat{N}_k\end{bmatrix},\quad B^{err} = \begin{bmatrix} B \\ \hat{B}\end{bmatrix},\quad C^{err} = \begin{bmatrix} C & -\hat{C}\end{bmatrix}.$$

Necessary $\mathcal{H}_2$-Optimality Conditions

Let us assume $\hat{\Sigma}$ is given by its eigenvalue decomposition:

$$\hat{A} = R\Lambda R^{-1}, \quad \tilde{N}_k = R^{-1}\hat{N}_k R, \quad \tilde{B} = R^{-1}\hat{B}, \quad \tilde{C} = \hat{C}R.$$

Necessary $\mathcal{H}_2$-Optimality Conditions

Let us assume $\hat{\Sigma}$ is given by its eigenvalue decomposition:

$$\hat{A} = R\Lambda R^{-1}, \quad \tilde{N}_k = R^{-1}\hat{N}_k R, \quad \tilde{B} = R^{-1}\hat{B}, \quad \tilde{C} = \hat{C}R.$$

Using Λ, $\tilde{N}_k$, $\tilde{B}$, $\tilde{C}$ as optimization parameters, we can derive necessary conditions for $\mathcal{H}_2$-optimality:

Necessary $\mathcal{H}_2$-Optimality Conditions

Let us assume $\hat{\Sigma}$ is given by its eigenvalue decomposition:

$$\hat{A} = R\Lambda R^{-1}, \quad \tilde{N}_k = R^{-1}\hat{N}_k R, \quad \tilde{B} = R^{-1}\hat{B}, \quad \tilde{C} = \hat{C}R.$$

Using Λ, $\tilde{N}_k$, $\tilde{B}$, $\tilde{C}$ as optimization parameters, we can derive necessary conditions for $\mathcal{H}_2$-optimality:

$$\begin{aligned}
&(\mathrm{vec}(I_p))^T \left(e_i e_j^T \otimes C\right) \left(-\Lambda \otimes I_n - I_{\hat{n}} \otimes A - \sum_{k=1}^{m} \tilde{N}_k \otimes N_k\right)^{-1} \left(\tilde{B} \otimes B\right) \mathrm{vec}(I_m) \\
&= (\mathrm{vec}(I_p))^T \left(e_i e_j^T \otimes \hat{C}\right) \left(-\Lambda \otimes I_n - I_{\hat{n}} \otimes \hat{A} - \sum_{k=1}^{m} \tilde{N}_k \otimes \hat{N}_k\right)^{-1} \left(\tilde{B} \otimes \hat{B}\right) \mathrm{vec}(I_m).
\end{aligned}$$

Necessary $\mathcal{H}_2$-Optimality Conditions

Let us assume $\hat{\Sigma}$ is given by its eigenvalue decomposition:

$$\hat{A} = R\Lambda R^{-1}, \quad \tilde{N}_k = R^{-1}\hat{N}_k R, \quad \tilde{B} = R^{-1}\hat{B}, \quad \tilde{C} = \hat{C}R.$$

Using Λ, $\tilde{N}_k$, $\tilde{B}$, $\tilde{C}$ as optimization parameters, we can derive necessary conditions for $\mathcal{H}_2$-optimality:

$$\begin{aligned}
&(\mathrm{vec}(I_p))^T \left(e_i e_j^T \otimes C\right) \left(-\Lambda \otimes I_n - I_{\hat{n}} \otimes A - \sum_{k=1}^{m} \tilde{N}_k \otimes N_k\right)^{-1} \left(\tilde{B} \otimes B\right) \mathrm{vec}(I_m) \\
&= (\mathrm{vec}(I_p))^T \left(e_i e_j^T \otimes \hat{C}\right) \left(-\Lambda \otimes I_n - I_{\hat{n}} \otimes \hat{A} - \sum_{k=1}^{m} \tilde{N}_k \otimes \hat{N}_k\right)^{-1} \left(\tilde{B} \otimes \hat{B}\right) \mathrm{vec}(I_m).
\end{aligned}$$

$$\begin{aligned}
&(\mathrm{vec}(I_p))^T \left(e_i e_j^T \otimes C\right) (-\Lambda \otimes I_n - I_{\hat{n}} \otimes A)^{-1} \left(\tilde{B} \otimes B\right) \mathrm{vec}(I_m) \\
&= (\mathrm{vec}(I_p))^T \left(e_i e_j^T \otimes \hat{C}\right) \left(-\Lambda \otimes I_n - I_{\hat{n}} \otimes \hat{A}\right)^{-1} \left(\tilde{B} \otimes \hat{B}\right) \mathrm{vec}(I_m).
\end{aligned}$$

Necessary $\mathcal{H}_2$-Optimality Conditions

Let us assume $\hat{\Sigma}$ is given by its eigenvalue decomposition:

$$\hat{A} = R\Lambda R^{-1}, \quad \tilde{N}_k = R^{-1}\hat{N}_k R, \quad \tilde{B} = R^{-1}\hat{B}, \quad \tilde{C} = \hat{C}R.$$

Using Λ, $\tilde{N}_k$, $\tilde{B}$, $\tilde{C}$ as optimization parameters, we can derive necessary conditions for $\mathcal{H}_2$-optimality:

$$\begin{aligned}
&(\mathrm{vec}(I_p))^T \left(e_i e_j^T \otimes C\right) \left(-\Lambda \otimes I_n - I_{\hat{n}} \otimes A - \sum_{k=1}^{m} \tilde{N}_k \otimes N_k\right)^{-1} \left(\tilde{B} \otimes B\right) \mathrm{vec}(I_m) \\
&= (\mathrm{vec}(I_p))^T \left(e_i e_j^T \otimes \hat{C}\right) \left(-\Lambda \otimes I_n - I_{\hat{n}} \otimes \hat{A} - \sum_{k=1}^{m} \tilde{N}_k \otimes \hat{N}_k\right)^{-1} \left(\tilde{B} \otimes \hat{B}\right) \mathrm{vec}(I_m).
\end{aligned}$$

$$\begin{aligned}
&(\mathrm{vec}(I_p))^T \left(e_i e_j^T \otimes C\right) (-\Lambda \otimes I_n - I_{\hat{n}} \otimes A)^{-1} \mathrm{vec}(B\tilde{B}^T) \\
&= (\mathrm{vec}(I_p))^T \left(e_i e_j^T \otimes \hat{C}\right) \left(-\Lambda \otimes I_n - I_{\hat{n}} \otimes \hat{A}\right)^{-1} \mathrm{vec}(\hat{B}\tilde{B}^T).
\end{aligned}$$

Necessary $\mathcal{H}_2$-Optimality Conditions

Let us assume $\hat{\Sigma}$ is given by its eigenvalue decomposition:

$$\hat{A} = R\Lambda R^{-1}, \quad \tilde{N}_k = R^{-1}\hat{N}_k R, \quad \tilde{B} = R^{-1}\hat{B}, \quad \tilde{C} = \hat{C}R.$$

Using Λ, $\tilde{N}_k$, $\tilde{B}$, $\tilde{C}$ as optimization parameters, we can derive necessary conditions for $\mathcal{H}_2$-optimality:

$$\begin{aligned}
&(\mathrm{vec}(I_p))^T \left(e_i e_j^T \otimes C\right) \left(-\Lambda \otimes I_n - I_{\hat{n}} \otimes A - \sum_{k=1}^{m} \tilde{N}_k \otimes N_k\right)^{-1} \left(\tilde{B} \otimes B\right) \mathrm{vec}(I_m) \\
&= (\mathrm{vec}(I_p))^T \left(e_i e_j^T \otimes \hat{C}\right) \left(-\Lambda \otimes I_n - I_{\hat{n}} \otimes \hat{A} - \sum_{k=1}^{m} \tilde{N}_k \otimes \hat{N}_k\right)^{-1} \left(\tilde{B} \otimes \hat{B}\right) \mathrm{vec}(I_m).
\end{aligned}$$

$$\begin{aligned}
&(\mathrm{vec}(I_p))^T \left(e_i e_j^T \otimes C\right) \begin{pmatrix} -\lambda_1 I - A & & \\ & \ddots & \\ & & -\lambda_{\hat{n}} I - A \end{pmatrix}^{-1} \begin{pmatrix} B\tilde{B}_1^T \\ \vdots \\ B\tilde{B}_{\hat{n}}^T \end{pmatrix} \\
&= (\mathrm{vec}(I_p))^T \left(e_i e_j^T \otimes \hat{C}\right) \begin{pmatrix} -\lambda_1 I - \hat{A} & & \\ & \ddots & \\ & & -\lambda_{\hat{n}} I - \hat{A} \end{pmatrix}^{-1} \begin{pmatrix} \hat{B}\tilde{B}_1^T \\ \vdots \\ \hat{B}\tilde{B}_{\hat{n}}^T \end{pmatrix}.
\end{aligned}$$

Necessary $\mathcal{H}_2$-Optimality Conditions

Let us assume $\hat{\Sigma}$ is given by its eigenvalue decomposition:

$$\hat{A} = R\Lambda R^{-1}, \quad \tilde{N}_k = R^{-1}\hat{N}_k R, \quad \tilde{B} = R^{-1}\hat{B}, \quad \tilde{C} = \hat{C}R.$$

Using Λ, $\tilde{N}_k$, $\tilde{B}$, $\tilde{C}$ as optimization parameters, we can derive necessary conditions for $\mathcal{H}_2$-optimality:

$$\begin{aligned}
&(\mathrm{vec}(I_p))^T \left(e_i e_j^T \otimes C\right) \left(-\Lambda \otimes I_n - I_{\hat{n}} \otimes A - \sum_{k=1}^{m} \tilde{N}_k \otimes N_k\right)^{-1} \left(\tilde{B} \otimes B\right) \mathrm{vec}(I_m) \\
&= (\mathrm{vec}(I_p))^T \left(e_i e_j^T \otimes \hat{C}\right) \left(-\Lambda \otimes I_n - I_{\hat{n}} \otimes \hat{A} - \sum_{k=1}^{m} \tilde{N}_k \otimes \hat{N}_k\right)^{-1} \left(\tilde{B} \otimes \hat{B}\right) \mathrm{vec}(I_m).
\end{aligned}$$

$$H(-\lambda_j)\tilde{B}_j^T = \hat{H}(-\lambda_j)\tilde{B}_j^T$$

A Bilinear IRKA Approach

Algorithm 1 Bilinear IRKA

Input: A, N_k, B, C, $\hat{A}$, $\hat{N}_k$, $\hat{B}$, $\hat{C}$
Output: A^{opt}, N_k^{opt}, B^{opt}, C^{opt}

1: **while** (change in $\Lambda > \epsilon$) **do**
2: $R\Lambda R^{-1} = \hat{A}$, $\tilde{B} = R^{-1}\hat{B}$, $\tilde{C} = \hat{C}R$, $\tilde{N}_k = R^{-1}\hat{N}_k R$
3: $\operatorname{vec}(V) = \left(-\Lambda \otimes I_n - I_{\hat{n}} \otimes A - \sum_{k=1}^{m} \tilde{N}_k \otimes N_k\right)^{-1} \left(\tilde{B} \otimes B\right) \operatorname{vec}(I_m)$
4: $\operatorname{vec}(W) = \left(-\Lambda \otimes I_n - I_{\hat{n}} \otimes A^T - \sum_{k=1}^{m} \tilde{N}_k^T \otimes N_k^T\right)^{-1} \left(\tilde{C}^T \otimes C^T\right) \operatorname{vec}(I_p)$
5: $V = \operatorname{orth}(V)$, $W = \operatorname{orth}(W)$
6: $\hat{A} = \left(W^T V\right)^{-1} W^T A V$, $\hat{N}_k = \left(W^T V\right)^{-1} W^T N_k V$,
$\hat{B} = \left(W^T V\right)^{-1} W^T B$, $\hat{C} = CV$
7: **end while**
8: $A^{opt} = \hat{A}$, $N_k^{opt} = \hat{N}_k$, $B^{opt} = \hat{B}$, $C^{opt} = \hat{C}$

A Heat Transfer Model

- 2-dimensional heat distribution [Benner, Saak '05]
- boundary control by spraying intensities of a cooling fluid

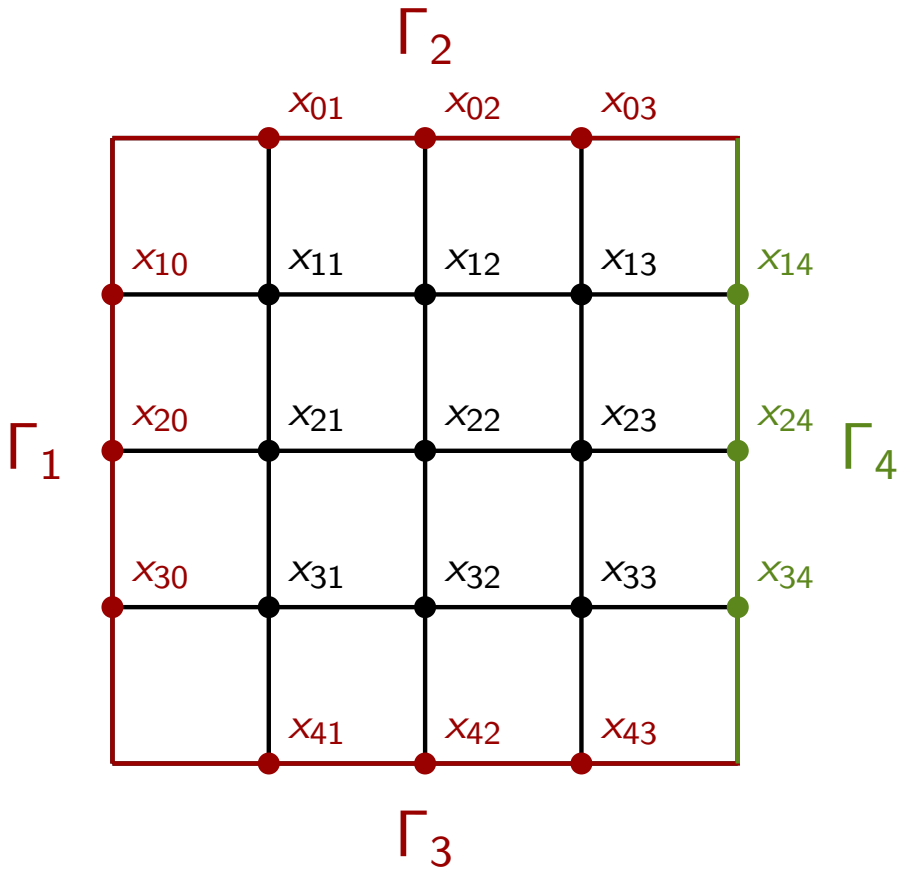
$$\begin{aligned} \Omega &= (0,1) \times (0,1) & \\ x_t &= \Delta x & \text{in } \Omega \\ n \cdot \nabla x &= u_{1,2,3}(x-1) & \text{on } \Gamma_1, \Gamma_2, \Gamma_3 \\ x &= u_4 & \text{on } \Gamma_4 \end{aligned}$$

- spatial discretization $k \times k$-grid

$$\Rightarrow \dot{x} \approx A_1 x + \sum_{i=1}^{3} N_i x u_i + Bu$$

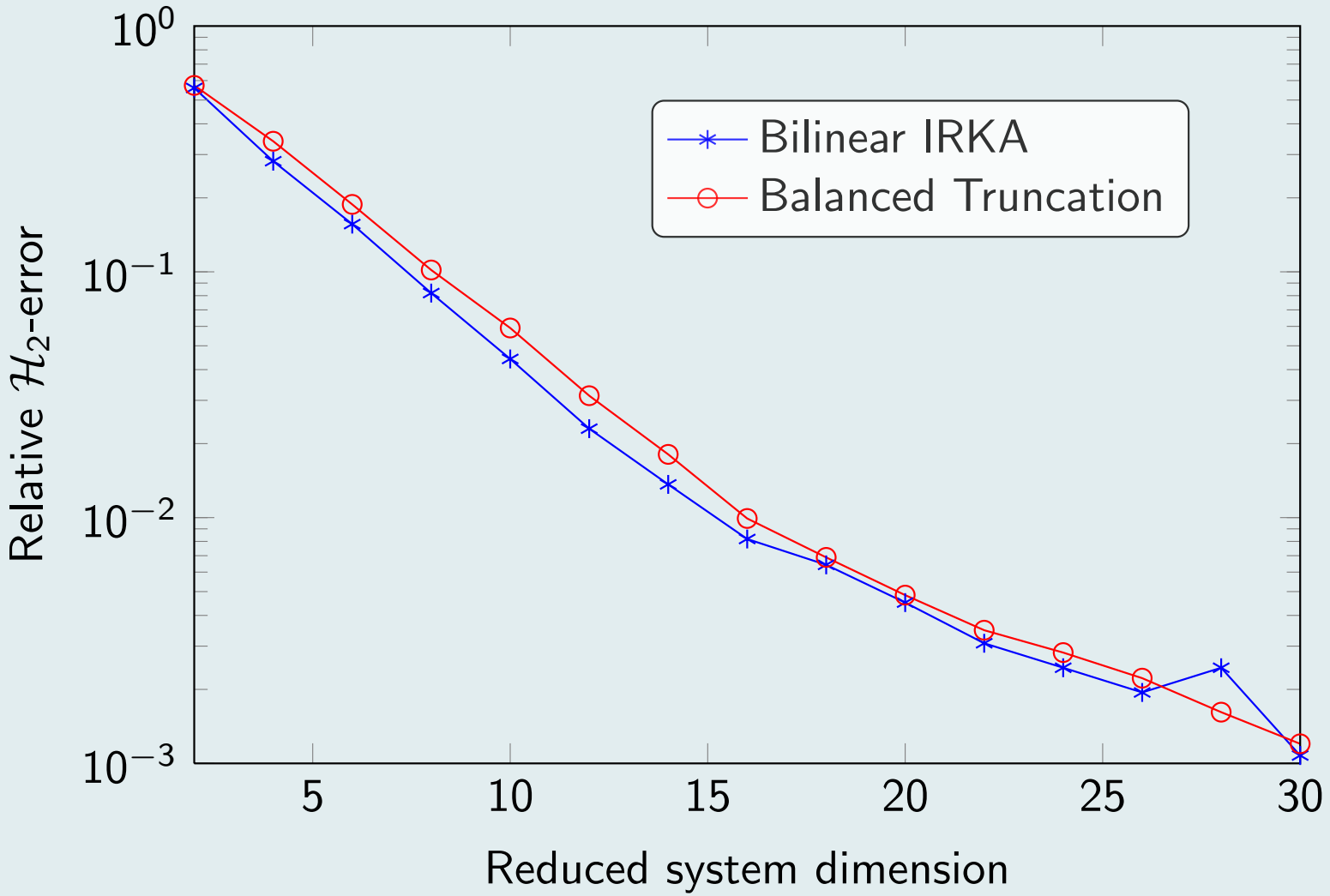
$$\Rightarrow A_2 = 0$$

- output: $y = \frac{1}{k^2}\begin{bmatrix} 1 & \dots & 1 \end{bmatrix}$



A Heat Transfer Model

Comparison of relative $\mathcal{H}_2$-error for $n = 10000$



Outlook

To do:

- investigate possible two-sided reduction methods for quadratic-bilinear systems.

Outlook

To do:

- investigate possible two-sided reduction methods for quadratic-bilinear systems.
- Note that $V \in \mathbb{R}^{n \times q}$ in general is dense
  $\rightsquigarrow$ computation of $\hat{A}_2 = V^T A_2 V \otimes V$ might cause problems,
  $\rightsquigarrow$ find approximations:

$$A_2 \approx g_1 \otimes G_1 + \ldots g_r \otimes G_r,$$

with $g_i^T \in \mathbb{R}^n$, $G_i \in \mathbb{R}^{n \times n}$ and $r \ll n$.

Outlook

To do:

- investigate possible two-sided reduction methods for quadratic-bilinear systems.
- Note that $V \in \mathbb{R}^{n \times q}$ in general is dense
  $\rightsquigarrow$ computation of $\hat{A}_2 = V^T A_2 V \otimes V$ might cause problems,
  $\rightsquigarrow$ find approximations:

$$A_2 \approx g_1 \otimes G_1 + \ldots g_r \otimes G_r,$$

  with $g_i^T \in \mathbb{R}^n$, $G_i \in \mathbb{R}^{n \times n}$ and $r \ll n$.
- Lyapunov-based reduction possible?

Outlook

To do:

- investigate possible two-sided reduction methods for quadratic-bilinear systems.
- Note that $V \in \mathbb{R}^{n \times q}$ in general is dense
  $\rightsquigarrow$ computation of $\hat{A}_2 = V^T A_2 V \otimes V$ might cause problems,
  $\rightsquigarrow$ find approximations:

$$A_2 \approx g_1 \otimes G_1 + \ldots g_r \otimes G_r,$$

  with $g_i^T \in \mathbb{R}^n$, $G_i \in \mathbb{R}^{n \times n}$ and $r \ll n$.
- Lyapunov-based reduction possible?

Thank you for your attention!