Operator inference for non-polynomial systems & control applications[1]

Boris Krämer

ICERM Workshop on Mathematics of Reduced Order Models
February 17-21, 2020, Providence, RI

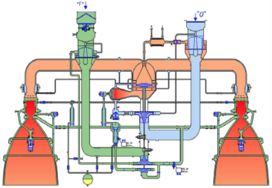
UC San Diego
JACOBS SCHOOL OF ENGINEERING
Mechanical and Aerospace Engineering

[1] Funded by: DARPA EQUiPS program award number UTA15-001067 and Air Force Center of Excellence on Multi-Fidelity Modeling of Rocket Combustor Dynamics, award FA9550-17-1-0195

Motivation: Learning ROMs for complex systems

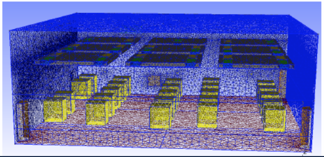
Challenges in complex applications

- Uncertainties in model, parameters, or both
- Details and access to governing equations, discretization, and solver typically unavailable when working with legacy codes $\Rightarrow$ Intrusive MOR infeasible in those situations



Opportunities

- Data is everywhere (cheaper memory, better sensors, more observations); can be used to build ROMs and/or reduce uncertainties



Starting comments

Dynamical systems nature of problem should not be ignored in model learning.

The more we know about the model, the more we can incorporate into the learning framework: Model structure, nonlinear terms, inputs, etc.

Partial Differential Equation Model

Many complex problems are modeled with PDEs of the form:

$$\frac{\partial s}{\partial t} = \mathcal{A}(s;q)+\mathcal{H}(s;q)+f(t,s;q)+\mathcal{B}(\mathbf{u};q)$$

- Input $\mathbf{u}(t)$
- State $s(x,t;\mu)$ with $x \in \Omega \subseteq \mathbf{R}^d,\ d=1,2,3$
- Parameters q
- $\mathcal{A}$ is a linear operator
- $\mathcal{H}$ is quadratic in s, the nonlinear function is $f(t,s)$ and $\mathcal{B}$ is a linear input operator.

Full-order model (FOM)

Semi-discretized numerical model of the PDE:

$$\begin{aligned}\dot{\mathbf{s}}(t,\mathbf{q}) =& \mathbf{A}(\mathbf{q})\mathbf{s}(t;\mathbf{q}) + \mathbf{H}(\mathbf{q})(\mathbf{s}(t;\mathbf{q}) \otimes \mathbf{s}(t;\mathbf{q})) \\ &+ \mathbf{f}(t,\mathbf{s};\mathbf{q}) + \mathbf{B}(\mathbf{q})\mathbf{u}(t),\end{aligned}$$

- State $\mathbf{s}(t;\mathbf{q}) \in \mathbf{R}^n$
- Parameters $\mathbf{q} \in \mathbf{R}^\ell$
- Matrices $\mathbf{A} \in \mathbf{R}^{n\times n}$, $\mathbf{H} \in \mathbf{R}^{n\times n^2}$, $\mathbf{B} \in \mathbf{R}^{n\times m}$

Reduced-order model (ROM)

$$\begin{aligned}\dot{\hat{\mathbf{s}}}(t;\mathbf{q}) =& \hat{\mathbf{A}}(\mathbf{q})\hat{\mathbf{s}}(t;\mathbf{q}) + \hat{\mathbf{H}}(\mathbf{q})(\hat{\mathbf{s}}(t;\mathbf{q}) \otimes \hat{\mathbf{s}}(t;\mathbf{q})) \\ &+ \mathbf{V}^\top \mathbf{f}(t,\mathbf{V}\hat{\mathbf{s}}(\mathbf{q})) + \hat{\mathbf{B}}(\mathbf{q})\mathbf{u}(t)\end{aligned}$$

- Reduced state $\widehat{\mathbf{s}}(t;\mathbf{q}) \in \mathbf{R}^n$
- Parameters $\mathbf{q} \in \mathbf{R}^\ell$
- Matrices $\hat{\mathbf{A}} \in \mathbf{R}^{r\times r}$, $\hat{\mathbf{H}} \in \mathbf{R}^{r\times r^2}$, $\hat{\mathbf{B}} \in \mathbf{R}^{r\times m}$

Intrusive vs. non-intrusive: Sometimes we don't have a choice

PDE

$$\frac{\partial s}{\partial t} = \mathcal{A}(s;q) + \mathcal{H}(s;q) + f(t,s;q) + \mathcal{B}(\mathbf{u};q)$$

FOM

$$\begin{aligned}\dot{\mathbf{s}}(t,\mathbf{q}) =& \mathbf{A}(\mathbf{q})\mathbf{s}(t;\mathbf{q}) + \mathbf{H}(\mathbf{q})(\mathbf{s}(t;\mathbf{q}) \otimes \mathbf{s}(t;\mathbf{q})) \\ &+ \mathbf{f}(t,\mathbf{s};\mathbf{q}) + \mathbf{B}(\mathbf{q})\mathbf{u}(t),\end{aligned}$$

DATA

$\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_k,$ and $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k$

ROM

$$\begin{aligned}\dot{\hat{\mathbf{s}}}(t;\mathbf{q}) =& \hat{\mathbf{A}}(\mathbf{q})\hat{\mathbf{s}}(t;\mathbf{q}) + \hat{\mathbf{H}}(\mathbf{q})(\hat{\mathbf{s}}(t;\mathbf{q}) \otimes \hat{\mathbf{s}}(t;\mathbf{q})) \\ &+ \mathbf{V}^\top \mathbf{f}(t, \mathbf{V}\hat{\mathbf{s}}(\mathbf{q})) + \hat{\mathbf{B}}(\mathbf{q})\mathbf{u}(t)\end{aligned}$$

Part 1:

Operator Inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms

(hopefully completed soon)

with Peter Benner, Pawan Goyal, Benjamin Peherstorfer, Karen Willcox

Part 1: Problem setting & Goal

1. **Model is known in PDE form (quadratic + non-polynomial + input):**

$$\frac{\partial s}{\partial t} = \mathcal{A}(s) + \mathcal{H}(s) + f(t,s) + \mathcal{B}(\mathbf{u})$$

2. **Data available from FOM[2]:**

$$\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k, \quad \text{and} \quad \mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$$

3. **The non-polynomial nonlinear term is such that:**

$$\mathbf{f}(t,\mathbf{s}) = [f(t,s_1), \ \cdots \ , \ f(t,s_n)]^\top$$

Goal

Leverage available information of nonlinear terms to learn a ROM:

$$\dot{\hat{\mathbf{s}}}(t) = \hat{\mathbf{A}}\hat{\mathbf{s}}(t) + \hat{\mathbf{H}}(\hat{\mathbf{s}}(t) \otimes' \hat{\mathbf{s}}(t)) + \mathbf{V}^\top \mathbf{f}(t, \mathbf{V}\hat{\mathbf{s}}) + \hat{\mathbf{B}}\mathbf{u}(t)$$

[2] The FOM data comes from time-stepping $\dot{\mathbf{s}}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{H}(\mathbf{s}(t) \otimes' \mathbf{s}(t)) + \mathbf{f}(t,\mathbf{s}) + \mathbf{B}\mathbf{u}(t)$

A short example

Consider the PDE

$$\frac{\partial}{\partial t}s(x,t) = \frac{\partial^2}{\partial x^2}s(x,t) + e^{-\beta t}s(x,t)^{-\alpha} + b(x)u(t)$$

with time-dependent reaction term $f(t,s) = e^{-\beta t}s(x,t)^{-\alpha}$.
After spatial discretization with a finite difference scheme, the system reads as $(\mathbf{s}^{-\alpha} := [s_i]^{-\alpha},\ i = 1,2,\dots,n)$:

$$\dot{\mathbf{s}}(t) = \mathbf{A}\mathbf{s}(t) + e^{-\beta t}\mathbf{s}(t)^{-\alpha} + \mathbf{B}u(t).$$

- $\mathbf{f}(t,\mathbf{s}(t)) = e^{-\beta t}\mathbf{s}(t)^{-\alpha}$ does not require approximation of spatial derivatives
- Evaluating the semi-discrete nonlinear function $\mathbf{f}(t,\mathbf{s})$ only requires application of $f(t,s_i)$ at every component of $\mathbf{s} = [s_1, s_2, \dots, s_n]$.
- **Other examples**: Arrhenius reaction model $\exp\left(\gamma - \frac{\gamma}{s}\right)$; rational functions $\dfrac{s}{\alpha + s}$, fractional powers s^α etc.

Intrusive projection-based ROMs

Given is the following FOM, and we would like to compute a ROM:

$$\dot{\mathbf{s}}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{H}(\mathbf{s}(t) \otimes' \mathbf{s}(t)) + \mathbf{f}(t, \mathbf{s}) + \mathbf{B}\mathbf{u}(t)$$

- $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_r] \in \mathbf{R}^{n \times r}$ orthonormal matrix, $r \ll n$, computed, e.g., with POD.
- Let $\tilde{\mathbf{s}}$ be the ROM state with $\mathbf{s}_i \approx \mathbf{V}\tilde{\mathbf{s}}_i$

Projection-based (intrusive) ROM

The projection-based ROM has the form

$$\dot{\widetilde{\mathbf{s}}}(t) = \widetilde{\mathbf{A}}\widetilde{\mathbf{s}}(t) + \widetilde{\mathbf{H}}(\widetilde{\mathbf{s}}(t) \otimes' \widetilde{\mathbf{s}}(t)) + \widetilde{\mathbf{f}}(t, \widetilde{\mathbf{s}}) + \widetilde{\mathbf{B}}\mathbf{u}(t)$$

where $\widetilde{\mathbf{f}}(t, \widetilde{\mathbf{s}}) = \mathbf{V}^\top \mathbf{f}(t, \mathbf{V}\widetilde{\mathbf{s}})$ and the reduced operators

$$\widetilde{\mathbf{A}} = \mathbf{V}^\top \mathbf{A}\mathbf{V} \in \mathbf{R}^{r \times r}, \quad \widetilde{\mathbf{H}} = \mathbf{V}^\top \mathbf{H}(\mathbf{V} \otimes' \mathbf{V}) \in \mathbf{R}^{r \times r^2}, \quad \widetilde{\mathbf{B}} = \mathbf{V}^\top \mathbf{B} \in \mathbf{R}^{r \times m}$$

Non-intrusive Operator Inference: Preparing the data

To learn a ROM, we build on the operator inference work from [Peherstorfer and Willcox, 2016]:

1. Start with state and input data:

$$\mathbf{S} := \begin{bmatrix} | & | & & | \\ \mathbf{s}_0 & \mathbf{s}_1 & \cdots & \mathbf{s}_k \\ | & | & & | \end{bmatrix}, \qquad \mathbf{U} := \begin{bmatrix} | & | & & | \\ \mathbf{u}(t_0) & \mathbf{u}(t_1) & \cdots & \mathbf{u}(t_k) \\ | & | & & | \end{bmatrix}.$$

2. Due to the specific form of the nonlinear terms, we can evaluate the nonlinear snapshot matrix:

$$\mathbf{F} = \begin{bmatrix} | & | & & | \\ \mathbf{f}(t_0, \mathbf{s}(t_0)) & \mathbf{f}(t_1, \mathbf{s}(t_1)) & \cdots & \mathbf{f}(t_k, \mathbf{s}(t_k)) \\ | & | & & | \end{bmatrix}.$$

3. Compute r dominant POD basis vectors of $\mathbf{S}$, resulting in $\mathbf{V}$ s.t. $\dfrac{\|\mathbf{S} - \mathbf{V}\mathbf{V}^\top \mathbf{S}\|}{\|\mathbf{S}\|} \leq \texttt{tol}$.

4. Project the state data and nolinear snapshot data

$$\hat{\mathbf{S}} = \mathbf{V}^\top \mathbf{S}, \qquad \hat{\mathbf{F}} = \mathbf{V}^\top \mathbf{F}.$$

Operator inference: Solving for the operators

Denote with $\dot{\hat{\mathrm{s}}}_k$ the time derivative approximation of $\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathrm{s}}(t_k)$, which can be computed from $\hat{\mathrm{s}}$ using a time derivative approximation. We store the time-derivative approximations in the matrix

$$\dot{\hat{\mathbf{S}}} := \begin{bmatrix} | & | & & | \\ \dot{\hat{\mathbf{s}}}(t_0) & \dot{\hat{\mathbf{s}}}(t_1) & \cdots & \dot{\hat{\mathbf{s}}}(t_k) \\ | & | & & | \end{bmatrix}.$$

Operator inference for non-polynomial nonlinear system

A non-intrusive ROM of the form

$$\dot{\hat{\mathbf{s}}}(t) = \hat{\mathbf{A}}\hat{\mathbf{s}}(t) + \hat{\mathbf{H}}(\hat{\mathbf{s}}(t) \otimes' \hat{\mathbf{s}}(t)) + \mathbf{V}^\top \mathbf{f}(t, \mathbf{V}\hat{\mathbf{s}}(t)) + \hat{\mathbf{B}}\mathbf{u}(t),$$

can be obtained by solving the optimization problem from the above projected:

$$\min_{\hat{\mathbf{A}},\hat{\mathbf{B}},\hat{\mathbf{H}}} \|\underbrace{\dot{\hat{\mathbf{S}}} - \hat{\mathbf{F}}}_{:=\hat{\mathbf{R}}} - \hat{\mathbf{A}}\hat{\mathbf{S}} - \hat{\mathbf{B}}\mathbf{U} - \hat{\mathbf{H}}(\hat{\mathbf{S}} \otimes' \hat{\mathbf{S}})\|_F.$$

Two assumptions needed for convergence analysis

Assumption 1

The time stepping scheme for the FOM is convergent, i.e.,

$$\max_{i \in \{1, \dots T/\Delta t\}} \|\mathbf{s}_i - \mathbf{s}(t_i)\|_2 \to 0 \quad \text{as} \quad \Delta t \to 0.$$

Assumption 2

The derivatives approximated from projected states, $\dot{\hat{\mathbf{s}}}_k$, converge to $\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{s}}(t_k)$ as the discretization time step $\Delta t \to 0$, i.e.,

$$\max_{i \in \{1, \dots T/\Delta t\}} \|\dot{\hat{\mathbf{s}}}_i - \frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{s}}(t_i)\|_2 \to 0 \quad \text{as} \quad \Delta t \to 0.$$

Convergence of the learned ROM to the intrusive ROM

Theorem [Benner/Goyal/K./Peherstorfer/Willcox, 2020]

Let Assumption 1 and Assumption 2 hold and let a POD basis matrix $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r] \in \mathbf{R}^{n\times r}$ be given. Let $\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}, \widetilde{\mathbf{H}}$ be the intrusively projected ROM operators. Let the data matrix $[\hat{\mathbf{S}},\ \mathbf{U},\ \hat{\mathbf{S}} \otimes' \hat{\mathbf{S}}]$ have full column rank.

Then for every $\varepsilon > 0$, there exists $r \leq n$ and a time step size $\Delta t > 0$ such that the learned operators satisfy:

$$\|\hat{\mathbf{A}} - \widetilde{\mathbf{A}}\| \leq \varepsilon, \qquad \|\hat{\mathbf{B}} - \widetilde{\mathbf{B}}\| \leq \varepsilon, \qquad \|\hat{\mathbf{H}} - \widetilde{\mathbf{H}}\| \leq \varepsilon.$$

Hyper-reduction to speed up nonlinear function evaluation

To accelerate the evaluation of $\mathbf{V}^\top \mathbf{f}(t, \mathbf{V}\hat{\mathbf{s}})$ in the learned ROM, hyper-reduction can be used (Barrault et al., 2004; Astrid et al., 2008; Nguyen et al., 2008, Chaturantabut & Sorensen, 2010; Carlberg et al., 2013; Drmac & Gugercin, 2016,...)

We employ the discrete empirical interpolation method (DEIM) to approximate

$$\hat{\mathbf{f}}(t, \mathbf{V}\hat{\mathbf{s}}) \approx \hat{\mathbf{f}}_r(t, \mathbf{V}\hat{\mathbf{s}}) = \mathbf{V}^\top \mathbf{W}(\mathbb{S}^\top \mathbf{W})^{-1}\mathbb{S}^\top \mathbf{f}(t, \mathbf{V}\hat{\mathbf{s}}).$$

- $\mathbf{W}$ is computed by taking the SVD of the nonlinear snapshot matrix $\mathbf{F}$ and setting $\mathbf{W}$ to the leading m left singular vectors of $\mathbf{F}$.
- $\mathbb{S}$ is an $n \times m$ matrix obtained by selecting certain columns of the $n \times n$ identity matrix, following the DEIM algorithm.

Results: Tubular reactor model

One-dimensional ($x \in (0,1)$) model with a single reaction, describing the evolution of the **species concentration** $\psi(x,t)$ and **temperature** $\theta(x,t)$ via

$$\frac{\partial \psi}{\partial t} = \frac{1}{\mathrm{Pe}} \frac{\partial^2 \psi}{\partial x^2} - \frac{\partial \psi}{\partial x} - \mathcal{D} f(\psi, \theta; \gamma),$$
$$\frac{\partial \theta}{\partial t} = \frac{1}{\mathrm{Pe}} \frac{\partial^2 \theta}{\partial x^2} - \frac{\partial \theta}{\partial x} - \beta(\theta - \theta_{\text{ref}}) + \mathcal{B}\mathcal{D} f(\psi, \theta; \gamma),$$

with **Arrhenius reaction term**

$$f(\psi, \theta; \gamma) = \psi \exp\left(\gamma - \frac{\gamma}{\theta}\right).$$

Boundary conditions

$$\frac{\partial \psi}{\partial x}(0,t) = \mathrm{Pe}(\psi(0,t) - 1), \quad \frac{\partial \theta}{\partial x}(0,t) = \mathrm{Pe}(\theta(0,t) - 1), \quad \frac{\partial \psi}{\partial x}(1,t) = 0, \quad \frac{\partial \theta}{\partial x}(1,t) = 0.$$

The **quantity of interest** is the temperature oscillation at the reactor exit:

$$y(t) = \theta(x = 1, t).$$

Results: Tubular reactor model

- FOM is semi-discrete model obtained via finite differences:

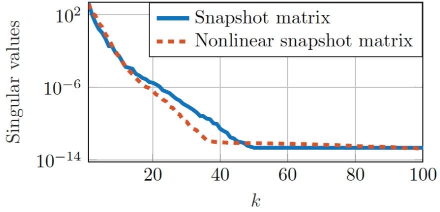
$$\dot{\mathbf{s}}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{f}(\mathbf{s}(t)) + \mathbf{B}.$$

with discretized state $\mathbf{s}(t) \in \mathbb{R}^{198}$.

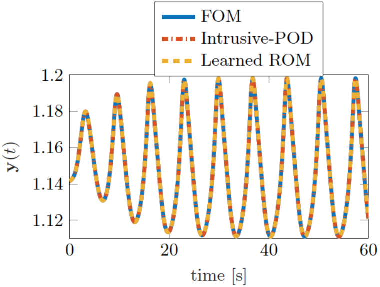
- Discretized Arrhenius term requires pointwise evaluations (local in space)

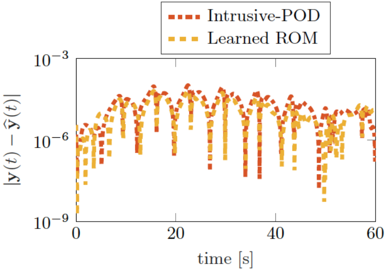
$$[\mathbf{f}(\boldsymbol{\psi}, \boldsymbol{\theta}; \gamma)]_i = \psi_i \exp\left(\gamma - \frac{\gamma}{\theta_i}\right).$$

- Collect snapshots in $T = (0, 30]$ with $\delta t = 10^{-3}$ spacing.

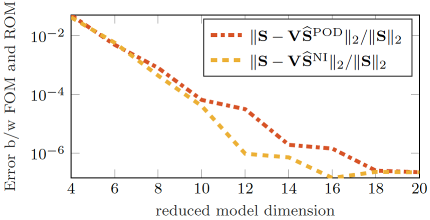


Learned ROM more accurate than intrusive ROM



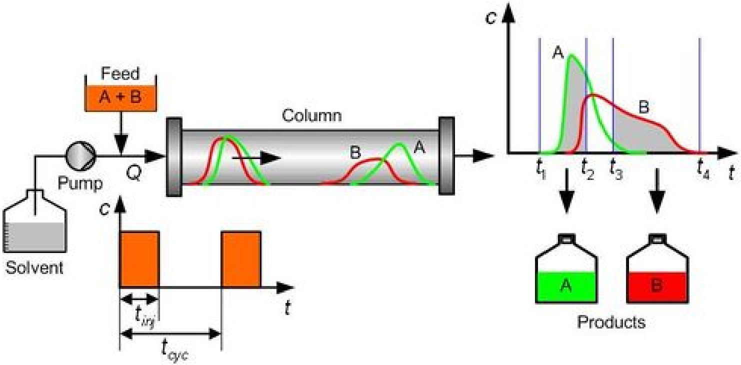


- ROM predictions for 100% longer than training interval
- $r = 10$ for both ROMs
- Learned ROM (non-Markovian) more accurate than projection-based ROM



Batch Chromatography: A chemical process

- Mixture of products A and B injected into column
- Move with different velocities, thus separating at the column exit
- Component A, which moves faster, is collected between t_1 and t_2, component B between t_3 and t_4



Batch Chromatography: A non-polynomial nonlinear model

The dynamics of the batch chromatographic column:

$$
\begin{aligned}
\frac{\partial c_i}{\partial t} + \frac{1-\epsilon}{\epsilon}\frac{\partial q_i}{\partial t} + \nu \frac{\partial c_i}{\partial x} - D_i \frac{\partial^2 c_i}{\partial x^2} &= 0 \\
\frac{\partial q_i}{\partial t} &= \kappa_i \left(q_i^{Eq} - q_i \right)
\end{aligned}
$$

- Adsorption equilibrium concentration

$$
q_i^{\mathsf{Eq}} = \frac{H_{i,1}\, c_i}{1 + \sum_{j=1,2} K_{j,1}\, c_j} + \frac{H_{i,2}\, c_i}{1 + \sum_{j=1,2} K_{j,2}\, c_j}
$$

- c_i, q_i: Liquid & solid phase concentration
- $H_{i,1}, H_{i,2}$: Henry constants
- $K_{j,1}, K_{j,2}$ the thermodynamic coefficients
- κ_i: mass-transfer coefficient of component i
- ϵ: column porosity

- Zero initial conditions
  $c_i(t=0,x) = q_i(t=0,x) = 0$
- Boundary conditions:

$$
D_i \left.\frac{\partial c_i}{\partial x}\right|_{x=0} = \nu \left(c_i|_{x=0} - c_i^{\mathsf{in}} \right), \qquad \left.\frac{\partial c_i}{\partial x}\right|_{x=L} = 0
$$

with

$$
c_i^{\mathsf{in}}(t) = \frac{1}{1 + e^{-5(t - t_{\mathsf{inj}})}}, \qquad t_{\mathsf{inj}} = 1.3
$$

Creating a block-structured model

We can simplify the previous PDE by inserting $\dot{q}_i$ into the first equation, and obtain:

$$
\begin{aligned}
\frac{\partial c_i}{\partial t} &= -\nu \frac{\partial c_i}{\partial x} + D_i \frac{\partial^2 c_i}{\partial x^2} + \epsilon_c \kappa_i \left(q_i^{Eq} - q_i \right) \\
\frac{\partial q_i}{\partial t} &= \kappa_i \left(q_i^{\mathsf{Eq}} - q_i \right)
\end{aligned}
$$

A finite volume discretization of the governing equations yields a discretized model of the form:

$$
\begin{bmatrix} \dot{\mathbf{c}}_1 \\ \dot{\mathbf{q}}_1 \\ \dot{\mathbf{c}}_2 \\ \dot{\mathbf{q}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{A}_2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{q}_1 \\ \mathbf{c}_2 \\ \mathbf{q}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \\ \mathbf{B} \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} \epsilon_c \\ 1 \end{bmatrix} \otimes \begin{bmatrix} \mathbf{f}_1(\mathbf{c}_1, \mathbf{q}_1, \mathbf{c}_2, \mathbf{q}_2) \\ \mathbf{f}_2(\mathbf{c}_1, \mathbf{q}_1, \mathbf{c}_2, \mathbf{q}_2) \end{bmatrix},
$$

where $\mathbf{c}_1, \mathbf{q}_1, \mathbf{c}_2, \mathbf{q}_2 \in \mathbf{R}^n$, $\mathbf{A}_1, \mathbf{A}_2 \in \mathbf{R}^{n \times n}$, $\mathbf{B} \in \mathbf{R}^n$. The nonlinear term is spatially local

$$
\mathbf{f}_i(\mathbf{c}_1, \mathbf{q}_1, \mathbf{c}_2, \mathbf{q}_2) = \kappa_i \left(\mathbf{q}_i^{\mathsf{Eq}} - \mathbf{q}_i \right)
$$

Maintaining the coupling structure in projection-based ROMs

In a projection-based framework, we would approximate

$$\begin{bmatrix} \mathbf{c}_1 \\ \mathbf{q}_1 \\ \mathbf{c}_2 \\ \mathbf{q}_2 \end{bmatrix} \approx \begin{bmatrix} \mathbf{V}_{\mathbf{c}_1} & & & \\ & \mathbf{V}_{\mathbf{q}_1} & & \\ & & \mathbf{V}_{\mathbf{c}_2} & \\ & & & \mathbf{V}_{\mathbf{q}_2} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{c}}_1 \\ \hat{\mathbf{q}}_1 \\ \hat{\mathbf{c}}_2 \\ \hat{\mathbf{q}}_2 \end{bmatrix}$$

so that the ROM preserves the coupling structure:

$$\begin{bmatrix} \dot{\hat{\mathbf{c}}}_1 \\ \dot{\hat{\mathbf{q}}}_1 \\ \dot{\hat{\mathbf{c}}}_2 \\ \dot{\hat{\mathbf{q}}}_2 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{A}}_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \hat{\mathbf{A}}_2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{c}}_1 \\ \hat{\mathbf{q}}_1 \\ \hat{\mathbf{c}}_2 \\ \hat{\mathbf{q}}_2 \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{B}}_1 \\ 0 \\ \hat{\mathbf{B}}_2 \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} \epsilon_c \\ 1 \end{bmatrix} \otimes \begin{bmatrix} \hat{\mathbf{f}}_1(\hat{\mathbf{c}}_1, \hat{\mathbf{q}}_1, \hat{\mathbf{c}}_2, \hat{\mathbf{q}}_2) \\ \hat{\mathbf{f}}_2(\hat{\mathbf{c}}_1, \hat{\mathbf{q}}_1, \hat{\mathbf{c}}_2, \hat{\mathbf{q}}_2) \end{bmatrix},$$

where $\hat{\mathbf{c}}_1, \hat{\mathbf{q}}_1, \hat{\mathbf{c}}_2, \hat{\mathbf{q}}_2 \in \mathbf{R}^r$, $\hat{\mathbf{A}}_1, \hat{\mathbf{A}}_2 \in \mathbf{R}^{r \times r}$, $\hat{\mathbf{B}}_1, \hat{\mathbf{B}}_2 \in \mathbf{R}^r$.

Using our model knowledge: Maintaining the coupling structure in learned ROMs

For Operator Inference, we assemble the following data (with $T = (0, 10]$s and $\delta t = 10^{-5}$) :

$$\hat{\mathbf{S}} = \begin{bmatrix} \mathbf{V}_{\mathbf{c}_1}^\top \mathbf{C}_1 \\ \mathbf{V}_{\mathbf{q}_1}^\top \mathbf{Q}_1 \\ \mathbf{V}_{\mathbf{c}_2}^\top \mathbf{C}_2 \\ \mathbf{V}_{\mathbf{q}_2}^\top \mathbf{Q}_2 \end{bmatrix} =: \begin{bmatrix} \hat{\mathbf{C}}_1 \\ \hat{\mathbf{Q}}_1 \\ \hat{\mathbf{C}}_2 \\ \hat{\mathbf{Q}}_2 \end{bmatrix}, \qquad \dot{\hat{\mathbf{S}}} = \begin{bmatrix} \mathbf{V}_{\mathbf{c}_1}^\top \dot{\mathbf{C}}_1 \\ \mathbf{V}_{\mathbf{q}_1}^\top \dot{\mathbf{Q}}_1 \\ \mathbf{V}_{\mathbf{c}_2}^\top \dot{\mathbf{C}}_2 \\ \mathbf{V}_{\mathbf{q}_2}^\top \dot{\mathbf{Q}}_2 \end{bmatrix} =: \begin{bmatrix} \dot{\hat{\mathbf{C}}}_1 \\ \dot{\hat{\mathbf{Q}}}_1 \\ \dot{\hat{\mathbf{C}}}_2 \\ \dot{\hat{\mathbf{Q}}}_2 \end{bmatrix}.$$

Block structure preservation in learned ROM

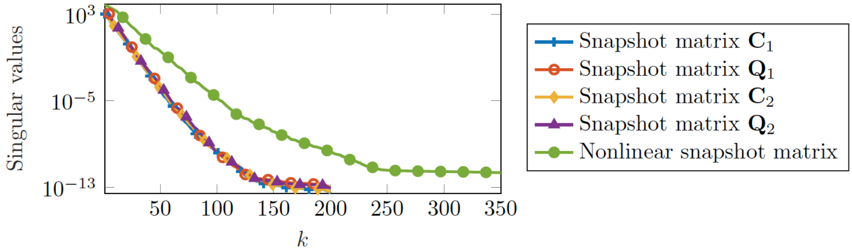
We can maintain the coupling structure by **solving separate least-squares problems** of the form

$$\min_{\hat{\mathbf{A}}_i, \hat{\mathbf{B}}_i} \|\dot{\hat{\mathbf{C}}}_i - \epsilon_c \mathbf{V}_{\mathbf{c}_i}^\top \mathbf{F} - \hat{\mathbf{A}}_i \hat{\mathbf{C}}_i - \hat{\mathbf{B}}_i \mathbf{U}\|_F, \quad i \in \{1, 2\}.$$

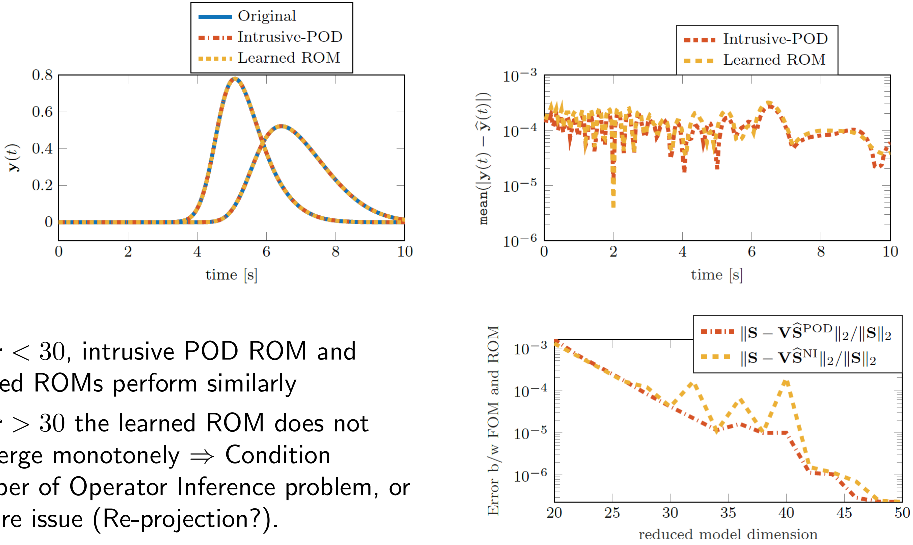
- Important for physical interpretability of the model
- Numerical implications, stability [Liao et al., 2007; Reis & Stykel, 2007, 2008; Benner & Feng, 2015; Kramer, 2016]

Learned ROM and projection-based ROM

- Both the learned ROM and intrusive ROM retain coupling structure
- Singular values decay rather slowly due to transport nature of Batch Chromatography
- ROMs of order $r = 22$ for each variable
- DEIM approximation used for nonlinear term $\mathbf{f}_i(\mathbf{c}_1, \mathbf{q}_1, \mathbf{c}_2, \mathbf{q}_2)$



Learned ROM accuracy



- For $r < 30$, intrusive POD ROM and learned ROMs perform similarly
- For $r > 30$ the learned ROM does not converge monotonely $\Rightarrow$ Condition number of Operator Inference problem, or closure issue (Re-projection?).

Conclusions: Learned ROMs with non-polynomial structure

- Accurate non-intrusive ROMs possible with Operator Inference learning framework
- Convergence results shows that under mild assumptions on the time stepping and step size, the non-intrusively learned reduced models converge to the same reduced models as obtained with intrusive model reduction methods.
- **The more we know about the model, the more we can incorporate into the learning framework: Model structure, nonlinear terms, etc.**

Operator Inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms, Benner/Goyal/K./Peherstorfer/Willcox, 2020

Part 2:

Operator Inference to learn ROMs for control applications

Feedback control for systems with uncertain parameters using online-adaptive reduced models.
K./Peherstorfer/Willcox, SIAM J. on Applied Dynamical Systems 16(3), pp. 1563-1586, 2017.

Part 2: Problem setting & Goal

Goal: Control a plant that has the form:

$$\dot{\mathbf{s}}(t) = \mathbf{A}(\mathbf{q}(t))\mathbf{s}(t) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{s}(0) = \mathbf{s}_0 \ \in \mathbb{R}^n$$

with time-dependent parameters that assume a residence time:

$$\mathbf{q}(t) = \mathbf{q}_{\mathcal{T}_i} \quad \text{for} \quad t \in \mathcal{T}_i = [t_{i-1}, t_i].$$

Problem setting and challenges:

- **Switching times** t_i not known a priori; have to be detected online
- Dynamical system response changes with $\mathbf{q}(t)$ (stability, equilibria)
- Large-scale setting, (n large)
- Cannot evaluate $\mathbf{A}(\mathbf{q}(t))$ online (b/c recourse to full model solver)

Opportunities:

- State-space data from plant available
- Can use operator inference to learn ROM state-space model

Formulating the control problem

Linear Quadratic Regulator Problem

$$\min_{\mathbf{u}} \; J(\mathbf{s},\mathbf{u};\mathbf{q}) = \int_0^\infty ||\mathbf{C}\mathbf{s}(t;\mathbf{q})||_2^2 + ||\mathbf{R}\mathbf{u}(t;q)||_2^2 \; \mathrm{d}t$$

$$\text{s.t.} \quad \dot{\mathbf{s}}(t) = \mathbf{A}(\mathbf{q}(t))\mathbf{s}(t) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{s}(0) = \mathbf{s}_0 \; \in \mathbb{R}^n$$

$\mathbf{u}(t;q)$: control; $\mathbf{B},\mathbf{C}$ known input and sensing matrices.

Control problem for plant with uncertainties

Let $\mathbf{A}(\mathbf{q}(t))$ be available offline, but not online. Let $\mathbf{B},\mathbf{C}$ be available online. For all $\mathcal{T}_i = [t_{i-1}, t_i], i = 1, 2, \ldots,$ solve the control problem:

$$\min_{\mathbf{s}_{\mathcal{T}_i}, \mathbf{u}_{\mathcal{T}_i}} \; J(\mathbf{s}_{\mathcal{T}_i}, \mathbf{u}_{\mathcal{T}_i})$$

$$\text{s.t.} \quad \dot{\mathbf{s}}_{\mathcal{T}_i}(t) = \mathbf{A}(\mathbf{q}_{\mathcal{T}_i})\mathbf{s}_{\mathcal{T}_i}(t) + \mathbf{B}\mathbf{u}_{\mathcal{T}_i}(t),$$

where the subscripts indicate the state and control in $\mathcal{T}_i = [t_{i-1}, t_i]$.

A solution approach to the problem

How can we detect switching t_i? How to deal with not-available $\mathbf{A}(\mathbf{q}_{\mathcal{T}_i})$

Approach:

- Reduce-then-design approach: Design a controller based-on low-dimensional ROM
- Use data to learn and update ROM matrix online $\Rightarrow$ robustness to parametric changes

Algorithmic details:

- **Offline:** Library of high-fidelity solutions (subspaces, feedback gains)
- **Online:**
  1. Detect parameter-dependent subspace $\mathbf{V} = \mathbf{V}(\mathbf{q}) \Rightarrow$ Deals with unknown switching times t_i
  2. Learn system matrix $\hat{\mathbf{A}}(\mathbf{q}(t)) = \mathbf{V}^\top \mathbf{A}(\mathbf{q}(t))\mathbf{V}$ in real-time through data-driven ROMs; recompute optimal feedback from surrogate model

Related work

Reduced-order modeling for control design:

[Burns and King, 1998, Burns et al., 1999,
Kunisch and Volkwein, 1999, Atwell et al., 2001,
Banks et al., 2000, Banks et al., 2002, Benner, 2004,
Lee and Tran, 2005, Borggaard and Stoyanov, 2008,
Sachs and Volkwein, 2010, Alla and Falcone, 2013,
Nicaise et al., 2014, Tissot et al., 2015, Pyta et al., 2015] .....

Online-Interpolated ROMs for parameter-dependent systems

[Poussot-Vassal and Sipp, 2015]:

- ROMs generated offline for linearized equations; online interpolation (parameter known)

Gain Scheduling for LPV systems

[Becker and Packard, 1994, Theis et al., 2016]

- Linearization of NL systems around operating conditions/equlibria

Offline (high fidelity)-online (compressed sensing detection) strategy [Mathelin et al., 2012]

- Controllers parametrized by initial condition

Statistical learning strategy [Guéniat et al., 2016]

- Markov process model + Reinforcement learning for control law

Learning LQR controller from random control input excitations [Dean et al., 2019, Cohen et al., 2019]

- Complete theory for error in learning unknown linear model ($||\mathbf{A} - \mathbf{A}^{\text{est}}|| < ....,||\mathbf{B} - \mathbf{B}^{\text{est}}|| < ...$) and robust controller
- Finite-data results, but assumes randomized controller trials can be made

Reduced-order models to represent dynamics

We approximate the dynamics as

$$\mathbf{s}(t;\mathbf{q}) \approx \mathbf{V}(\mathbf{q})\hat{\mathbf{s}}(t;\mathbf{q}), \qquad \mathbf{V}(\mathbf{q}) \in \mathbb{R}^{n\times r}, \quad r \ll n,$$

where $\mathbf{V}(\mathbf{q})$ contains basis vectors for a low-dimensional, accurate representation of the dynamics. Enforcing orthogonality of the residual yields a ROM of similar structure

$$\dot{\hat{\mathbf{s}}}(t;\mathbf{q}) = \hat{\mathbf{A}}(\mathbf{q}(t))\hat{\mathbf{s}}(t;\mathbf{q}) + \hat{\mathbf{B}}\mathbf{u}(t),$$

where

$$\hat{\mathbf{A}}(\mathbf{q}) = \mathbf{V}(\mathbf{q})^\top \mathbf{A}(\mathbf{q})\mathbf{V}(\mathbf{q}), \qquad \hat{\mathbf{B}}(\mathbf{q}) = \mathbf{V}(\mathbf{q})^\top \mathbf{B}.$$

We compute a library of subspaces $\mathbf{V}(\mathbf{q})$ for a suitably chosen selection $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M\}$.

Low-rank structure in LQR

If $(\mathbf{A}(\mathbf{q}), \mathbf{B})$ is stabilizable, then control assumes *linear* state feedback:

$$\mathbf{u}(t;q) = -\mathbf{K}(\mathbf{q})\mathbf{s}(t;\mathbf{q}) = -[\mathbf{R}^{-1}\mathbf{B}^\top \Pi(\mathbf{q})]\mathbf{s}(t;\mathbf{q}), \tag{2}$$

$$\mathbf{0} = \mathbf{A}^\top(\mathbf{q})\Pi(\mathbf{q}) + \Pi(\mathbf{q})\mathbf{A}(\mathbf{q}) - \Pi(\mathbf{q})\mathbf{B}\mathbf{B}^\top\Pi(\mathbf{q}) + \mathbf{C}^\top\mathbf{C} \tag{3}$$

where $\mathbf{K}(\mathbf{q})$ is the gain matrix.

$\Pi(\mathbf{q})$ **often of low numerical rank:**

$$\Pi(\mathbf{q}) = \mathbf{W}(\mathbf{q})\mathbf{W}(\mathbf{q})^\top, \quad \mathbf{W}(\mathbf{q}) \in \mathbb{R}^{n\times r}$$

$\Rightarrow$ Work with low-dimensional operators through $\mathbf{V} \in \mathbb{R}^{n\times r}$:

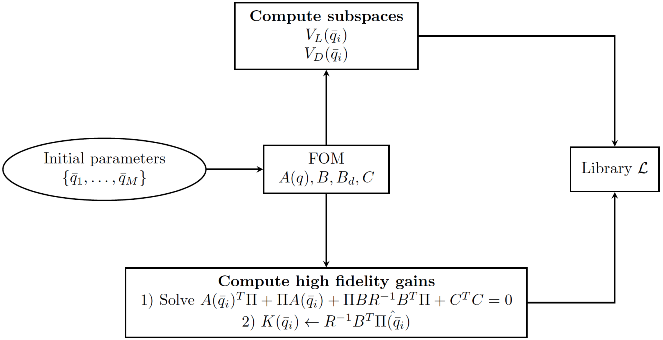
$$\hat{\mathbf{A}}(\mathbf{q}) = \mathbf{V}^\top\mathbf{A}(\mathbf{q})\mathbf{V}, \quad \hat{\mathbf{B}} = \mathbf{V}^\top\mathbf{B}, \quad \hat{\mathbf{C}} = \mathbf{C}\mathbf{V}$$

Solve (3) in low dimensions (*reduce-then-design*):

$$\mathbf{0} = \hat{\mathbf{A}}^\top(\mathbf{q})\hat{\Pi}(\mathbf{q}) + \hat{\Pi}(\mathbf{q})\hat{\mathbf{A}}(\mathbf{q}) - \hat{\Pi}(\mathbf{q})\hat{\mathbf{B}}\hat{\mathbf{B}}^\top\hat{\Pi}(\mathbf{q}) + \hat{\mathbf{C}}^\top\hat{\mathbf{C}}$$

[Jbilou, 2003, Jbilou, 2006, Heyouni and Jbilou, 2009, Benner et al., 2008a, Benner et al., 2008b, Simoncini et al., 2013, Lin and Simoncini, 2014, Wang et al., 2014, Li et al., 2013, Kramer and Singler, 2016]. . .

Offline: Pre-computing library of feedback gains



Feedback gains $\mathbf{K}(\mathbf{q}_i)$, learning bases $\mathbf{V}_L(\mathbf{q}_i)$, detection bases $\mathbf{V}_D(\mathbf{q}_i)$, for $i = 1, \ldots, M$:

$$\mathcal{L} := \left\{ \left\{ \begin{matrix} \mathbf{V}_L(\mathbf{q}_1) \\ \mathbf{V}_D(\mathbf{q}_1) \\ \mathbf{K}(\mathbf{q}_1) \end{matrix} \right\}, \ldots, \left\{ \begin{matrix} \mathbf{V}_L(\mathbf{q}_M) \\ \mathbf{V}_D(\mathbf{q}_M) \\ \mathbf{K}(q_M) \end{matrix} \right\} \right\}.$$

Online: Solving the classification problem

- $\mathcal{S}$: selection operator that selects p' entries from the states $\mathbf{s}(t;\mathbf{q}(t))$ with $p' \in \{1,\ldots,n\}$ and $p' \ll n$
- Detection subspaces $\mathbf{V}_D(\bar{\mathbf{q}}_i),\ i = 1,\ldots,M$
- Define classifier $h : \mathbb{R}^{p'} \to \{1,\ldots,M\}$ via

$$h(\mathcal{S}\mathbf{s}(t;\bar{\mathbf{q}}_k)) = k$$

- Solve classification problem by projection $\mathbf{P}_i : \mathbb{R}^{p'} \mapsto \mathbb{R}^{p'}$:

$$\mathbf{P}_i = \mathcal{S}\mathbf{V}_D(\bar{\mathbf{q}}_i)\left[(\mathcal{S}\mathbf{V}_D(\bar{\mathbf{q}}_i))^\top(\mathcal{S}\mathbf{V}_D(\bar{\mathbf{q}}_i))\right]^{-1}(\mathcal{S}\mathbf{V}_D(\bar{\mathbf{q}}_i))^\top$$

- Selected subspace:

$$k = \arg\max_{i=1,\ldots,M} ||\mathbf{P}_i(\mathcal{S}\mathbf{s}(t;\mathbf{q}))||_2$$

- **Next step: Act on information and initiate model learning**

Online: ROM learning from recorded data

Goal: Learn ROM system matrix $\hat{\mathbf{A}} = \hat{\mathbf{A}}(\tilde{\mathbf{q}}) \in \mathbb{R}^{r\times r}$ from data $\mathbf{s}_1(\tilde{\mathbf{q}}), \dots, \mathbf{s}_\ell(\tilde{\mathbf{q}})$ and selected subspace $\mathbf{V} = \mathbf{V}_L(\bar{\mathbf{q}}_k) \in \mathcal{L}$.

- Compute $\hat{\mathbf{B}} = \mathbf{V}^\top \mathbf{B}$ and $\hat{\mathbf{B}}_d = \mathbf{V}^\top \mathbf{B}_d$
- Assemble past control inputs $\mathbf{u}_k = \mathbf{u}(t_k; \mathbf{q}(t_k))$:

$$\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_s]^\top \;\in\; \mathbb{R}^{s\times m},$$

- Reduced states $\hat{\mathbf{s}}_i := \mathbf{V}^\top \mathbf{s}(t_i)$ stored in

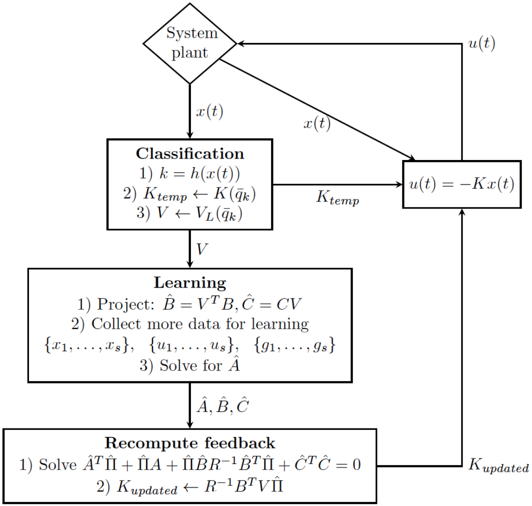
$$\hat{\mathbf{S}} = [\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2, \dots, \hat{\mathbf{s}}_\ell]^\top \;\in\; \mathbb{R}^{s\times r}$$

- Derivative approximation $\Rightarrow \dot{\hat{\mathbf{s}}}_1, \dots, \dot{\hat{\mathbf{s}}}_\ell$
- **Operator inference problem for $\hat{\mathbf{A}} = \hat{\mathbf{A}}(\tilde{\mathbf{q}})$:**

$$\min_{\hat{\mathbf{A}} \in \mathbb{R}^{r\times r}} \sum_{i=1}^{s} \left\| \dot{\hat{\mathbf{s}}}_i - \hat{\mathbf{A}}\hat{\mathbf{s}}_i - \hat{\mathbf{B}}\mathbf{u}_i \right\|_2^2$$

Convergence to projected matrices established [Peherstorfer and Willcox, 2016]

Online: Detect subspace and learn model



- Online classification gives "best-fit" low-dimensional subspace $\mathbf{V}_L(\mathbf{q}_k)$ for learning
- **Learn/adapt** new ROM system matrix $\hat{\mathbf{A}}(\mathbf{q})$ by incorporating real-time data

Numerical example: Convection-diffusion PDE

Let $\mathbf{x} = [x_1, x_2]^\top \in \Omega = [0,1]^2$ and

$$\frac{\partial \theta}{\partial t}(t,\mathbf{x}) = q(t)\left(\frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2}\right)\theta(t,\mathbf{x}) - x_2 \frac{\partial \theta}{\partial x_2}(t,\mathbf{x}) + b(\mathbf{x})u(t) + b_d g(t)$$

- Boundary conditions:

$$\theta(t, x_1, 0) = 0, \quad \theta(t, 1, x_2) = 0, \quad \theta(t, x_1, 1) = 0, \quad \frac{\partial \theta}{\partial x_1}(t, 0, x_2) = 0$$

- Control enters through $b(\mathbf{x}) = 5$ if $x_1 \geq 1/2$ and 0 otherwise
- Uncertainty $\Rightarrow$ diffusion coefficient $q(t) \in \mathbb{R}$

Spatially discretized system (piecewise linear FE):

$$\begin{aligned} \dot{\mathbf{s}}(t;q) &= \mathbf{A}(q(t))\mathbf{s}(t;q) + \mathbf{B}\tilde{u}(t) + \mathbf{B}_d g(t), \quad \mathbf{s}(0) = \mathbf{s}_0 \in \mathbb{R}^n \\ y(t) &= \mathbf{C}\mathbf{s}(t) \in \mathbb{R} \end{aligned}$$

- $\mathbf{C} = \frac{5}{n}[1, \ldots, 1]$.
- $n = 3540$ for FEM discretization = "truth model"

Eigenvalues of $\mathrm{A}(Pe)$ for different Péclet numbers

- Parameter $\bar{q}_i$ = Péclet number = $\frac{\text{convective transport rate}}{\text{diffusive transport rate}}$
- High Péclet numbers indicate strongly convective flows
- Four Péclet numbers to generate library $\mathcal{L}: \; Pe \in \{2, 10, 50, 1000\}$

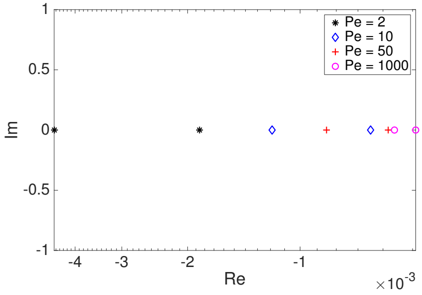
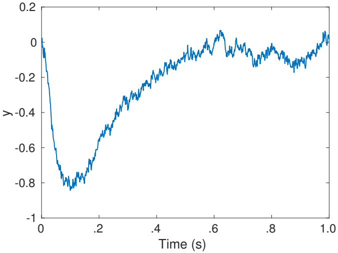
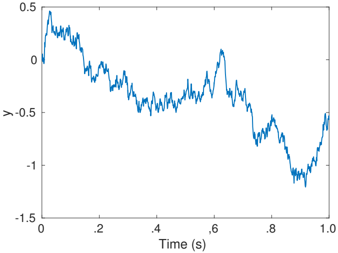


Figure: The two eigenvalues with largest real part of the system matrix $\mathbf{A}(\bar{q}_i)$.

Open loop output and gains for different Péclet numbers

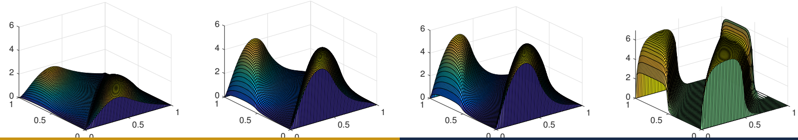


(a) Output for $Pe_1 = 2$



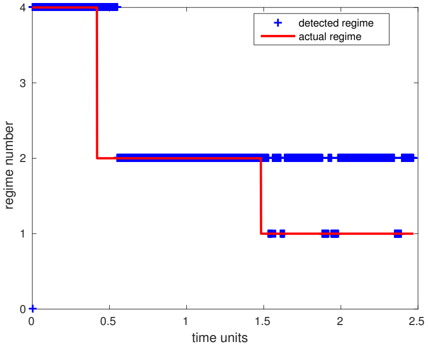
(b) Output for $Pe_2 = 1000$

Figure: Output $y(t)$ of the open loop convection diffusion system, excited with nonzero initial condition $s_0(x, y) = 15\sin(2\pi x)\sin(\pi y)$; disturbance $g(t) \propto \mathcal{N}(0, 0.5)$ applied through a disturbance term at $0 \leq x_1 \leq 0.05$.

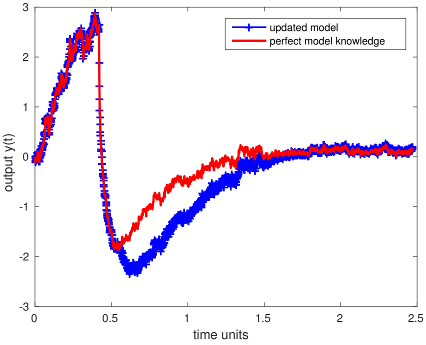


Output with learned and intrusively-projected controller

- Simulate online until $T = 2.5s$
- Learning basis $r = 10$ (eigenbasis)
- Detection basis (POD) $r = 30$ from $S = 1,000$ snapshots
- Misclassification after $t > 1.5s$ due to similar equilibrium solutions



(e) Selected library elements as indicated by the detection function$h(\cdot)$.



(f) Controlled output $y(t)$.

Numerical example II: Permeablity

Laplace equation on $\Omega = [0,1]^2$ as model of flow through porous medium

$$\frac{\partial}{\partial t}\theta(t,\mathbf{x}) = \nu(t,\mathbf{x}) \cdot \left(\frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2}\right)\theta(t,\mathbf{x}) + b(\mathbf{x})u(t) + b_d^1(\mathbf{x})g_1(t) + b_d^2(\mathbf{x})g_2(t)$$

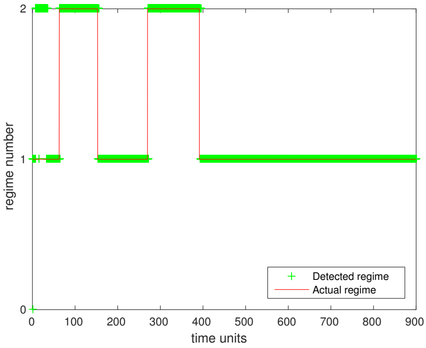
- Uncertain permeability: $q(t,\mathbf{x}) = \nu(t,\mathbf{x})$
- Spatial discretization: $\mathbf{s} \in \mathbb{R}^n$:

$$\begin{aligned}\dot{\mathbf{s}}(t) &= \mathbf{A}(\mathbf{q}(t))\mathbf{s}(t) + \mathbf{B}u(t) + \mathbf{B}_d g(t)\\ y(t) &= \mathbf{C}\mathbf{s}(t)\end{aligned}$$

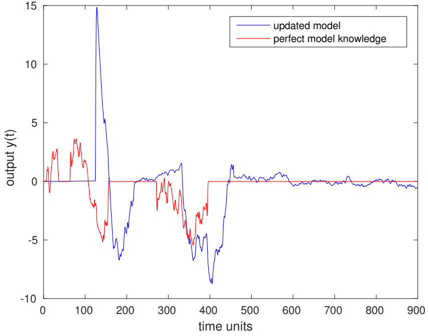
- White noise disturbance $g(\cdot)$ enters through $\mathbf{B}_d$ at $[x_1, x_2] = [0.3, 0.3]$ and $[x_1, x_2] = [0.3, 0.7]$
- Sensor location at $[x_1, x_2] = [0.5, 0.6]$
- Control location at $[x_1, x_2] = [0.6, 0.7]$
- Compute library $\mathcal{L}$ using three permeability fields $\mathbf{q}_1(\mathbf{x}), \mathbf{q}_2(\mathbf{x}), \mathbf{q}_3(\mathbf{x})$

Output of controlled system

- Learning basis $\mathbf{V}_L(\mathbf{q}_i)$: Eigenbasis of order 20 for $\mathbf{A}(\mathbf{q}_i), i = 1, 2, 3$
- Detection basis $\mathbf{V}_D(\mathbf{q}_i)$: POD basis of order 20 from closed-loop system excited with disturbances $\mathbf{g}(t) = [g_1(t), g_2(t)]^\top$. The system was simulated for $500s$, and $\ell = 10,000$ snapshots were used to compute the POD basis
- Stable regime 1, unstable regime 2



(g) Permeabilities selected by detection func-



(h) Output $y(t)$ of controlled systems

Control performance

- Case 1: Controller designed from projection-based ROM with perfect knowledge of $q(t)$
- Case 2: Controller designed from learned ROM (Operator Inference model)

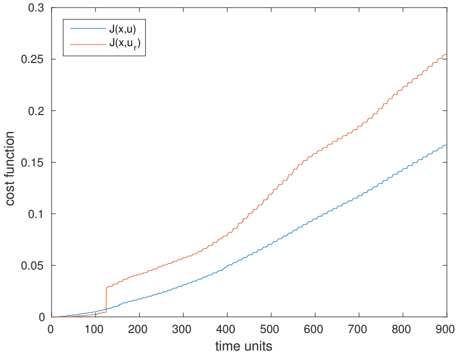
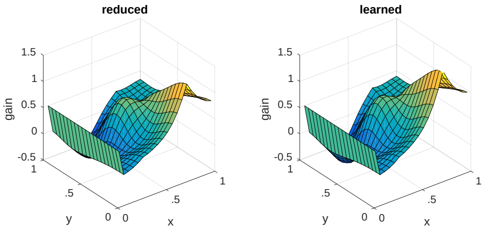
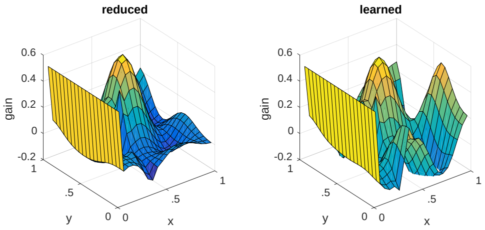


Figure: Control cost function $J(\mathbf{s}, u)$ for both controllers on the full-order model.

Feedback gains: Intrusive ROM vs learned ROM



(a) Feedback gains at $t = 122s$.



(b) Feedback gains at $t = 574s$

Review and conclusions

Review:

- **Situation:** $\mathbf{A}(\mathbf{q}(t))$ is not available online, neither are switching times t_i
- Learning-based control framework for high-dimensional LPV system:
  1. Learning unknown ROM system matrix from data (Operator Inference)
  2. Detected switching time with localized subspace approach

Conclusions:

- Learning-based low-dimensional controller performs well and feedback gains are accurate
- Robustness to parametric changes can be addressed through online learning
- More robust controller designs could be obtained through $(\mathcal{H}_\infty)$

Feedback control for systems with uncertain parameters using online-adaptive reduced models.
K./Peherstorfer/Willcox, SIAM J. on Applied Dynamical Systems 16(3), pp. 1563-1586, 2017.


Scripps Institute of Oceanography
Geisel Library
Jacobs School of Engineering
SD Supercomputer Center

THANK YOU

Alla, A. and Falcone, M. (2013).
An adaptive pod approximation method for the control of advection-diffusion equations.
In *Control and Optimization with PDE Constraints*, pages 1–17. Springer.

Atwell, J., Borggaard, J., and King, B. (2001).
Reduced order controllers for burgers' equation with a nonlinear observer.
Applied Mathematics And Computer Science, 11(6):1311–1330.

Banks, H. T., Beeler, S. C., Kepler, G. M., and Tran, H. T. (2002).
Reduced order modeling and control of thin film growth in an HPCVD reactor.
SIAM J. Appl. Math., 62(4):1251–1280.

Banks, H. T., del Rosario, R. C. H., and Smith, R. C. (2000).
Reduced-order model feedback control design: numerical implementation in a thin shell model.
IEEE Trans. Automat. Control, 45(7):1312–1324.

Becker, G. and Packard, A. (1994).
Robust performance of linear parametrically varying systems using parametrically-dependent linear feedback.
Systems & Control Letters, 23(3):205–215.

Benner, P. (2004).
Solving large-scale control problems.
IEEE Control Systems Magazine, 24(1):44–59.

Benner, P., Li, J.-R., and Penzl, T. (2008a).
Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems.
Numer. Linear Algebra Appl., 15(9):755–777.

Benner, P., Mena, H., and Saak, J. (2008b).
On the parameter selection problem in the Newton-ADI iteration for large-scale Riccati equations.
Electronic Transactions on Numerical Analysis, 29:136–149.

Borggaard, J. and Stoyanov, M. (2008).
An efficient long-time integrator for Chandrasekhar equations.
In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 3983–3988.

Burns, J. A. and King, B. B. (1998).
A reduced basis approach to the design of low-order feedback controllers for nonlinear continuous systems.
Journal of Vibration and Control, 4(3):297–323.

Burns, J. A., King, B. B., and Rubio, D. (1999).
On the design of feedback controllers for a convecting fluid flow via reduced order modeling.
In *Proceedings of the 1999 IEEE International Conference on Control Applications (Cat. No. 99CH36328)*, volume 2, pages 1157–1162. IEEE.

Clason, C., Ito, K., and Kunisch, K. (2016).
A convex analysis approach to optimal controls with switching structure for partial differential equations.
ESAIM: Control, Optimisation and Calculus of Variations, 22(2):581–609.

Cohen, A., Koren, T., and Mansour, Y. (2019).
Learning linear-quadratic regulators efficiently with only $\mathcal{O}(\sqrt{T})$ regret.
arXiv:1902.06223.

Dean, S., Mania, H., Matni, N., Recht, B., and Tu, S. (2019).
On the sample complexity of the linear quadratic regulator.
Foundations of Computational Mathematics.

Guéniat, F., Mathelin, L., and Hussaini, M. (2016).
A statistical learning strategy for closed-loop control of fluid flows.
Theoretical and Computational Fluid Dynamics, pages 1–14.

Heyouni, M. and Jbilou, K. (2009).
An extended block Arnoldi algorithm for large-scale solutions of the continuous-time algebraic Riccati equation.
Electron. Trans. Numer. Anal., 33:53–62.

Jbilou, K. (2003).
Block Krylov subspace methods for large algebraic Riccati equations.
Numerical Algorithms, 34(2-4):339–353.

Jbilou, K. (2006).
An Arnoldi based algorithm for large algebraic Riccati equations.
Applied Mathematics Letters, 19(5):437–444.

Kramer, B., Peherstorfer, B., and Willcox, K. (2017).
Feedback control for systems with uncertain parameters using online-adaptive reduced models.
SIAM Journal on Applied Dynamical Systems, 16(3):1563–1586.

Kramer, B. and Singler, J. R. (2016).
A POD projection method for large-scale algebraic Riccati equations.
Numerical Algebra, Control & Optimization, 6:413–435.

Kunisch, K. and Volkwein, S. (1999).
Control of the Burgers equation by a reduced-order approach using proper orthogonal decomposition.
J. Optim. Theory Appl., 102(2):345–371.

Lee, C. H. and Tran, H. T. (2005).
Reduced-order-based feedback control of the Kuramoto-Sivashinsky equation.
J. Comput. Appl. Math., 173(1):1–19.

Li, T., wah Chu, E. K., Lin, W.-W., and Weng, P. C.-Y. (2013).
Solving large-scale continuous-time algebraic riccati equations by doubling.
Journal of Computational and Applied Mathematics, 237(1):373 – 383.

Lin, Y. and Simoncini, V. (2014).
A new subspace iteration method for the algebraic riccati equation.
Numerical Linear Algebra with Applications.

Mathelin, L., Pastur, L., and Le Maître, O. (2012).
A compressed-sensing approach for closed-loop optimal control of nonlinear systems.
Theoretical and computational fluid dynamics, 26(1-4):319–337.

Nicaise, S., Stingelin, S., and Tröltzsch, F. (2014).
On two optimal control problems for magnetic fields.
Computational Methods in Applied Mathematics, 14(4):555–573.

Peherstorfer, B. and Willcox, K. (2016).
Data-driven operator inference for nonintrusive projection-based model reduction.
Computer Methods in Applied Mechanics and Engineering, 306:196–215.

Poussot-Vassal, C. and Sipp, D. (2015).
Parametric reduced order dynamical model construction of a fluid flow control problem.
IFAC-PapersOnLine, 48(26):133–138.

Pyta, L., Herty, M., and Abel, D. (2015).
Optimal feedback control of the incompressible navier-stokes-equations using reduced order models.
In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 2519–2524. IEEE.

Sachs, E. and Volkwein, S. (2010).
Pod-galerkin approximations in pde-constrained optimization.
GAMM-Mitteilungen, 33(2):194–208.

Simoncini, V., Szyld, D. B., and Monsalve, M. (2013).
On two numerical methods for the solution of large-scale algebraic Riccati equations.
IMA Journal of Numerical Analysis.

Theis, J., Seiler, P., and Werner, H. (2016).
Model order reduction by parameter-varying oblique projection.
In *2016 American Control Conference (ACC)*, pages 4586–4591. IEEE.

Tissot, G., Cordier, L., and Noack, B. R. (2015).
Feedback stabilization of an oscillating vertical cylinder by pod reduced-order model.
In *Journal of Physics: Conference Series*, volume 574, page 012137. IOP Publishing.

Wang, X., Li, W.-W., and Dai, L. (2014).
On inexact newton methods based on doubling iteration scheme for symmetric algebraic riccati equations.
Journal of Computational and Applied Mathematics, 260(0):364 – 374.