MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

1998
2023

CSC

COMPUTATIONAL METHODS IN
SYSTEMS AND CONTROL THEORY

Quadrature-based balanced truncation for Quadratic-bilinear systems

Reetish Padhi

joint work with Ion Victor Gosea at Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany

Young Mathematicians in model order reduction 2024, University of Stuttgart, Germany

March 5th, 2024

Outline

- Part I: Linear time-invariant systems
  1. Definition and properties
  2. Balanced truncation (BT)
  3. Data-driven balancing (Quad BT)

- Part II: Quadratic-bilinear time-invariant systems
  1. Definition and properties
  2. Balanced truncation (BT)
  3. Data-driven balancing (Quad BT)

- Conclusion and outlook

Part I: Linear systems theory

LTI definition and kernels

- A linear time-invariant (LTI) system $\mathbf{\Sigma} = (\mathbf{A}, \mathbf{B}, \mathbf{C})$ is a dynamical system given by the following set of differential equations:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t),\\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t).\end{aligned} \tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^n, \mathbf{y}(t) \in \mathbb{R}$ and $\mathbf{u}(t) \in \mathbb{R}$.

LTI definition and kernels

- A linear time-invariant (LTI) system $\mathbf{\Sigma} = (\mathbf{A}, \mathbf{B}, \mathbf{C})$ is a dynamical system given by the following set of differential equations:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t). \end{aligned} \tag{1}$$

  where $\mathbf{x}(t) \in \mathbb{R}^n, \mathbf{y}(t) \in \mathbb{R}$ and $\mathbf{u}(t) \in \mathbb{R}$.

- Solving for $\mathbf{y}(t)$ while assuming $\mathbf{x}(0) = 0$:

$$\mathbf{y}(t) = \int_0^t \mathbf{C}e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau = \int_0^t \underbrace{\mathbf{C}e^{\mathbf{A}\tau}\mathbf{B}}_{\mathbf{h}(\tau)}\mathbf{u}(t-\tau)d\tau = \underbrace{(\mathbf{h} * \mathbf{u})}_{\text{convolution}}(t)$$

LTI definition and kernels

- A linear time-invariant (LTI) system $\boldsymbol{\Sigma} = (\mathbf{A}, \mathbf{B}, \mathbf{C})$ is a dynamical system given by the following set of differential equations:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t). \end{aligned} \tag{1}$$

  where $\mathbf{x}(t) \in \mathbb{R}^n, \mathbf{y}(t) \in \mathbb{R}$ and $\mathbf{u}(t) \in \mathbb{R}$.

- Solving for $\mathbf{y}(t)$ while assuming $\mathbf{x}(0) = 0$:

$$\mathbf{y}(t) = \int_0^t \mathbf{C}e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau = \int_0^t \underbrace{\mathbf{C}e^{\mathbf{A}\tau}\mathbf{B}}_{\mathbf{h}(\tau)}\mathbf{u}(t-\tau)d\tau = \underbrace{(\mathbf{h} * \mathbf{u})}_{\text{convolution}}(t)$$

- The **impulse response** or **kernel** $\mathbf{h}(\tau) = \mathbf{C}e^{\mathbf{A}\tau}\mathbf{B}$ describes the input-output transfer in the time domain.

Gramians and Lyapunov equations

Definition

The infinite Gramians for a stable LTI system $\mathbf{\Sigma} = (\mathbf{A}, \mathbf{B}, \mathbf{C})$ *are defined as,*

$$\mathbf{P} = \int_0^\infty e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^\top e^{\mathbf{A}^\top t} dt, \tag{2}$$

$$\mathbf{Q} = \int_0^\infty e^{\mathbf{A}^\top t}\mathbf{C}^\top \mathbf{C} e^{\mathbf{A}t} dt. \tag{3}$$

- The reachability Gramian $\mathbf{P}$ quantifies how easily a state is reachable from the zero state.
- Similarly, the observability Gramian $\mathbf{Q}$ quantifies how easily a state is distinguishable from the zero state.

Gramians and Lyapunov equations

Definition

The infinite Gramians for a stable LTI system $\mathbf{\Sigma} = (\mathbf{A}, \mathbf{B}, \mathbf{C})$ *are defined as,*

$$\mathbf{P} = \int_0^\infty e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^\top e^{\mathbf{A}^\top t} dt, \tag{2}$$

$$\mathbf{Q} = \int_0^\infty e^{\mathbf{A}^\top t}\mathbf{C}^\top \mathbf{C} e^{\mathbf{A}t} dt. \tag{3}$$

- The reachability Gramian $\mathbf{P}$ quantifies how easily a state is reachable from the zero state.
- Similarly, the observability Gramian $\mathbf{Q}$ quantifies how easily a state is distinguishable from the zero state.
- The Gramians $\mathbf{P}, \mathbf{Q}$ satisfy the following (linear) Lyapunov matrix equations,

$$\mathbf{A}\mathbf{P} + \mathbf{P}\mathbf{A}^\top + \mathbf{B}\mathbf{B}^\top = \mathbf{0},$$

$$\mathbf{A}^\top \mathbf{Q} + \mathbf{Q}\mathbf{A} + \mathbf{C}^\top \mathbf{C} = \mathbf{0}.$$

Balanced Truncation

- Originally introduced in:

Principal component analysis in linear systems: Controllability, observability, and model reduction, B. Moore, IEEE Transactions on Automatic Control, 26 (1), 1981.

and

Model reduction via balanced state space representations , L. Pernebo, L. Silverman, IEEE Transactions on Automatic Control, 27 (2), 382-387, 1982.

Square-root BT algorithm (LTI systems)

- Let $\mathbf{U}, \mathbf{L}$ be square factors of the Gramians, i.e., $\mathbf{P} = \mathbf{U}\mathbf{U}^T$ and $\mathbf{Q} = \mathbf{L}\mathbf{L}^T$.

Square-root BT algorithm (LTI systems)

- Let $\mathbf{U}, \mathbf{L}$ be square factors of the Gramians, i.e., $\mathbf{P} = \mathbf{U}\mathbf{U}^T$ and $\mathbf{Q} = \mathbf{L}\mathbf{L}^T$.
- Compute the truncated **SVD** of matrix $\mathbf{L}^T\mathbf{U}$:

$$\mathbf{L}^T\mathbf{U} = \begin{bmatrix} \mathbf{Z}_1 & \mathbf{Z}_2 \end{bmatrix} \begin{bmatrix} \mathbf{S}_1 \in \mathbb{R}^{r\times r} & \\ & \mathbf{S}_2 \end{bmatrix} \begin{bmatrix} \mathbf{Y}_1^\top \\ \mathbf{Y}_2^\top \end{bmatrix}$$

Square-root BT algorithm (LTI systems)

- Let $\mathbf{U}, \mathbf{L}$ be square factors of the Gramians, i.e., $\mathbf{P} = \mathbf{U}\mathbf{U}^T$ and $\mathbf{Q} = \mathbf{L}\mathbf{L}^T$.
- Compute the truncated **SVD** of matrix $\mathbf{L}^T\mathbf{U}$:

$$\mathbf{L}^T\mathbf{U} = \begin{bmatrix} \mathbf{Z}_1 & \mathbf{Z}_2 \end{bmatrix} \begin{bmatrix} \mathbf{S}_1 \in \mathbb{R}^{r \times r} & \\ & \mathbf{S}_2 \end{bmatrix} \begin{bmatrix} \mathbf{Y}_1^\top \\ \mathbf{Y}_2^\top \end{bmatrix}$$

- Construct the model reduction bases

$$\mathbf{W}_r = \mathbf{L}\mathbf{Z}_1\mathbf{S}_1^{-1/2} \text{ and } \mathbf{V}_r = \mathbf{U}\mathbf{Y}_1\mathbf{S}_1^{-1/2}.$$

Square-root BT algorithm (LTI systems)

- Let $\mathbf{U}, \mathbf{L}$ be square factors of the Gramians, i.e., $\mathbf{P} = \mathbf{U}\mathbf{U}^T$ and $\mathbf{Q} = \mathbf{L}\mathbf{L}^T$.
- Compute the truncated **SVD** of matrix $\mathbf{L}^T\mathbf{U}$:

$$\mathbf{L}^T\mathbf{U} = \begin{bmatrix} \mathbf{Z}_1 & \mathbf{Z}_2 \end{bmatrix} \begin{bmatrix} \mathbf{S}_1 \in \mathbb{R}^{r\times r} & \\ & \mathbf{S}_2 \end{bmatrix} \begin{bmatrix} \mathbf{Y}_1^\top \\ \mathbf{Y}_2^\top \end{bmatrix}$$

- Construct the model reduction bases

$$\mathbf{W}_r = \mathbf{L}\mathbf{Z}_1\mathbf{S}_1^{-1/2} \text{ and } \mathbf{V}_r = \mathbf{U}\mathbf{Y}_1\mathbf{S}_1^{-1/2}.$$

- The matrices of reduced-order balanced system $\mathbf{\Sigma}_r$ are given by

$$\mathbf{A}_r = \mathbf{W}_r^T\mathbf{A}\mathbf{V}_r = \mathbf{S}_1^{-1/2}\mathbf{Z}_1^T\left(\mathbf{L}^T\mathbf{A}\mathbf{U}\right)\mathbf{Y}_1\mathbf{S}_1^{-1/2},$$

$$\mathbf{B}_r = \mathbf{W}_r^T\mathbf{B} = \mathbf{S}_1^{-1/2}\mathbf{Z}_1^T\left(\mathbf{L}^T\mathbf{B}\right),$$

$$\mathbf{C}_r = \mathbf{C}\mathbf{V}_r = \left(\mathbf{C}\mathbf{U}\right)\mathbf{Y}_1\mathbf{S}_1^{-1/2}.$$

Data-driven balancing (QuadBT)

- Originally introduced in:

Data-driven balancing of linear dynamical systems, I.V. Gosea, S. Gugercin, C. Beattie, SIAM Journal on Scientific Computing 44 (1), A554-A582, 2022.

Motivation

- In many applications, we might not have explicit access to the system matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$!

Motivation

- In many applications, we might not have explicit access to the system matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$!

- **QuadBT** algorithm tries to address this very scenario by approximating the terms $\mathbf{L}^\top \mathbf{U}, \mathbf{L}^\top \mathbf{A} \mathbf{U}, \mathbf{L}^\top \mathbf{B}$ and $\mathbf{C} \mathbf{U}$ (**appearing in orange**) in the previous slide with data-driven quantities.

Motivation

- In many applications, we might not have explicit access to the system matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$!
- **QuadBT** algorithm tries to address this very scenario by approximating the terms $\mathbf{L}^\top\mathbf{U}, \mathbf{L}^\top\mathbf{A}\mathbf{U}, \mathbf{L}^\top\mathbf{B}$ and $\mathbf{C}\mathbf{U}$ (**appearing in orange**) in the previous slide with data-driven quantities.
- **QuadBT** is different from other data-driven **BT** methods in that it is based solely on samples of the **kernels** (time domain) or **transfer function** (frequency domain) without requiring access to state trajectories or other state-based quantities.

Quadrature approximation of Gramians

- We use quadrature nodes to approximate the square root factors of the Gramians $\mathbf{L}$ and $\mathbf{U}$ as follows:

$$\mathbf{P} = \int_0^\infty e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^\top e^{\mathbf{A}^\top t}dt \approx \widetilde{\mathcal{P}} = \sum_{i=1}^{N_p} \rho_i^2 e^{\mathbf{A}\mu_i}\mathbf{B}\mathbf{B}^\top e^{\mathbf{A}^\top \mu_i} = \widetilde{\mathbf{U}}\widetilde{\mathbf{U}}^\top,$$

$$\mathbf{Q} = \int_0^\infty e^{\mathbf{A}^\top t}\mathbf{C}^\top\mathbf{C}e^{\mathbf{A}t}dt \approx \widetilde{\mathcal{Q}} = \sum_{j=1}^{N_q} \phi_j^2 e^{\mathbf{A}^\top \omega_j}\mathbf{C}^\top\mathbf{C}e^{\mathbf{A}\omega_j} = \widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^\top,$$

where N_p, N_q denote the number of quadrature nodes used in $\widetilde{\mathcal{P}}$ and $\widetilde{\mathcal{Q}}$ respectively, $\{\mu_i, \omega_j\}$ and $\{\rho_i, \phi_j\}$ denote the quadrature nodes and weights.

Quadrature approximation of Gramians

- We use quadrature nodes to approximate the square root factors of the Gramians $\mathbf{L}$ and $\mathbf{U}$ as follows:

$$\mathbf{P} = \int_0^\infty e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^\top e^{\mathbf{A}^\top t}dt \approx \widetilde{\mathcal{P}} = \sum_{i=1}^{N_p} \rho_i^2 e^{\mathbf{A}\mu_i}\mathbf{B}\mathbf{B}^\top e^{\mathbf{A}^\top \mu_i} = \widetilde{\mathbf{U}}\widetilde{\mathbf{U}}^\top,$$

$$\mathbf{Q} = \int_0^\infty e^{\mathbf{A}^\top t}\mathbf{C}^\top\mathbf{C}e^{\mathbf{A}t}dt \approx \widetilde{\mathcal{Q}} = \sum_{j=1}^{N_q} \phi_j^2 e^{\mathbf{A}^\top \omega_j}\mathbf{C}^\top\mathbf{C}e^{\mathbf{A}\omega_j} = \widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^\top,$$

  where N_p, N_q denote the number of quadrature nodes used in $\widetilde{\mathcal{P}}$ and $\widetilde{\mathcal{Q}}$ respectively, $\{\mu_i, \omega_j\}$ and $\{\rho_i, \phi_j\}$ denote the quadrature nodes and weights.

- Observing $\widetilde{\mathcal{P}}$ and $\widetilde{\mathcal{Q}}$, we construct the quadrature based square factors $\widetilde{\mathbf{U}}$ and $\widetilde{\mathbf{L}}$.

$$\mathbf{U} \approx \widetilde{\mathbf{U}} = \left[\rho_1 e^{\mathbf{A}\mu_1}\mathbf{B} \cdots \rho_{N_p} e^{\mathbf{A}\mu_{N_p}}\mathbf{B}\right] \tag{4}$$

$$\mathbf{L} \approx \widetilde{\mathbf{L}} = \left[\phi_1 e^{\mathbf{A}^\top \omega_1}\mathbf{C}^\top \cdots \phi_{N_q} e^{\mathbf{A}^\top \omega_{N_q}}\mathbf{C}^\top\right]. \tag{5}$$

Replacing intrusive terms with data

- Recall the terms appearing in the **BT** algorithm i.e. $\mathbf{L}^\top\mathbf{U}$,$\mathbf{L}^\top\mathbf{A}\mathbf{U}$,$\mathbf{L}^\top\mathbf{B}$ and $\mathbf{C}\mathbf{U}$. Start by replacing the $\mathbf{L}^\top\mathbf{U}$ term with $\widetilde{\mathbb{H}} = \widetilde{\mathbf{L}}^\top\widetilde{\mathbf{U}}$.

Replacing intrusive terms with data

- Recall the terms appearing in the **BT** algorithm i.e. $\mathbf{L}^\top\mathbf{U}$,$\mathbf{L}^\top\mathbf{A}\mathbf{U}$,$\mathbf{L}^\top\mathbf{B}$ and $\mathbf{C}\mathbf{U}$. Start by replacing the $\mathbf{L}^\top\mathbf{U}$ term with $\widetilde{\mathbb{H}} = \widetilde{\mathbf{L}}^\top\widetilde{\mathbf{U}}$.
- The $\widetilde{\mathbb{H}}$ matrix can be expressed as samples of the kernel $\mathbf{h}(t)$ as shown below,

$$\widetilde{\mathbb{H}} = \begin{bmatrix} \rho_1\phi_1\mathbf{h}(\mu_1+\omega_1) & \dots & \rho_{N_p}\phi_1\mathbf{h}(\mu_{N_p}+\omega_1) \\ \vdots & \ddots & \vdots \\ \rho_1\phi_{N_q}\mathbf{h}(\mu_1+\omega_{N_q}) & \dots & \rho_{N_p}\phi_{N_q}\mathbf{h}(\mu_{N_p}+\omega_{N_q}) \end{bmatrix}$$

Replacing intrusive terms with data

- Recall the terms appearing in the **BT** algorithm i.e. $\mathbf{L}^\top\mathbf{U}$,$\mathbf{L}^\top\mathbf{A}\mathbf{U}$,$\mathbf{L}^\top\mathbf{B}$ and $\mathbf{C}\mathbf{U}$. Start by replacing the $\mathbf{L}^\top\mathbf{U}$ term with $\widetilde{\mathbb{H}} = \widetilde{\mathbf{L}}^\top\widetilde{\mathbf{U}}$.
- The $\widetilde{\mathbb{H}}$ matrix can be expressed as samples of the kernel $\mathbf{h}(t)$ as shown below,

$$\widetilde{\mathbb{H}} = \begin{bmatrix} \rho_1\phi_1\mathbf{h}(\mu_1+\omega_1) & \dots & \rho_{N_p}\phi_1\mathbf{h}(\mu_{N_p}+\omega_1) \\ \vdots & \ddots & \vdots \\ \rho_1\phi_{N_q}\mathbf{h}(\mu_1+\omega_{N_q}) & \dots & \rho_{N_p}\phi_{N_q}\mathbf{h}(\mu_{N_p}+\omega_{N_q}) \end{bmatrix}$$

- Similarly, for $\mathbf{L}^\top\mathbf{A}\mathbf{U}$ we have the term $\widetilde{\mathbb{M}} = \widetilde{\mathbf{L}}^\top\mathbf{A}\widetilde{\mathbf{U}}$ which can be written as

$$\widetilde{\mathbb{M}} = \begin{bmatrix} \rho_1\phi_1\mathbf{h}'(\mu_1+\omega_1) & \dots & \rho_{N_p}\phi_1\mathbf{h}'(\mu_{N_p}+\omega_1) \\ \vdots & \ddots & \vdots \\ \rho_1\phi_{N_q}\mathbf{h}'(\mu_1+\omega_{N_q}) & \dots & \rho_{N_p}\phi_{N_q}\mathbf{h}'(\mu_{N_p}+\omega_{N_q}) \end{bmatrix}$$

Replacing intrusive terms with data

- Similarly, $\mathbf{L}^\top \mathbf{B} \approx \widetilde{\mathbb{g}} = \widetilde{\mathbf{L}}^\top \mathbf{B}$ can be written as

$$\widetilde{\mathbb{g}} = \begin{bmatrix} \phi_1 \mathbf{h}(\omega_1) \\ \vdots \\ \phi_1 \mathbf{h}(\omega_{N_q}) \end{bmatrix}$$

Replacing intrusive terms with data

- Similarly, $\mathbf{L}^\top \mathbf{B} \approx \widetilde{\mathbb{g}} = \widetilde{\mathbf{L}}^\top \mathbf{B}$ can be written as

$$\widetilde{\mathbb{g}} = \begin{bmatrix} \phi_1 \mathbf{h}(\omega_1) \\ \vdots \\ \phi_1 \mathbf{h}(\omega_{N_q}) \end{bmatrix}$$

- Finally $\widetilde{\mathbb{h}} = \mathbf{C}\widetilde{\mathbf{U}}$ vector can also be expressed as samples of the kernel $\mathbf{h}(t)$ as shown below,

$$\widetilde{\mathbb{h}} = \begin{bmatrix} \rho_1 \mathbf{h}(\mu_1) & \dots & \rho_{N_p} \mathbf{h}(\mu_{N_p}) \end{bmatrix}$$

Replacing intrusive terms with data

- Similarly, $\mathbf{L}^\top \mathbf{B} \approx \widetilde{\mathbb{g}} = \widetilde{\mathbf{L}}^\top \mathbf{B}$ can be written as

$$\widetilde{\mathbb{g}} = \begin{bmatrix} \phi_1 \mathbf{h}(\omega_1) \\ \vdots \\ \phi_1 \mathbf{h}(\omega_{N_q}) \end{bmatrix}$$

- Finally $\widetilde{\mathbb{h}} = \mathbf{C}\widetilde{\mathbf{U}}$ vector can also be expressed as samples of the kernel $\mathbf{h}(t)$ as shown below,

$$\widetilde{\mathbb{h}} = \begin{bmatrix} \rho_1 \mathbf{h}(\mu_1) & \ldots & \rho_{N_p} \mathbf{h}(\mu_{N_p}) \end{bmatrix}$$

- Having replaced all four terms with data-driven quantities, we are now ready to state the **QuadBT** algorithm.

- Put together the data matrices:

$$\widetilde{\mathbb{H}}, \quad \widetilde{\mathbb{M}}, \quad \widetilde{\mathbb{h}}, \quad \widetilde{\mathbb{g}}.$$

- Compute the truncated **SVD** of matrix $\widetilde{\mathbb{H}} \in \mathbb{R}^{N_q \times N_p}$:

$$\widetilde{\mathbb{H}} = \begin{bmatrix} \widetilde{\mathbf{Z}}_1 & \widetilde{\mathbf{Z}}_2 \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{S}}_1 & \\ & \widetilde{\mathbf{S}}_2 \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{Y}}_1^\top \\ \widetilde{\mathbf{Y}}_2^\top \end{bmatrix}$$

- The matrices of the reduced-order data-driven system $\mathbf{\Sigma}_r$ are given by

$$\tilde{\mathbf{A}}_r = \cancel{\mathbf{W}_r^T \mathbf{A} \mathbf{V}_r} = \widetilde{\mathbf{S}}_1^{-1/2} \widetilde{\mathbf{Z}}_1^* \left(\widetilde{\mathbb{M}}\right) \widetilde{\mathbf{Y}}_1 \widetilde{\mathbf{S}}_1^{-1/2},$$

$$\tilde{\mathbf{B}}_r = \cancel{\mathbf{W}_r^T \mathbf{B}} = \mathbf{S}_1^{-1/2} \mathbf{Z}_1^* \left(\widetilde{\mathbb{g}}\right),$$

$$\tilde{\mathbf{C}}_r = \cancel{\mathbf{C} \mathbf{V}_r} = \left(\widetilde{\mathbb{h}}\right) \widetilde{\mathbf{Y}}_1 \widetilde{\mathbf{S}}_1^{-1/2}.$$

Part II: Quadratic-bilinear (QB) systems

Definition and properties

- Quadratic-bilinear (QB) systems are denoted by the following equations,

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{H}(\mathbf{x}(t)\otimes\mathbf{x}(t)) + \mathbf{N}\mathbf{x}(t)\mathbf{u}(t) + \mathbf{B}\mathbf{u}(t),\\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t),\end{aligned}$$

  where $\mathbf{u}(t)\in\mathbb{R}, \mathbf{x}(t)\in\mathbb{R}^n$ and $\mathbf{y}(t)\in\mathbb{R}$.
- We stick to the SISO case for simplicity.
- WLOG the matrix $\mathbf{H}$ can be considered to be the 1-matricization of a tensor $\mathcal{H}$, where $\mathcal{H}^{(2)} = \mathcal{H}^{(3)}$.
- A large class of nonlinear systems can be written as QB systems using lifting making it an important class of systems.

Kernels of QB systems

- The solution $\mathbf{x}(t)$ of the QB system $\boldsymbol{\Sigma}^{\mathsf{QB}}$ can be expressed as the sum of infinite subsystems as $\mathbf{x}(t) = \sum_{k=1}^{\infty} \mathbf{x}_k(t)$ given by

$$\dot{\mathbf{x}}_1(t) = \mathbf{A}\mathbf{x}_1(t) + \mathbf{B}\mathbf{u}(t),$$

$$\dot{\mathbf{x}}_k(t) = \mathbf{A}\mathbf{x}_k(t) + \mathbf{N}\mathbf{x}_{k-1}(t)\mathbf{u}(t) + \sum_{i=1}^{k-1} \mathbf{H}(\mathbf{x}_i(t) \otimes \mathbf{x}_{k-i}(t)) \text{ for } \; k > 1.$$

Kernels of QB systems

- The solution $\mathbf{x}(t)$ of the QB system Σ^{QB} can be expressed as the sum of infinite subsystems as $\mathbf{x}(t) = \sum_{k=1}^{\infty} \mathbf{x}_k(t)$ given by

$$\dot{\mathbf{x}}_1(t) = \mathbf{A}\mathbf{x}_1(t) + \mathbf{B}\mathbf{u}(t),$$

$$\dot{\mathbf{x}}_k(t) = \mathbf{A}\mathbf{x}_k(t) + \mathbf{N}\mathbf{x}_{k-1}(t)\mathbf{u}(t) + \sum_{i=1}^{k-1} \mathbf{H}(\mathbf{x}_i(t) \otimes \mathbf{x}_{k-i}(t)) \text{ for } \; k > 1.$$

- The solution of $\mathbf{x}_k(t)$ is given by

$$\mathbf{x}_k(t) = \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{N}\mathbf{x}_{k-1}(\tau)\mathbf{u}(\tau)d\tau + \sum_{i=1}^{k-1} \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{H}(\mathbf{x}_i(\tau) \otimes \mathbf{x}_{k-i}(\tau))d\tau$$

Kernels of QB systems

- The solution $\mathbf{x}(t)$ of the QB system Σ^{QB} can be expressed as the sum of infinite subsystems as $\mathbf{x}(t) = \sum_{k=1}^{\infty} \mathbf{x}_k(t)$ given by

$$\dot{\mathbf{x}}_1(t) = \mathbf{A}\mathbf{x}_1(t) + \mathbf{B}\mathbf{u}(t),$$

$$\dot{\mathbf{x}}_k(t) = \mathbf{A}\mathbf{x}_k(t) + \mathbf{N}\mathbf{x}_{k-1}(t)\mathbf{u}(t) + \sum_{i=1}^{k-1} \mathbf{H}(\mathbf{x}_i(t) \otimes \mathbf{x}_{k-i}(t)) \text{ for } \ k > 1.$$

- The solution of $\mathbf{x}_k(t)$ is given by

$$\mathbf{x}_k(t) = \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{N}\mathbf{x}_{k-1}(\tau)\mathbf{u}(\tau)d\tau + \sum_{i=1}^{k-1} \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{H}(\mathbf{x}_i(\tau) \otimes \mathbf{x}_{k-i}(\tau))d\tau$$

- Finally this gives us, $\mathbf{y}(t) = \mathbf{C}\sum_{k=1}^{\infty} \mathbf{x}_k(t)$.

Kernels of QB systems

$$\mathbf{y}(t) = \sum_{k=1}^{\infty}\left(\int_0^t \underbrace{\mathbf{C}e^{\mathbf{A}(t-\tau)}\mathbf{N}\mathbf{x}_{k-1}(\tau)}_{\text{Bilinear contribution}}\mathbf{u}(\tau)d\tau + \sum_{i=1}^{k-1}\int_0^t \underbrace{\mathbf{C}e^{\mathbf{A}(t-\tau)}\mathbf{H}(\mathbf{x}_i(\tau)\otimes\mathbf{x}_{k-i}(\tau))}_{\text{Quadratic contribution}}d\tau\right)$$

- Unlike the linear case, we don't have a single kernel $\mathbf{h}$ but a collection of them.
- Then, the QB kernels are,

$$\begin{aligned}
\mathbf{K}_1 &= \{\mathbf{C}e^{\mathbf{A}t}\mathbf{B}\}\\
\mathbf{K}_2 &= \{\mathbf{C}e^{\mathbf{A}t_1}\mathbf{N}e^{\mathbf{A}t_2}\mathbf{B}, \mathbf{C}e^{\mathbf{A}t_1}\mathbf{H}(e^{\mathbf{A}t_2}\mathbf{B}\otimes e^{\mathbf{A}t_3}\mathbf{B})\}\\
\mathbf{K}_3 &= \{\mathbf{C}e^{\mathbf{A}t_1}\mathbf{N}e^{\mathbf{A}t_2}\mathbf{N}e^{\mathbf{A}t_3}\mathbf{B}, \mathbf{C}e^{\mathbf{A}t_1}\mathbf{N}e^{\mathbf{A}t_2}\mathbf{H}(e^{\mathbf{A}t_3}\mathbf{B}\otimes e^{\mathbf{A}t_4}\mathbf{B}),\\
&\quad \mathbf{C}e^{\mathbf{A}t_1}\mathbf{H}(e^{\mathbf{A}t_2}\mathbf{B}\otimes e^{\mathbf{A}t_3}\mathbf{H}(e^{\mathbf{A}t_4}\mathbf{B}\otimes e^{\mathbf{A}t_5}\mathbf{B}),\\
&\quad \mathbf{C}e^{\mathbf{A}t_1}\mathbf{H}(e^{\mathbf{A}t_2}\mathbf{B}\otimes e^{\mathbf{A}t_3}\mathbf{N}e^{\mathbf{A}t_4}\mathbf{B})), \mathbf{C}e^{\mathbf{A}t_1}\mathbf{H}(e^{\mathbf{A}t_2}\mathbf{N}e^{\mathbf{A}t_3}\mathbf{B}\otimes e^{\mathbf{A}t_4}\mathbf{B}),\\
&\quad \mathbf{C}e^{\mathbf{A}t_1}\mathbf{H}(e^{\mathbf{A}t_2}\mathbf{H}(e^{\mathbf{A}t_3}\mathbf{B}\otimes e^{\mathbf{A}t_4}\mathbf{B})\otimes e^{\mathbf{A}t_5}\mathbf{B}))\}\\
&\;\;\vdots
\end{aligned}$$

Kernel notation

- Having defined the kernels, we now need a way to notate them.
- Some examples:

$$\mathbf{h}_3^{\mathbf{N},\mathbf{N}}(t_1,t_2,t_3) = \mathbf{C}e^{\mathbf{A}t_1}\mathbf{N}e^{\mathbf{A}t_2}\mathbf{N}e^{\mathbf{A}t_3}\mathbf{B}$$

$$\mathbf{h}_3^{\mathbf{N},\mathbf{H}(-,-)}(t_1,t_2,t_3,t_4) = \mathbf{C}e^{\mathbf{A}t_1}\mathbf{N}e^{\mathbf{A}t_2}\mathbf{H}\left(e^{\mathbf{A}t_3}\mathbf{B}\otimes e^{\mathbf{A}t_4}\mathbf{B}\right)$$

$$\mathbf{h}_3^{\mathbf{H}(-,\mathbf{N})}(t_1,t_2,t_3,t_4) = \mathbf{C}e^{\mathbf{A}t_1}\mathbf{H}\left(e^{\mathbf{A}t_2}\mathbf{B}\otimes e^{\mathbf{A}t_3}\mathbf{N}e^{\mathbf{A}t_4}\mathbf{B}\right)$$

$$\mathbf{h}_4^{\mathbf{H}(\mathbf{H}(-,\mathbf{N}),-)}(t_1,\dots,t_6) = \mathbf{C}e^{\mathbf{A}t_1}\mathbf{H}\left(e^{\mathbf{A}t_2}\mathbf{H}\left(e^{\mathbf{A}t_3}\mathbf{B}\otimes e^{\mathbf{A}t_4}\mathbf{N}e^{\mathbf{A}t_5}\mathbf{B}\right)\otimes e^{\mathbf{A}t_6}\mathbf{B}\right)$$

Summary - QB kernels

- The output of the QB system can be broken into contributions from infinite subsystems.

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}_1(t) + \mathbf{C}\mathbf{x}_2(t) + \dots$$

- Each subsystem has associated with it some finite number of kernels.

$$\mathbf{K}_2 = \{\mathbf{C}e^{\mathbf{A}t_1}\mathbf{N}e^{\mathbf{A}t_2}\mathbf{B}, \mathbf{C}e^{\mathbf{A}t_1}\mathbf{H}(e^{\mathbf{A}t_2}\mathbf{B} \otimes e^{\mathbf{A}t_3}\mathbf{B})\}$$

- There is a way to enumerate and notate the kernels of the QB system.

$$\mathbf{h}_3^{\mathbf{N},\mathbf{N}}(t_1, t_2, t_3) = \mathbf{C}e^{\mathbf{A}t_1}\mathbf{N}e^{\mathbf{A}t_2}\mathbf{N}e^{\mathbf{A}t_3}\mathbf{B}$$

Balanced truncation (BT) for QB systems

Gramians for QB systems

- The QB Gramians are difficult to compute, which is why using truncated Gramians is sometimes advised [Benner/Goyal '17] .

Gramians for QB systems

- The QB Gramians are difficult to compute, which is why using truncated Gramians is sometimes advised [Benner/Goyal '17] .

- The (truncated) Gramians of the QB system $\mathbf{P}_\tau, \mathbf{Q}_\tau$ are defined as,

$$\mathbf{P}_1 = \int_0^\infty e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^\top e^{\mathbf{A}^\top t}dt \qquad \mathbf{Q}_1 = \int_0^\infty e^{\mathbf{A}^\top t}\mathbf{C}^\top \mathbf{C} e^{\mathbf{A}t}dt$$

$$\mathbf{P}_2 = \int_0^\infty e^{\mathbf{A}t}\mathbf{N}\mathbf{P}_1\mathbf{N}^\top e^{\mathbf{A}^\top t}dt \qquad \mathbf{Q}_2 = \int_0^\infty e^{\mathbf{A}^\top t}\mathbf{N}^\top \mathbf{P}_1 \mathbf{N} e^{\mathbf{A}t}dt$$

$$\mathbf{P}_3 = \int_0^\infty e^{\mathbf{A}t}\mathbf{H}(\mathbf{P}_1 \otimes \mathbf{P}_1)\mathbf{H}^\top e^{\mathbf{A}^\top t}dt \qquad \mathbf{Q}_3 = \int_0^\infty e^{\mathbf{A}^\top t}\mathbf{H}^{(2)}(\mathbf{P}_1 \otimes \mathbf{Q}_1)\mathbf{H}^{(2)\top} e^{\mathbf{A}t}dt$$

$$\mathbf{P}_\tau = \mathbf{P}_1 + \mathbf{P}_2 + \mathbf{P}_3 \qquad \mathbf{Q}_\tau = \mathbf{Q}_1 + \mathbf{Q}_2 + \mathbf{Q}_3$$

Gramians for QB systems

- The QB Gramians are difficult to compute, which is why using truncated Gramians is sometimes advised [Benner/Goyal '17] .

- The (truncated) Gramians of the QB system $\mathbf{P}_\tau, \mathbf{Q}_\tau$ are defined as,

$$
\begin{aligned}
\mathbf{P}_1 &= \int_0^\infty e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^\top e^{\mathbf{A}^\top t} dt & \mathbf{Q}_1 &= \int_0^\infty e^{\mathbf{A}^\top t}\mathbf{C}^\top \mathbf{C} e^{\mathbf{A}t} dt \\
\mathbf{P}_2 &= \int_0^\infty e^{\mathbf{A}t}\mathbf{N}\mathbf{P}_1\mathbf{N}^\top e^{\mathbf{A}^\top t} dt & \mathbf{Q}_2 &= \int_0^\infty e^{\mathbf{A}^\top t}\mathbf{N}^\top \mathbf{P}_1 \mathbf{N} e^{\mathbf{A}t} dt \\
\mathbf{P}_3 &= \int_0^\infty e^{\mathbf{A}t}\mathbf{H}(\mathbf{P}_1 \otimes \mathbf{P}_1)\mathbf{H}^\top e^{\mathbf{A}^\top t} dt & \mathbf{Q}_3 &= \int_0^\infty e^{\mathbf{A}^\top t}\mathbf{H}^{(2)}(\mathbf{P}_1 \otimes \mathbf{Q}_1)\mathbf{H}^{(2)\top} e^{\mathbf{A}t} dt \\
\mathbf{P}_\tau &= \mathbf{P}_1 + \mathbf{P}_2 + \mathbf{P}_3 & \mathbf{Q}_\tau &= \mathbf{Q}_1 + \mathbf{Q}_2 + \mathbf{Q}_3
\end{aligned}
$$

- They present the balanced truncation for QB systems using the truncated Gramians.

Classical BT algorithm (QB systems)

- Let $\mathbf{U}, \mathbf{L}$ be square factors of $\mathbf{P}_\tau = \mathbf{U}\mathbf{U}^T$ and $\mathbf{Q}_\tau = \mathbf{L}\mathbf{L}^T$.
- Compute the truncated **SVD** of matrix $\mathbf{L}^T\mathbf{U}$:

$$\mathbf{L}^T\mathbf{U} = \begin{bmatrix} \mathbf{Z}_1 & \mathbf{Z}_2 \end{bmatrix} \begin{bmatrix} \mathbf{S}_1 & \\ & \mathbf{S}_2 \end{bmatrix} \begin{bmatrix} \mathbf{Y}_1^\top \\ \mathbf{Y}_2^\top \end{bmatrix}$$

- Construct the model reduction bases

$$\mathbf{W}_r = \mathbf{L}\mathbf{Z}_1\mathbf{S}_1^{-1/2} \text{ and } \mathbf{V}_r = \mathbf{U}\mathbf{Y}_1\mathbf{S}_1^{-1/2}$$

- The matrices of reduced-order balanced system $\mathbf{\Sigma}_r$ are given by

$$\mathbf{A}_r = \mathbf{W}_r^T\mathbf{A}\mathbf{V}_r = \mathbf{S}_1^{-1/2}\mathbf{Z}_1^T\left(\mathbf{L}^T\mathbf{A}\mathbf{U}\right)\mathbf{Y}_1\mathbf{S}_1^{-1/2},$$

$$\mathbf{B}_r = \mathbf{W}_r^T\mathbf{B} = \mathbf{S}_1^{-1/2}\mathbf{Z}_1^T\left(\mathbf{L}^T\mathbf{B}\right),$$

$$\mathbf{C}_r = \mathbf{C}\mathbf{V}_r = \left(\mathbf{C}\mathbf{U}\right)\mathbf{Y}_1\mathbf{S}_1^{-1/2},$$

$$\mathbf{H}_r = \mathbf{W}_r^T\mathbf{H}(\mathbf{V}_r \otimes \mathbf{V}_r) = \mathbf{S}_1^{-1/2}\mathbf{Z}_1^T\left(\mathbf{L}^T\mathbf{H}(\mathbf{U}\otimes\mathbf{U})\right)\left(\mathbf{Y}_1\mathbf{S}_1^{-1/2}\otimes\mathbf{Y}_1\mathbf{S}_1^{-1/2}\right),$$

$$\mathbf{N}_r = \mathbf{W}_r^T\mathbf{N}\mathbf{V}_r = \mathbf{S}_1^{-1/2}\mathbf{Z}_1^T\left(\mathbf{L}^T\mathbf{N}\mathbf{U}\right)\mathbf{Y}_1\mathbf{S}_1^{-1/2}.$$

QuadBT for QB systems

- Just like in the linear and LQO case, the truncated QB Gramians and their square root factors are approximated as:

$$\widetilde{\mathcal{P}}_1 = \sum_{i=1}^{N_p} \rho_i^2 e^{\mathbf{A}\mu_i}\mathbf{B}\mathbf{B}^\top e^{\mathbf{A}^\top \mu_i} \qquad \widetilde{\mathbf{U}}_1 = \left[\rho_1 e^{\mathbf{A}\mu_1}\mathbf{B} \ldots \rho_{N_p} e^{\mathbf{A}\mu_{N_p}}\mathbf{B}\right]$$

$$\widetilde{\mathcal{P}}_2 = \sum_{i=1}^{N_p} \rho_i^2 e^{\mathbf{A}\mu_i}\mathbf{N}\widetilde{\mathbf{U}}_1\widetilde{\mathbf{U}}_1^\top\mathbf{N}^\top e^{\mathbf{A}^\top \mu_i} \qquad \widetilde{\mathbf{U}}_2 = \left[\rho_1 e^{\mathbf{A}\mu_1}\mathbf{N}\widetilde{\mathbf{U}}_1 \ldots \rho_{N_p} e^{\mathbf{A}\mu_{N_p}}\mathbf{N}\widetilde{\mathbf{U}}_1\right]$$

$$\widetilde{\mathcal{P}}_3 = \sum_{i=1}^{N_p} \rho_i^2 e^{\mathbf{A}\mu_i}\mathbf{H}\left(\widetilde{\mathcal{P}}_1 \otimes \widetilde{\mathcal{P}}_1\right)\mathbf{H}^\top e^{\mathbf{A}\mu_i} \qquad \widetilde{\mathbf{U}}_3 = \left[\rho_1 e^{\mathbf{A}\mu_1}\mathbf{H}\left(\widetilde{\mathbf{U}}_1 \otimes \widetilde{\mathbf{U}}_1\right) \ldots \right.$$

$$\left. \ldots \rho_{N_p} e^{\mathbf{A}\mu_{N_p}}\mathbf{H}\left(\widetilde{\mathbf{U}}_1 \otimes \widetilde{\mathbf{U}}_1\right)\right]$$

- Finally we write $\widetilde{\mathbf{U}} = \left[\widetilde{\mathbf{U}}_1\widetilde{\mathbf{U}}_2\widetilde{\mathbf{U}}_3\right]$.
- A similar calculation follows for $\widetilde{\mathbf{L}}$.
- For simplicity, we assume unity weights and $N_p = N_q = N$.

Intrusive terms to be replaced!

Let $\widetilde{\mathbf{U}}$ and $\widetilde{\mathbf{L}}$ be as discussed above, define the matrix $\widetilde{\mathbb{H}} = \widetilde{\mathbf{L}}^\top \widetilde{\mathbf{U}}$, then we have,

$$\widetilde{\mathbb{H}} = \begin{cases} \mathbf{h}_1(\omega_i + \mu_j), & \text{for } \widetilde{\mathbf{L}}_1^\top \widetilde{\mathbf{U}}_1, i,j = \overline{1,N} \\ \mathbf{h}_2^{\mathbf{N}}(\omega_i + \mu_j, \mu_k), & \text{for } \widetilde{\mathbf{L}}_1^\top \widetilde{\mathbf{U}}_2, i-k = \overline{1,N} \\ \mathbf{h}_2^{\mathbf{H}}(\omega_i + \mu_j, \mu_k, \mu_l), & \text{for } \widetilde{\mathbf{L}}_1^\top \widetilde{\mathbf{U}}_3, i-l = \overline{1,N} \\ \mathbf{h}_2^{\mathbf{N}}(\omega_i, \omega_j + \mu_k), & \text{for } \widetilde{\mathbf{L}}_2^\top \widetilde{\mathbf{U}}_1, i-k = \overline{1,N} \\ \mathbf{h}_3^{\mathbf{N},\mathbf{N}}(\omega_i, \omega_j + \mu_k, \mu_l), & \text{for } \widetilde{\mathbf{L}}_2^\top \widetilde{\mathbf{U}}_2, i-l = \overline{1,N} \\ \mathbf{h}_3^{\mathbf{N},\mathbf{H}}(\omega_i, \omega_j + \mu_k, \mu_l, \mu_m), & \text{for } \widetilde{\mathbf{L}}_2^\top \widetilde{\mathbf{U}}_3, i-m = \overline{1,N} \\ \mathbf{h}_2^{\mathbf{H}}(\omega_i, \mu_k, \omega_j + \mu_l)^\top, & \text{for } \widetilde{\mathbf{L}}_3^\top \widetilde{\mathbf{U}}_1, i-l = \overline{1,N} \\ \mathbf{h}_3^{\mathbf{H}(-,\mathbf{N})}(\omega_i, \mu_k, \omega_j + \mu_l, \mu_m)^\top, & \text{for } \widetilde{\mathbf{L}}_3^\top \widetilde{\mathbf{U}}_2, i-m = \overline{1,N} \\ \mathbf{h}_3^{\mathbf{H}(-,\mathbf{H})}(\omega_i, \mu_k, \omega_j + \mu_l, \mu_m, \omega_n)^\top, & \text{for } \widetilde{\mathbf{L}}_3^\top \widetilde{\mathbf{U}}_3, i-n = \overline{1,N} \end{cases}$$

$\widetilde{\mathbb{H}}$ has the structure of a Hankel matrix (see [Antoulas '05]) given by

$$\widetilde{\mathbb{H}} = \begin{bmatrix} \widetilde{\mathbf{L}}_1^\top \widetilde{\mathbf{U}}_1 & \widetilde{\mathbf{L}}_1^\top \widetilde{\mathbf{U}}_2 & \widetilde{\mathbf{L}}_1^\top \widetilde{\mathbf{U}}_3 \\ \widetilde{\mathbf{L}}_2^\top \widetilde{\mathbf{U}}_1 & \widetilde{\mathbf{L}}_2^\top \widetilde{\mathbf{U}}_2 & \widetilde{\mathbf{L}}_2^\top \widetilde{\mathbf{U}}_3 \\ \widetilde{\mathbf{L}}_3^\top \widetilde{\mathbf{U}}_1 & \widetilde{\mathbf{L}}_3^\top \widetilde{\mathbf{U}}_2 & \widetilde{\mathbf{L}}_3^\top \widetilde{\mathbf{U}}_3 \end{bmatrix} \in \mathbb{R}^{(N+N^2+N^3)\times(N+N^2+N^3)}$$

It's all data!

- Construct the quadrature-based approximations:

$$\widetilde{\mathbb{M}} = \widetilde{\mathbf{L}}^T \mathbf{A} \widetilde{\mathbf{U}} \approx \mathbf{L}^\top \mathbf{A} \mathbf{U}$$

$$\widetilde{\mathbb{N}} = \widetilde{\mathbf{L}}^T \mathbf{N} \widetilde{\mathbf{U}} \approx \mathbf{L}^\top \mathbf{N} \mathbf{U}$$

$$\widetilde{\mathbb{K}} = \widetilde{\mathbf{L}}^T \mathbf{H}(\widetilde{\mathbf{U}} \otimes \widetilde{\mathbf{U}}) \approx \mathbf{L}^T \mathbf{H}(\mathbf{U} \otimes \mathbf{U})$$

$$\widetilde{\mathbb{h}} = \widetilde{\mathbf{L}}^\top \mathbf{B} \approx \mathbf{L}^\top \mathbf{B}$$

$$\widetilde{\mathbb{g}} = \mathbf{C} \widetilde{\mathbf{U}} \approx \mathbf{C} \mathbf{U}$$

It's all data!

- Construct the quadrature-based approximations:

$$\widetilde{\mathbb{M}} = \widetilde{\mathbf{L}}^T \mathbf{A} \widetilde{\mathbf{U}} \approx \mathbf{L}^\top \mathbf{A} \mathbf{U}$$
$$\widetilde{\mathbb{N}} = \widetilde{\mathbf{L}}^T \mathbf{N} \widetilde{\mathbf{U}} \approx \mathbf{L}^\top \mathbf{N} \mathbf{U}$$
$$\widetilde{\mathbb{K}} = \widetilde{\mathbf{L}}^T \mathbf{H} (\widetilde{\mathbf{U}} \otimes \widetilde{\mathbf{U}}) \approx \mathbf{L}^T \mathbf{H} (\mathbf{U} \otimes \mathbf{U})$$
$$\widetilde{\mathbb{h}} = \widetilde{\mathbf{L}}^\top \mathbf{B} \approx \mathbf{L}^\top \mathbf{B}$$
$$\widetilde{\mathbb{g}} = \mathbf{C} \widetilde{\mathbf{U}} \approx \mathbf{C} \mathbf{U}$$

- The entries of each of these matrices $\widetilde{\mathbb{M}}, \widetilde{\mathbb{N}}, \widetilde{\mathbb{K}}, \widetilde{\mathbb{h}}$ and $\widetilde{\mathbb{g}}$ can be written in terms of the samples of QB kernels.

It's all data!

- Construct the quadrature-based approximations:

$$\widetilde{\mathbb{M}} = \widetilde{\mathbf{L}}^T \mathbf{A} \widetilde{\mathbf{U}} \approx \mathbf{L}^\top \mathbf{A} \mathbf{U}$$

$$\widetilde{\mathbb{N}} = \widetilde{\mathbf{L}}^T \mathbf{N} \widetilde{\mathbf{U}} \approx \mathbf{L}^\top \mathbf{N} \mathbf{U}$$

$$\widetilde{\mathbb{K}} = \widetilde{\mathbf{L}}^T \mathbf{H} (\widetilde{\mathbf{U}} \otimes \widetilde{\mathbf{U}}) \approx \mathbf{L}^T \mathbf{H} (\mathbf{U} \otimes \mathbf{U})$$

$$\widetilde{\mathbb{h}} = \widetilde{\mathbf{L}}^\top \mathbf{B} \approx \mathbf{L}^\top \mathbf{B}$$

$$\widetilde{\mathbb{g}} = \mathbf{C} \widetilde{\mathbf{U}} \approx \mathbf{C} \mathbf{U}$$

- The entries of each of these matrices $\widetilde{\mathbb{M}}$, $\widetilde{\mathbb{N}}$, $\widetilde{\mathbb{K}}$, $\widetilde{\mathbb{h}}$ and $\widetilde{\mathbb{g}}$ can be written in terms of the samples of QB kernels.
- We are finally ready to present the Quad BT algorithm for QB systems.

QuadBT algorithm for QB systems

- Put together the data matrices:

$$\widetilde{\mathbb{H}},\quad \widetilde{\mathbb{K}},\quad \widetilde{\mathbb{N}},\quad \widetilde{\mathbb{M}},\quad \widetilde{\mathbb{h}},\quad \widetilde{\mathbb{g}}.$$

- Compute the truncated **SVD** of matrix $\widetilde{\mathbb{H}}$:

$$\widetilde{\mathbb{H}} = \begin{bmatrix} \widetilde{\mathbf{Z}}_1 & \widetilde{\mathbf{Z}}_2 \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{S}}_1 & \\ & \widetilde{\mathbf{S}}_2 \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{Y}}_1^\top \\ \widetilde{\mathbf{Y}}_2^\top \end{bmatrix}.$$

- The matrices of the reduced-order data-driven system $\mathbf{\Sigma}_r$ are given by

$$\begin{aligned}
\tilde{\mathbf{A}}_r &= \cancel{\mathbf{W}_r^T \mathbf{A} \mathbf{V}_r} = \widetilde{\mathbf{S}}_1^{-1/2} \widetilde{\mathbf{Z}}_1^\top \left(\widetilde{\mathbb{M}}\right) \widetilde{\mathbf{Y}}_1 \widetilde{\mathbf{S}}_1^{-1/2},\\
\tilde{\mathbf{B}}_r &= \cancel{\mathbf{W}_r^T \mathbf{B}} = \mathbf{S}_1^{-1/2} \mathbf{Z}_1^\top \left(\widetilde{\mathbb{g}}\right),\\
\tilde{\mathbf{C}}_r &= \cancel{\mathbf{C}\mathbf{V}_r} = \left(\widetilde{\mathbb{h}}\right) \widetilde{\mathbf{Y}}_1 \widetilde{\mathbf{S}}_1^{-1/2}.\\
\tilde{\mathbf{N}}_r &= \cancel{\mathbf{L}^\top \mathbf{N} \mathbf{U}} = \mathbf{S}_1^{-1/2} \mathbf{Z}_1^\top (\widetilde{\mathbb{N}}) \widetilde{\mathbf{Y}}_1 \widetilde{\mathbf{S}}_1^{-1/2}\\
\tilde{\mathbf{H}}_r &= \cancel{\mathbf{L}^\top \mathbf{H}(\mathbf{U} \otimes \mathbf{U})} = \mathbf{S}_1^{-1/2} \mathbf{Z}_1^\top (\widetilde{\mathbb{K}}) (\widetilde{\mathbf{Y}}_1 \widetilde{\mathbf{S}}_1^{-1/2} \otimes \widetilde{\mathbf{Y}}_1 \widetilde{\mathbf{S}}_1^{-1/2})
\end{aligned}$$

Are we done? Not quite.

1. How to avoid forming the Kronecker product $\widetilde{\mathbf{Y}}_1\widetilde{\mathbf{S}}_1^{-1/2} \otimes \widetilde{\mathbf{Y}}_1\widetilde{\mathbf{S}}_1^{-1/2}$ appearing in the expression for $\tilde{\mathbf{H}}_r$?

Are we done? Not quite.

1. How to avoid forming the Kronecker product $\widetilde{\mathbf{Y}}_1\widetilde{\mathbf{S}}_1^{-1/2} \otimes \widetilde{\mathbf{Y}}_1\widetilde{\mathbf{S}}_1^{-1/2}$ appearing in the expression for $\widetilde{\mathbf{H}}_r$?

   [Benner/Breiten '15] address this problem and provide ways to compute this without explicitly forming the Kronecker product.

Are we done? Not quite.

1. How to avoid forming the Kronecker product $\widetilde{\mathbf{Y}}_1\widetilde{\mathbf{S}}_1^{-1/2} \otimes \widetilde{\mathbf{Y}}_1\widetilde{\mathbf{S}}_1^{-1/2}$ appearing in the expression for $\tilde{\mathbf{H}}_r$?

   [Benner/Breiten '15] address this problem and provide ways to compute this without explicitly forming the Kronecker product.

2. The size of $\widetilde{\mathbb{H}}$ grows cubically with N which makes calculating SVD of $\widetilde{\mathbb{H}}$ difficult.

Are we done? Not quite.

1. How to avoid forming the Kronecker product $\widetilde{\mathbf{Y}}_1\widetilde{\mathbf{S}}_1^{-1/2} \otimes \widetilde{\mathbf{Y}}_1\widetilde{\mathbf{S}}_1^{-1/2}$ appearing in the expression for $\tilde{\mathbf{H}}_r$?

   [Benner/Breiten '15] address this problem and provide ways to compute this without explicitly forming the Kronecker product.

2. The size of $\widetilde{\mathbb{H}}$ grows cubically with N which makes calculating SVD of $\widetilde{\mathbb{H}}$ difficult.

   To deal with this, we consider replacing $\widetilde{\mathbf{U}}_1$ with a fewer node approximation $\widehat{\widetilde{\mathbf{U}}}_1$ in the expression for $\widetilde{\mathbf{U}}_2$ and $\widetilde{\mathbf{U}}_3$.

Are we done? Not quite.

1. How to avoid forming the Kronecker product $\widetilde{\mathbf{Y}}_1\widetilde{\mathbf{S}}_1^{-1/2} \otimes \widetilde{\mathbf{Y}}_1\widetilde{\mathbf{S}}_1^{-1/2}$ appearing in the expression for $\tilde{\mathbf{H}}_r$?

   [Benner/Breiten '15] address this problem and provide ways to compute this without explicitly forming the Kronecker product.

2. The size of $\widetilde{\mathbb{H}}$ grows cubically with N which makes calculating SVD of $\widetilde{\mathbb{H}}$ difficult.

   To deal with this, we consider replacing $\widetilde{\mathbf{U}}_1$ with a fewer node approximation $\widehat{\widetilde{\mathbf{U}}}_1$ in the expression for $\widetilde{\mathbf{U}}_2$ and $\widetilde{\mathbf{U}}_3$.

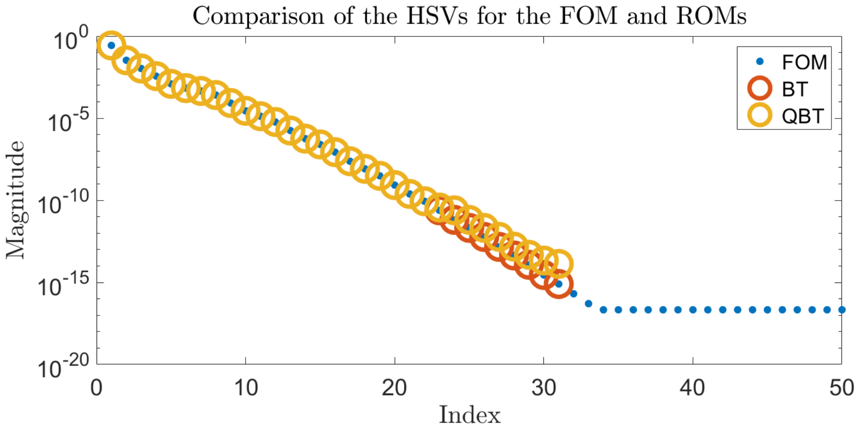
$$\widetilde{\mathbf{U}}_2 = \left[\rho_1 e^{\mathbf{A}\mu_1}\mathbf{N}\cancel{\widetilde{\mathbf{U}}_1} \dots \rho_{N_p} e^{\mathbf{A}\mu_{N_p}}\mathbf{N}\cancel{\widetilde{\mathbf{U}}_1}\right] \in \mathbb{R}^{n \times N_p^2}$$

$$\widehat{\widetilde{\mathbf{U}}}_2 = \left[\rho_1 e^{\mathbf{A}\mu_1}\mathbf{N}\widehat{\widetilde{\mathbf{U}}}_1 \dots \rho_{N_p} e^{\mathbf{A}\mu_{N_p}}\mathbf{N}\widehat{\widetilde{\mathbf{U}}}_1\right] \in \mathbb{R}^{n \times N_p n_1}$$

   where $n_1 << N_p$.

Numerical results (Burgers Quadratic)

- FOM = Burgers quadratic equation with (n=150)
- Construct ROMs of dimension $r = 31$ using **BT** and **QuadBT** .



Conclusions and Outlook

- Successfully extended Quad BT to QB systems
- MIMO case follows immediately with the only difference being that the kernels are no longer scalars but matrices in $\mathbb{R}^{m \times p}$
- Adaptive schemes to pick quadrature nodes cleverly would be something interesting to try