MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

CSC
COMPUTATIONAL METHODS IN
SYSTEMS AND CONTROL THEORY

Learning Globally Stable Dynamics

A Matrix-theoretic Perspective

Peter Benner

Householder Symposium XXII
June 8–13, 2025
Cornell University, Ithaca, NY (USA)

Supported by:

DFG-Graduiertenkolleg
MATHEMATISCHE
KOMPLEXITÄTSREDUKTION

IMPRS
ProEng
Magdeburg

Partners:

OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

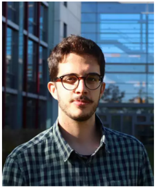
Collaborators

Pawan. K Goyal

Yevgeniya Filanova

Igor Pontes Duff

Leonidas Gkimisis



Siddarth Mamidisetti

Süleyman Yıldız

Overview

Original System

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

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

Model Order Reduction of Dynamical Systems

Original System

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

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

Reduced-Order Model (ROM)

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

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

Model Order Reduction of Dynamical Systems

Original System

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

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

Reduced-Order Model (ROM)

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

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

Goals:

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

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

Original System

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

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

Reduced-Order Model (ROM)

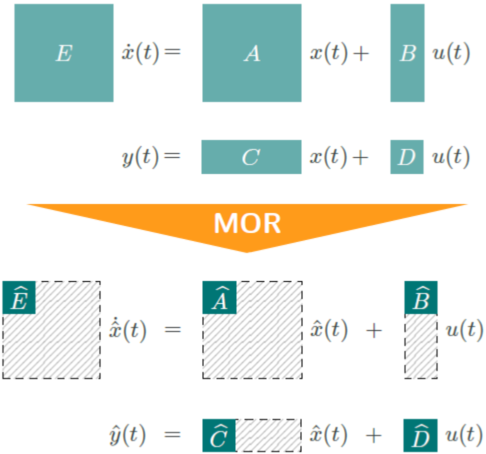
$$\widehat{\Sigma} : \begin{cases} \dot{\hat{x}}(t) = \widehat{A}\hat{x}(t) + \widehat{B}u(t), \\ \hat{y}(t) = \widehat{C}\hat{x}(t) + \widehat{D}u(t). \end{cases}$$

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

Goals:

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

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



- $E, A \in \mathbb{R}^{n\times n}$
- $B \in \mathbb{R}^{n\times m}$
- $C \in \mathbb{R}^{p\times n}$
- $D \in \mathbb{R}^{p\times m}$

- $\hat{E}, \hat{A} \in \mathbb{R}^{r\times r}$
- $\hat{B} \in \mathbb{R}^{r\times m}$
- $\hat{C} \in \mathbb{R}^{p\times r}$
- $\hat{D} \in \mathbb{R}^{p\times m}$

Assumption: trajectory $x(t;u)$ is contained in low-dimensional subspace $\mathcal{V} \subset \mathbb{R}^n$.
Thus, use Galerkin or Petrov-Galerkin-type projection of state-space onto $\mathcal{V}$ (trial space) along complementary subspace $\mathcal{W}$ (test space), where

$$\operatorname{range}(V) = \mathcal{V}, \quad \operatorname{range}(W) = \mathcal{W}, \quad W^T V = I_r.$$

The reduced-order model is

$$\hat{x} = W^T x, \quad \hat{A} := W^T A V, \quad \hat{B} := W^T B, \quad \hat{C} := CV, \quad (\hat{D} := D).$$

Assumption: trajectory $x(t;u)$ is contained in low-dimensional subspace $\mathcal{V} \subset \mathbb{R}^n$.
Thus, use Galerkin or Petrov-Galerkin-type projection of state-space onto $\mathcal{V}$ (trial space) along complementary subspace $\mathcal{W}$ (test space), where

$$\mathrm{range}(V) = \mathcal{V}, \quad \mathrm{range}(W) = \mathcal{W}, \quad W^T V = I_r.$$

The reduced-order model is

$$\hat{x} = W^T x, \quad \hat{A} := W^T A V, \quad \hat{B} := W^T B, \quad \hat{C} := CV, \quad (\hat{D} := D).$$

But: we need the matrices A, B, C, D to compute the reduced-order model!

From intrusive to non-intrusive MOR

Assumption: trajectory $x(t;u)$ is contained in low-dimensional subspace $\mathcal{V} \subset \mathbb{R}^n$.
Thus, use Galerkin or Petrov-Galerkin-type projection of state-space onto $\mathcal{V}$ (trial space) along complementary subspace $\mathcal{W}$ (test space), where

$$\operatorname{range}(V) = \mathcal{V}, \quad \operatorname{range}(W) = \mathcal{W}, \quad W^T V = I_r.$$

The reduced-order model is

$$\hat{x} = W^T x, \quad \hat{A} := W^T A V, \quad \hat{B} := W^T B, \quad \hat{C} := CV, \quad (\hat{D} := D).$$

But: we need the matrices A, B, C, D to compute the reduced-order model!

Using proprietary simulation software, we would need to **intrude** the software to get the matrices
$\rightsquigarrow$ **intrusive MOR**

$=$ *learning (compact, surrogate) models from (full, detailed) models.*

This is often impossible!

Assumption: trajectory $x(t;u)$ is contained in low-dimensional subspace $\mathcal{V} \subset \mathbb{R}^n$.
Thus, use Galerkin or Petrov-Galerkin-type projection of state-space onto $\mathcal{V}$ (trial space) along complementary subspace $\mathcal{W}$ (test space), where

$$\mathrm{range}(V) = \mathcal{V}, \quad \mathrm{range}(W) = \mathcal{W}, \quad W^T V = I_r.$$

The reduced-order model is

$$\hat{x} = W^T x, \quad \hat{A} := W^T AV, \quad \hat{B} := W^T B, \quad \hat{C} := CV, \quad (\hat{D} := D).$$

But: we need the matrices A, B, C, D to compute the reduced-order model!

Using proprietary simulation software, we would need to **intrude** the software to get the matrices
$\rightsquigarrow$ **intrusive MOR**

$=$ *learning (compact, surrogate) models from (full, detailed) models.*

This is often impossible!

$\rightsquigarrow$ **non-intrusive MOR**

$=$ *LEARNING (compact, surrogate) MODELS FROM DATA!*

Now assume we are only given an oracle, allowing us to compute y (including cases with $y \equiv x$), given $u(t)$ or $U(s)$:

Now assume we are only given an oracle, allowing us to compute y (including cases with $y \equiv x$), given $u(t)$ or $U(s)$:

Black box Σ: the only information we can get is either

- time domain data / times series: $u_k \approx u(t_k)$ and $x_k \approx x(t_k)$ or $y_k \approx y(t_k)$, or

Now assume we are only given an oracle, allowing us to compute y (including cases with $y \equiv x$), given $u(t)$ or $U(s)$:

Black box Σ: the only information we can get is either

- time domain data / times series: $u_k \approx u(t_k)$ and $x_k \approx x(t_k)$ or $y_k \approx y(t_k)$, or
- frequency domain data / measurements: $U_k \approx U(\jmath\omega_k)$ and $X_k \approx X(\jmath\omega_k)$ or $Y_k \approx Y(\jmath\omega_k)$.

Now assume we are only given an oracle, allowing us to compute y (including cases with $y \equiv x$), given $u(t)$ or $U(s)$:

Black box Σ: the only information we can get is either

- time domain data / times series: $u_k \approx u(t_k)$ and $x_k \approx x(t_k)$ or $y_k \approx y(t_k)$, or
- frequency domain data / measurements: $U_k \approx U(\jmath\omega_k)$ and $X_k \approx X(\jmath\omega_k)$ or $Y_k \approx Y(\jmath\omega_k)$.

Some methods:

- System identification (incl. ERA, N4SID, MOESP): frequency and time domain
  [Ho/Kalman 1966; Ljung 1987/1999; Van Overschee/De Moor 1994; Verhaegen 1994; De Wilde, Eykhoff, Moonen, Sima, ...]

Now assume we are only given an oracle, allowing us to compute y (including cases with $y \equiv x$), given $u(t)$ or $U(s)$:

Black box Σ: the only information we can get is either

- time domain data / times series: $u_k \approx u(t_k)$ and $x_k \approx x(t_k)$ or $y_k \approx y(t_k)$, or
- frequency domain data / measurements: $U_k \approx U(\jmath\omega_k)$ and $X_k \approx X(\jmath\omega_k)$ or $Y_k \approx Y(\jmath\omega_k)$.

Some methods:

- System identification (incl. ERA, N4SID, MOESP): frequency and time domain
  [Ho/Kalman 1966; Ljung 1987/1999; Van Overschee/De Moor 1994; Verhaegen 1994; De Wilde, Eykhoff, Moonen, Sima, ...]
- Neural networks: frequency and time domain [Narendra/Parthasarathy 1990; Lee/Carlberg 2019; Antonini/B./Feng/Romano 2022; ...]

Data-driven/-enhanced Model Reduction

Learning Models from Data

Now assume we are only given an oracle, allowing us to compute y (including cases with $y \equiv x$), given $u(t)$ or $U(s)$:

Black box Σ: the only information we can get is either

- time domain data / times series: $u_k \approx u(t_k)$ and $x_k \approx x(t_k)$ or $y_k \approx y(t_k)$, or
- frequency domain data / measurements: $U_k \approx U(\jmath\omega_k)$ and $X_k \approx X(\jmath\omega_k)$ or $Y_k \approx Y(\jmath\omega_k)$.

Some methods:

- System identification (incl. ERA, N4SID, MOESP): frequency and time domain
  [Ho/Kalman 1966; Ljung 1987/1999; Van Overschee/De Moor 1994; Verhaegen 1994; De Wilde, Eykhoff, Moonen, Sima, ...]
- Neural networks: frequency and time domain [Narendra/Parthasarathy 1990; Lee/Carlberg 2019; Antonini/B./Feng/Romano 2022; ...]
- Loewner interpolation: frequency (and time) domain [Antoulas/Anderson 1986; Mayo/Antoulas 2007; Gosea, Gugercin, Ionita, Lefteriu, Peherstorfer, ...]

Now assume we are only given an oracle, allowing us to compute y (including cases with $y \equiv x$), given $u(t)$ or $U(s)$:

Black box Σ: the only information we can get is either

- time domain data / times series: $u_k \approx u(t_k)$ and $x_k \approx x(t_k)$ or $y_k \approx y(t_k)$, or
- frequency domain data / measurements: $U_k \approx U(\jmath\omega_k)$ and $X_k \approx X(\jmath\omega_k)$ or $Y_k \approx Y(\jmath\omega_k)$.

Some methods:

- System identification (incl. ERA, N4SID, MOESP): frequency and time domain
  [Ho/Kalman 1966; Ljung 1987/1999; Van Overschee/De Moor 1994; Verhaegen 1994; De Wilde, Eykhoff, Moonen, Sima, . . .]
- Neural networks: frequency and time domain [Narendra/Parthasarathy 1990; Lee/Carlberg 2019; Antonini/B./Feng/Romano 2022; . . .]
- Loewner interpolation: frequency (and time) domain [Antoulas/Anderson 1986; Mayo/Antoulas 2007; Gosea, Gugercin, Ionita, Lefteriu, Peherstorfer, . . .]
- Koopman/Dynamic Mode Decomposition (DMD): time domain
  [Mezič 2005; Schmid 2008; Brunton, Kevrekidis, Kutz, Rowley, Noé, Nüske, Schütte, Peitz, Klus, . . .],
  for control systems [Kaiser/Kutz/Brunton 2017, B./Himpe/Mitchell 2018]

Now assume we are only given an oracle, allowing us to compute y (including cases with $y \equiv x$), given $u(t)$ or $U(s)$:

Black box Σ: the only information we can get is either

- time domain data / times series: $u_k \approx u(t_k)$ and $x_k \approx x(t_k)$ or $y_k \approx y(t_k)$, or
- frequency domain data / measurements: $U_k \approx U(\jmath\omega_k)$ and $X_k \approx X(\jmath\omega_k)$ or $Y_k \approx Y(\jmath\omega_k)$.

Some methods:

- System identification (incl. ERA, N4SID, MOESP): frequency and time domain
  [Ho/Kalman 1966; Ljung 1987/1999; Van Overschee/De Moor 1994; Verhaegen 1994; De Wilde, Eykhoff, Moonen, Sima, ...]
- Neural networks: frequency and time domain [Narendra/Parthasarathy 1990; Lee/Carlberg 2019; Antonini/B./Feng/Romano 2022; ...]
- Loewner interpolation: frequency (and time) domain [Antoulas/Anderson 1986; Mayo/Antoulas 2007; Gosea, Gugercin, Ionita, Lefteriu, Peherstorfer, ...]
- Koopman/Dynamic Mode Decomposition (DMD): time domain
  [Mezič 2005; Schmid 2008; Brunton, Kevrekidis, Kutz, Rowley, Noé, Nüske, Schütte, Peitz, Klus, ...],
  for control systems [Kaiser/Kutz/Brunton 2017, B./Himpe/Mitchell 2018]
- Operator inference (OpInf): time domain [Peherstorfer/Willcox 2016; Kramer, Qian, Farcas, B., Goyal, Pontes Duff, Yildiz,...]

A paper from 1990. . .

4 IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 1, NO. 1, MARCH 1990

Identification and Control of Dynamical Systems Using Neural Networks

KUMPATI S. NARENDRA FELLOW, IEEE, AND KANNAN PARTHASARATHY

Abstract—The paper demonstrates that neural networks can be used effectively for the identification and control of nonlinear dynamical systems. The emphasis of the paper is on models for both identification and control. Static and dynamic back-propagation methods for the adjustment of parameters are discussed. In the models that are introduced, multilayer and recurrent networks are interconnected in novel configurations and hence there is a real need to study them in a unified fashion. Simulation results reveal that the identification and adaptive control schemes suggested are practically feasible. Basic concepts and definitions are introduced throughout the paper, and theoretical questions which have to be addressed are also described.

are well known for such systems [1]. In this paper our interest is in the identification and control of nonlinear dynamic plants using neural networks. Since very few results exist in nonlinear systems theory which can be directly applied, considerable care has to be exercised in the statement of the problems, the choice of the identifier and controller structures, as well as the generation of adaptive laws for the adjustment of the parameters.

Two classes of neural networks which have received considerable attention in the area of artificial neural net-

Narendra, K.S., Parthasarathy, K. (1990): Identification and control of dynamical systems using neural networks. IEEE Transactions on Neural Networks 1(1):4–27.

A paper from 1990...

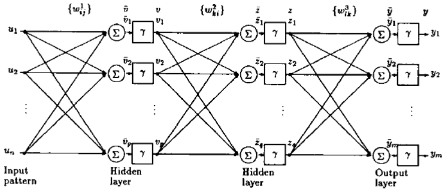


Fig. 2. A three layer neural network.

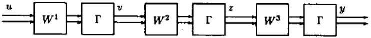
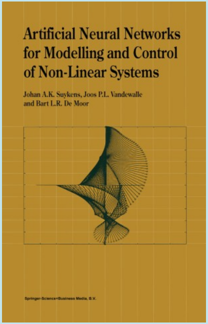


Fig. 3. A block diagram representation of a three layer network.

Narendra, K.S., Parthasarathy, K. (1990): Identification and control of dynamical systems using neural networks. IEEE Transactions on Neural Networks 1(1):4–27.

A book from 1996...



Narendra, K.S., Parthasarathy, K. (1990): Identification and control of dynamical systems using neural networks. IEEE Transactions on Neural Networks 1(1):4–27.

Suykens, J.A.K., Vandewalle, J.P.L., de Moor, B.L. (1996): Artificial Neural Networks for Modelling and Control of Non-Linear Systems. Springer US.

Problem setting: infer (smooth) nonlinear dynamical system,

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0 \in \mathbb{R}^n.$$

Goal: learn dynamical system from given snapshots $x_k := x(t_k)$, $u_k := u(t_k)$, $t_k := kh$, $h > 0$ for $k = 0, 1, \ldots, K$ (using simulation software, or measurements from real life experiment).

Problem setting: infer (smooth) nonlinear dynamical system,

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0 \in \mathbb{R}^n.$$

Goal: learn dynamical system from given snapshots $x_k := x(t_k)$, $u_k := u(t_k)$, $t_k := kh$, $h > 0$ for $k = 0, 1, \ldots, K$ (using simulation software, or measurements from real life experiment).

Impose an **expressive, yet simple** nonlinear structure.

Operator Inference

Basic Algorithm: "Empirical Model Reduction" [Kravtsov/Kondrashov/Ghil 2005]

Problem setting: infer (smooth) nonlinear dynamical system,

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0 \in \mathbb{R}^n.$$

Goal: learn dynamical system from given snapshots $x_k := x(t_k)$, $u_k := u(t_k)$, $t_k := kh$, $h > 0$ for $k = 0, 1, \ldots, K$ (using simulation software, or measurements from real life experiment).

Impose an **expressive, yet simple** nonlinear structure.

Here: try to infer quadratic-linear system

$$\dot{\hat{x}}(t) = \hat{A}\hat{x}(t) + \hat{H}\left(\hat{x}(t) \otimes \hat{x}(t)\right) + \hat{B}u(t),$$

where $P \otimes Q := [p_{ij}Q]_{ij}$ denotes the Kronecker (tensor) product, from data

$$X := [\, x_0, x_1, \ldots, x_K \,] \in \mathbb{R}^{n \times (K+1)}, \quad U := [\, u_0, u_1, \ldots, u_K \,] \in \mathbb{R}^{m \times (K+1)}.$$

Problem setting: infer (smooth) nonlinear dynamical system,

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0 \in \mathbb{R}^n.$$

Goal: learn dynamical system from given snapshots $x_k := x(t_k)$, $u_k := u(t_k)$, $t_k := kh$, $h > 0$ for $k = 0, 1, \ldots, K$ (using simulation software, or measurements from real life experiment).

Impose an **expressive, yet simple** nonlinear structure.

Here: try to infer quadratic-linear system

$$\dot{\hat{x}}(t) = \hat{A}\hat{x}(t) + \hat{H}\left(\hat{x}(t) \otimes \hat{x}(t)\right) + \hat{B}u(t),$$

where $P \otimes Q := \left[p_{ij}Q\right]_{ij}$ denotes the Kronecker (tensor) product, from data

$$X := [\, x_0, x_1, \ldots, x_K \,] \in \mathbb{R}^{n \times (K+1)}, \quad U := [\, u_0, u_1, \ldots, u_K \,] \in \mathbb{R}^{m \times (K+1)}.$$

- Requires snapshot matrix of time derivatives: if residuals $f(x_j, u_j)$ are available

  $$\dot{X} := [\, \dot{x}(0), \dot{x}(t_1), \ldots, \dot{x}(t_K) \,] = [\, f(x_0, u_0), f(x_1, u_1), \ldots, f(x_K, u_K) \,] \in \mathbb{R}^{n \times (K+1)},$$

  otherwise, approximate time-derivatives by finite differences $\rightsquigarrow \approx \dot{X}$.

Operator Inference

Basic Algorithm: "Empirical Model Reduction" [Kravtsov/Kondrashov/Ghil 2005]

Problem setting: infer (smooth) nonlinear dynamical system,

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0 \in \mathbb{R}^n.$$

Goal: learn dynamical system from given snapshots $x_k := x(t_k)$, $u_k := u(t_k)$, $t_k := kh$, $h > 0$ for $k = 0, 1, \ldots, K$ (using simulation software, or measurements from real life experiment).

Impose an **expressive, yet simple** nonlinear structure.

Here: try to infer quadratic-linear system

$$\dot{\hat{x}}(t) = \hat{A}\hat{x}(t) + \hat{H}\left(\hat{x}(t) \otimes \hat{x}(t)\right) + \hat{B}u(t),$$

where $P \otimes Q := [p_{ij}Q]_{ij}$ denotes the Kronecker (tensor) product, from data

$$X := [\, x_0, x_1, \ldots, x_K \,] \in \mathbb{R}^{n \times (K+1)}, \quad U := [\, u_0, u_1, \ldots, u_K \,] \in \mathbb{R}^{m \times (K+1)}.$$

- Requires snapshot matrix of time derivatives: if residuals $f(x_j, u_j)$ are available

  $$\dot{X} := [\, \dot{x}(0), \dot{x}(t_1), \ldots, \dot{x}(t_K) \,] = [\, f(x_0, u_0), f(x_1, u_1), \ldots, f(x_K, u_K) \,] \in \mathbb{R}^{n \times (K+1)},$$

  otherwise, approximate time-derivatives by finite differences $\rightsquigarrow \approx \dot{X}$.
- Solve the linear least-squares problem (regression):

  $$(A_*, H_*, B_*) := \operatorname{argmin}_{(A,H,B)} \left\| \dot{X} - \begin{bmatrix} A & H & B \end{bmatrix} \begin{bmatrix} X \\ X^2 \\ U \end{bmatrix} \right\|_F^2 + \mathcal{R}(A, H, B)$$

  with potential regularization $\mathcal{R}$ and $X^2 := [x_0 \otimes x_0, \ldots, x_K \otimes x_K]$.

(Regularized) **Operator Inference least-squares (OpInf)** problem

$$(A_*, H_*, B_*) := \operatorname{argmin}_{(A,H,B)} \left\| \dot{X} - \begin{bmatrix} A & H & B \end{bmatrix} \begin{bmatrix} X \\ X^2 \\ U \end{bmatrix} \right\|_F^2 + \mathcal{R}(A,\, H,\, B)$$

may be computationally too complex if state-space is too large (say, $n > 30$).

(Regularized) **Operator Inference least-squares (OpInf)** problem

$$(A_*, H_*, B_*) := \operatorname{argmin}_{(A,H,B)} \left\| \dot{X} - \begin{bmatrix} A & H & B \end{bmatrix} \begin{bmatrix} X \\ X^2 \\ U \end{bmatrix} \right\|_F^2 + \mathcal{R}(A,\, H,\, B)$$

may be computationally too complex if state-space is too large (say, $n > 30$).

Idea: compress trajectories using POD / PCA:

1. Let $X := [\, x_0, x_1, \ldots, x_{K-1}, x_K \,] \in \mathbb{R}^{n \times K+1}$ be the matrix of all snapshots.

(Regularized) **Operator Inference least-squares (OpInf)** problem

$$(A_*, H_*, B_*) := \operatorname{argmin}_{(A,H,B)} \left\| \dot{X} - \begin{bmatrix} A & H & B \end{bmatrix} \begin{bmatrix} X \\ X^2 \\ U \end{bmatrix} \right\|_F^2 + \mathcal{R}(A,\, H,\, B)$$

may be computationally too complex if state-space is too large (say, $n > 30$).

Idea: compress trajectories using POD / PCA:

1. Let $X := [\, x_0, x_1, \ldots, x_{K-1}, x_K \,] \in \mathbb{R}^{n \times K+1}$ be the matrix of all snapshots.
2. Compute principal / dominant singular vectors via SVD $X = Q \Sigma V^T$ and set $W := Q(:, 1 : r)$ such that $\sum_{k=r+1}^{K+1} \sigma_k < \varepsilon$ (potentially, use centered data).

(Regularized) **Operator Inference least-squares (OpInf)** problem

$$(A_*, H_*, B_*) := \operatorname{argmin}_{(A,H,B)} \left\| \dot{X} - \begin{bmatrix} A & H & B \end{bmatrix} \begin{bmatrix} X \\ X^2 \\ U \end{bmatrix} \right\|_F^2 + \mathcal{R}(A, H, B)$$

may be computationally too complex if state-space is too large (say, $n > 30$).

Idea: compress trajectories using POD / PCA:

1. Let $X := [\, x_0, x_1, \ldots, x_{K-1}, x_K \,] \in \mathbb{R}^{n \times K+1}$ be the matrix of all snapshots.
2. Compute principal / dominant singular vectors via SVD $X = Q\Sigma V^T$ and set $W := Q(:, 1:r)$ such that $\sum_{k=r+1}^{K+1} \sigma_k < \varepsilon$ (potentially, use centered data).
3. Compute compressed snapshot matrices $\hat{X} := W^T X$, $\hat{X}^2 := \hat{X} \otimes \hat{X}$, $\dot{\hat{X}} := W^T \dot{X}$.

Operator Inference

Full algorithm with data compression [Peherstorfer/Willcox 2016]

(Regularized) **Operator Inference least-squares (OpInf)** problem

$$(A_*, H_*, B_*) := \operatorname{argmin}_{(A,H,B)} \left\| \dot{X} - \begin{bmatrix} A & H & B \end{bmatrix} \begin{bmatrix} X \\ X^2 \\ U \end{bmatrix} \right\|_F^2 + \mathcal{R}(A,\, H,\, B)$$

may be computationally too complex if state-space is too large (say, $n > 30$).

Idea: compress trajectories using POD / PCA:

1. Let $X := [\, x_0, x_1, \ldots, x_{K-1}, x_K \,] \in \mathbb{R}^{n \times K+1}$ be the matrix of all snapshots.
2. Compute principal / dominant singular vectors via SVD $X = Q \Sigma V^T$ and set $W := Q(:, 1:r)$ such that $\sum_{k=r+1}^{K+1} \sigma_k < \varepsilon$ (potentially, use centered data).
3. Compute compressed snapshot matrices $\hat{X} := W^T X$, $\hat{X}^2 := \hat{X} \otimes \hat{X}$, $\dot{\hat{X}} := W^T \dot{X}$.
4. Apply OpInf using $\hat{X}, \hat{X}^2, \dot{\hat{X}}$ and compute reduced-order model via

$$(\hat{A}_*, \hat{H}_*, \hat{B}_*) := \operatorname{argmin}_{(\hat{A},\hat{H},\hat{B})} \left\| \dot{\hat{X}} - \begin{bmatrix} \hat{A} & \hat{H} & \hat{B} \end{bmatrix} \begin{bmatrix} \hat{X} \\ \hat{X}^2 \\ U \end{bmatrix} \right\|_F^2 + \mathcal{R}(\hat{A},\, \hat{H},\, \hat{B}).$$

- Parameterized shallow water equations are given by [Yildiz et al 2021]

$$\frac{\partial}{\partial t}\tilde{u} = -h_x + \sin\theta\ \tilde{v} - \tilde{u}\tilde{u}_x - \tilde{v}\tilde{u}_y + \delta\cos\theta(h\tilde{u})_x - \frac{3}{8}\left(\delta\cos\theta\right)^2\left(h^2\right)_x,$$

$$\frac{\partial}{\partial t}\tilde{v} = -h_y + \sin\theta\ \tilde{u} + \frac{1}{2}\delta\sin\theta\cos\theta\ h - \tilde{u}\tilde{v}_x - \tilde{v}\tilde{v}_y$$

$$+\delta\cos\theta\left((h\tilde{u})_y + \frac{1}{2}h\left(\tilde{v}_x - \tilde{u}_y\right)\right) - \frac{3}{8}\left(\delta\cos\theta\right)^2\left(h^2\right)_y,$$

$$\frac{\partial}{\partial t}h = -(h\tilde{u})_x - (h\tilde{v})_y + \frac{1}{2}\delta\cos\theta(h^2)_x.$$

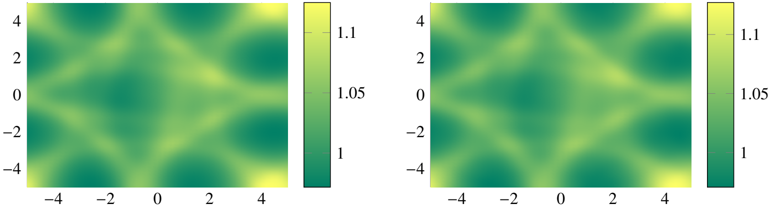
- Parameterized by the latitude θ.
- $\tilde{\mathbf{u}} =: (\tilde{u};\tilde{v})$ is the canonical velocity.
- h is the height field.
- We collect the training data for 5 different parameter realizations θ in $\left[\frac{\pi}{6},\frac{\pi}{3}\right]$.
- Infer a reduced parametric model of order $r = 75$ directly from data.

Operator Inference: Numerical Example
Parameterized Shallow Water Equations

- **Parameterized shallow water equations** are given by [Yildiz et al 2021]

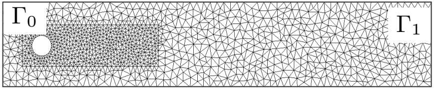
$$\frac{\partial}{\partial t}\tilde{u} = -h_x + \sin\theta\ \tilde{v} - \tilde{u}\tilde{u}_x - \tilde{v}\tilde{u}_y + \delta\cos\theta(h\tilde{u})_x - \frac{3}{8}\left(\delta\cos\theta\right)^2\left(h^2\right)_x,$$
$$\frac{\partial}{\partial t}\tilde{v} = -h_y + \sin\theta\ \tilde{u} + \frac{1}{2}\delta\sin\theta\cos\theta\ h - \tilde{u}\tilde{v}_x - \tilde{v}\tilde{v}_y$$
$$+\,\delta\cos\theta\left((h\tilde{u})_y + \frac{1}{2}h\left(\tilde{v}_x - \tilde{u}_y\right)\right) - \frac{3}{8}\left(\delta\cos\theta\right)^2\left(h^2\right)_y,$$
$$\frac{\partial}{\partial t}h = -(h\tilde{u})_x - (h\tilde{v})_y + \frac{1}{2}\delta\cos\theta(h^2)_x.$$

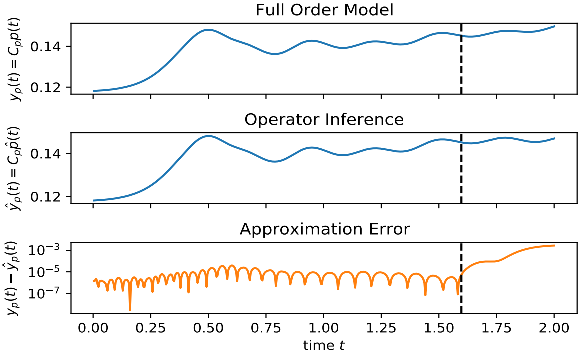
- Comparison of the height field for the parameter $\theta = \dfrac{5\pi}{24}$:



(a) FOM (b) Learned parametric model

Tailored operator inference for incompressible Navier-Stokes equations, by heeding incompressibility condition. [B./Goyal/Heiland/Pontes Duff 2022]





Asymptotic (exponential, Lyapunov) stability of linear systems

$$\dot{x}(t) = Ax(t), \qquad x(0) = x_0,$$

can be explicitly parameterized:

Theorem (Gillis/Sharma 2017)

A matrix $A \in \mathbb{R}^{n \times n}$ is asymptotically stable (Hurwitz, Lyapunov stable) if and only if it can be represented as

$$A = (J - R)Q,$$

where $J = -J^T$ and $R = R^T$, $Q = Q^T$ are both positive definite.

Asymptotic (exponential, Lyapunov) stability of linear systems

$$\dot{x}(t) = Ax(t), \qquad x(0) = x_0,$$

can be explicitly parameterized:

Theorem (Gillis/Sharma 2017)

A matrix $A \in \mathbb{R}^{n \times n}$ is asymptotically stable (Hurwitz, Lyapunov stable) if and only if it can be represented as

$$A = (J - R)Q,$$

where $J = -J^T$ and $R = R^T$, $Q = Q^T$ are both positive definite.

$\Longrightarrow$ **Stability-preserving OpInf for linear systems** [Goyal/Pontes Duff/B. 2023]:

$$(S_*, L_*, K_*) := \operatorname{argmin}_{\substack{L, K \text{ upper triangular} \\ \text{with positive diagonals}}} \left(\|\dot{X} - (S - S^T - L^T L) K^T K X\|_F^2 + \mathcal{R}(L, K, S) \right).$$

The matrix obtained from this nonlinear (regularized) least-squares problem,

$$A_* = \left(S_* - S_*^T - L_*^T L_* \right) K_*^T K_*,$$

is guaranteed to be stable due to [Gillis/Sharma 2017].

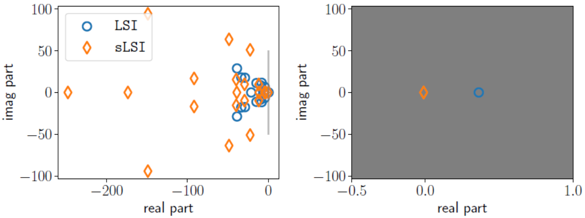
Related work by *Schwerdtner, Voigt, ...*

Consider 1D Burgers' equation for viscous flow

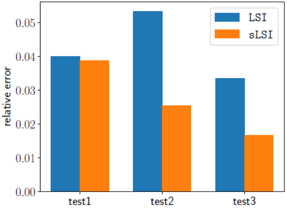
$$\begin{aligned} v_t + v v_x &= \nu v_{xx} \text{ in } (0,1) \times (0,T) \\ v_x(0,t) &= v_x(1,t) = 0, \\ v(x,0) &= v_0(x,\mu), \end{aligned}$$

discretized on uniform 1000×500 space-time grid for $17 + 3$ training+testing initial conditions.

Reduced-order model ($r = 21$) computed using standard ("LSI") and stabilized ("SLSI") OpInf applied to (POD)-projected data.
(Implementation using `PyTorch` and `Adam` optimizer for solving nonlinear regression problem.)



Eigenvalues of linearization



Errors for different initial conditions (test data)

Solving the OpInf regression problem

$$(A_*, H_*) := \operatorname{argmin}_{(A,H)} \left\| \dot{X} - \begin{bmatrix} A & H \end{bmatrix} \begin{bmatrix} X \\ X^2 \end{bmatrix} \right\|_F^2 + \mathcal{R}(A\,H)$$

using the stability-constraint on A as just discussed leads to a nonlinear system with local Lyapunov stability, noting that the inferred $Q_* = K_*^T K_* > 0$ provides a quadratic Lyapunov function for the identified system [Goyal/Pontes Duff/B. 2023].

Solving the OpInf regression problem

$$(A_*, H_*) := \operatorname{argmin}_{(A,H)} \left\| \dot{X} - \begin{bmatrix} A & H \end{bmatrix} \begin{bmatrix} X \\ X^2 \end{bmatrix} \right\|_F^2 + \mathcal{R}(A\,H)$$

using the stability-constraint on A as just discussed leads to a nonlinear system with local Lyapunov stability, noting that the inferred $Q_* = K_*^T K_* > 0$ provides a quadratic Lyapunov function for the identified system [Goyal/Pontes Duff/B. 2023].

We can achieve more for energy-preserving quadratic systems, i.e.,

$$H_{ijk} + H_{ikj} + H_{jik} + H_{jki} + H_{kij} + H_{kji} = 0 \quad \text{for all } i, j, k \in \{1, \ldots, n\}.$$

Note: the latter is equivalent to $x^T H(x \otimes x) = 0$ for all $x \in \mathbb{R}^n$ [Schlegel/Noack 2015].

Solving the OpInf regression problem

$$(A_*, H_*) := \operatorname{argmin}_{(A,H)} \left\| \dot{X} - \begin{bmatrix} A & H \end{bmatrix} \begin{bmatrix} X \\ X^2 \end{bmatrix} \right\|_F^2 + \mathcal{R}(A\,H)$$

using the stability-constraint on A as just discussed leads to a nonlinear system with local Lyapunov stability, noting that the inferred $Q_* = K_*^T K_* > 0$ provides a quadratic Lyapunov function for the identified system [Goyal/Pontes Duff/B. 2023].

We can achieve more for energy-preserving quadratic systems, i.e.,

$$H_{ijk} + H_{ikj} + H_{jik} + H_{jki} + H_{kij} + H_{kji} = 0 \quad \text{for all } i, j, k \in \{1, \ldots, n\}.$$

Note: the latter is equivalent to $x^T H(x \otimes x) = 0$ for all $x \in \mathbb{R}^n$ [Schlegel/Noack 2015].

Theorem (Goyal/Pontes Duff/B. 2023)

An energy-preserving quadratic system

$$\dot{z} = Az + H(z \otimes z)$$

is monotonically and globally asymptotically stable if and only if the symmetric part of A is asymptotically stable.

Theorem (Goyal/Pontes Duff/B. 2023)

An energy-preserving quadratic system

$$\dot{z} = Az + H(z \otimes z)$$

is monotonically and globally asymptotically stable (GAS) if and only if the symmetric part of A is asymptotically stable.

Theorem (Goyal/Pontes Duff/B. 2023)

An energy-preserving quadratic system

$$\dot{z} = Az + H(z \otimes z)$$

is monotonically and globally asymptotically stable (GAS) if and only if the symmetric part of A is asymptotically stable.

Question: can we encode the energy-preservation property explicitly, so that we constrain the OpInf problem accordingly? (If the answer is yes, then we can learn a GAS model using OpInf.)

Answer: yes, we can!

Theorem (Goyal/Pontes Duff/B. 2023)

A locally Lyapunov stable quadratic system in $\mathbb{R}^n$

$$\dot{z} = Az + H(z \otimes z), \qquad A = (J - R)Q,\ J = -J^T,\ R = R^T > 0,\ Q = Q^T > 0,$$

is generalized energy-preserving w.r.t. Q, i.e., $x^T QH(x \otimes x) = 0$ for all x, if

$$H = [H_1 Q, \ldots, H_n Q], \quad \text{where} \quad H_j = -H_j^T, \quad j = 1, \ldots, n.$$

Moreover, $V(x) = \frac{1}{2} x^T Q x$ is a global Lyapunov function for the quadratic system.

Note: the converse is true, too! [Gkimisis/Pontes Duff/Goyal/B. 2025]

Constrained OpInf problem for learning GAS systems [Goyal/Pontes Duff/B. 2023]

$$(A_*, H_*) := \operatorname{argmin}_{(A,H)} \left\| \dot{X} - \begin{bmatrix} A & H \end{bmatrix} \begin{bmatrix} X \\ X^2 \end{bmatrix} \right\|_F^2 + \mathcal{R}(A, H)$$

subject to the stability constraints

$A = \left(S - S^T - L^T L\right) K^T K$ with L, K upper triangular with positive diagonals

$H = [H_1 Q, \ldots, H_n Q]$, with $H_j = -H_j^T$, $j = 1, \ldots, n$.

Constrained OpInf problem for learning GAS systems [Goyal/Pontes Duff/B. 2023]

$$(A_*, H_*) := \operatorname{argmin}_{(A,H)} \left\| \dot{X} - \begin{bmatrix} A & H \end{bmatrix} \begin{bmatrix} X \\ X^2 \end{bmatrix} \right\|_F^2 + \mathcal{R}(A, H)$$

subject to the stability constraints

$$A = \left(S - S^T - L^T L\right) K^T K \quad \text{with } L, K \text{ upper triangular with positive diagonals}$$

$$H = [H_1 Q, \ldots, H_n Q], \quad \text{with} \quad H_j = -H_j^T, \quad j = 1, \ldots, n.$$

Implementation:

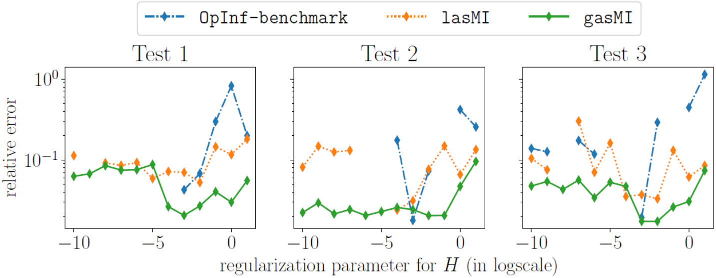
- Usually, as discussed before, the data are projected onto the leading r PCA modes for dimension reduction.
- Quite involved optimization problem, can be solved via stochastic gradient descent (Adam) and backpropagation (setting $Q = I_r$ may be necessary).
- We do not explicitly need derivative data by using a Neural ODE approach for noisy data [Goyal/B. 2023].

Consider again 1D Burgers' equation for viscous flow

$$\begin{aligned} v_t + v v_x &= \nu v_{xx} \text{ in } (0,1)\times(0,T) \\ v(0,t) &= v(1,t) = 0, \\ v(x,0) &= v_0(x,\mu), \end{aligned}$$

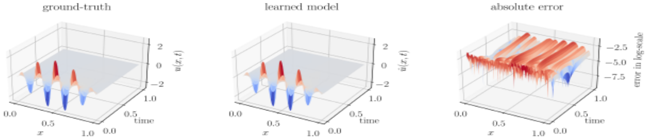
discretized on uniform 250×500 space-time grid for $17+3$ training+testing initial conditions and $\nu = 0.05$.

Reduced-order model ($r = 20$) computed using standard, locally stable (lasMI) and globally stable (gasMI) OpInf applied to (POD)-projected data.
(Implementation using `PyTorch` and `Adam` optimizer for solving nonlinear regression problem.)

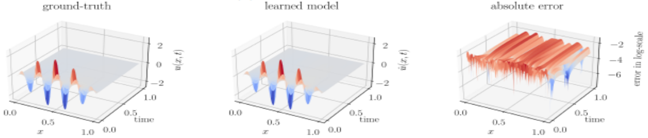


(missing data = blow-up in numerical simulation)

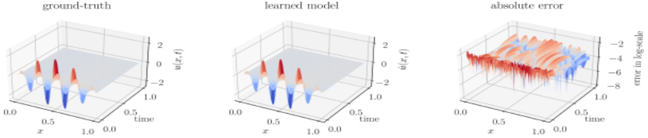
Consider again 1D Burgers' equation for viscous flow



(a) Using **opInf-benchmark**.



(b) Using **lasMI**.



(c) Using **gasMI**.

Full simulation for test initial condition (not seen during training)

- So far, we considered asymptotically stable systems.

- So far, we considered asymptotically stable systems.
- However, there exist quadratic systems without any stable points, e.g., chaotic Lorenz example.

- So far, we considered asymptotically stable systems.
- However, there exist quadratic systems without any stable points, e.g., chaotic Lorenz example.
- Despite having no stable point, these systems might have an attractor, meaning there exists a bounded region (a ball) where all trajectories for some set of initial conditions get trapped. (Attractor is sometimes also called "trapping region".) Call such systems ATR systems.

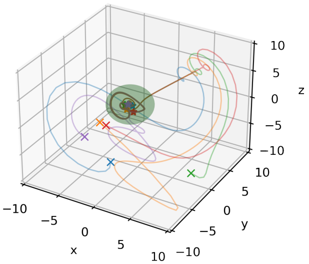


Figure: An illustration of nonlinear dynamics with attractor.

Preserving Stability in Operator Inference

Nonlinear Dynamics with Attractor

- So far, we considered asymptotically stable systems.
- However, there exist quadratic systems without any stable points, e.g., chaotic Lorenz example.
- Despite having no stable point, these systems might have an attractor, meaning there exists a bounded region (a ball) where all trajectories for some set of initial conditions get trapped. (Attractor is sometimes also called "trapping region".) Call such systems ATR systems.

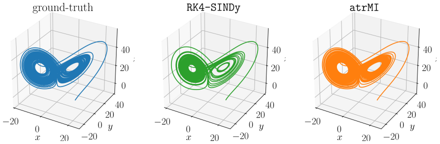
Inference of ATR quadratic systems [Goyal/Pontes Duff/B. 2023]

- For energy-preserving quadratic systems, an ATR system can be turned into a GAS system by translation $x(t) \to x(t) - y$
- We, thus, require to solve the following constraint problem:

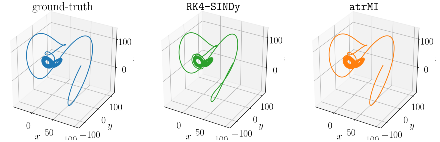
$$\min_{A,H,y} \left\| \dot{X} - A(X - y) - H(X - y)^2 \right\|$$

subject to $\Lambda(A) \in \mathbb{C}^-$ and H is energy preserving.

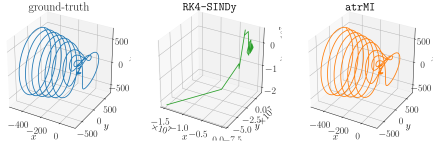
- Note that we do not know y a priori, it is learned from the data.
- The radius r can be computed based on the minimum eigenvalues of $\mathbf{A}$.



(a) For initial condition $[10, 10, -10]$.



(b) For initial condition $[100, -100, 100]$.



(c) For initial condition $[-500, 500, 500]$.

Conclusions

- OpInf is a regression-based powerful method to infer linear and certain nonlinear dynamical systems from data. Looks simple, but devil is in the details.
- Stability constraints can be encoded explicitly in the regression problem for the model inference.
- Extension to nonlinear systems with attractor [Goyal/Pontes Duff/B. 2023].
- For application to control problems ("BIBO stability"), see [Pontes Duff/Goyal/B. 2024].
- For application to parametric problems, see [Mamidisetti/Pontes Duff/Goyal/B. 2025].
- The same approach can also be used to infer stable systems from a richer (than just quadratics) dictionary using sparse regression (SINDy).
- Recent work combines OpInf with neural networks to solve nonlinear identification problems.
- Applications to surrogate modeling for Digital Twins of, e.g., energy conversion processes show promising results when stability encoding is used.
- Error bounds for non-intrusive MOR not well developed yet, but theoretic results indicate that the OpInf model asymptotically (when increasing the number of snapshots) yields the POD model. Then, intrusive MOR error bounds can be applied.

References

Kravtsov, S., Kondrashov, D., Ghil, M. (2005): Multilevel regression modeling of nonlinear processes: Derivation and applications to climatic variability. J. Climate, 18(21):4404–4424.

Peherstorfer, B., Willcox, K. (2016): Data-driven operator inference for nonintrusive projection-based model reduction. Comput. Methods Appl. Mech. Eng. 306:196–215.

Brunton, B.W., Johnson, L.A., Ojemann, J.G., Kutz, J.N. (2016): Extracting spatial-temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition. J. Neurosci. Methods 258:1–15.

Annoni, J., Seiler, P. (2017): A method to construct reduced-order parameter-varying models. Int. J. Robust Nonlinear Control 27(4):582–597.

Qian, E., Kramer, B., Peherstorfer, B., Willcox, K. (2020): Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems. Physica D: Nonlinear Phenomena 406:132401.

Benner, P., Goyal, P., Kramer, B., Peherstorfer, B., Willcox, K. (2020): Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms. Comp. Meth. Appl. Mech. Eng., 372:113433.

Yıldız, S., Goyal, P., Benner, P., Karasozen, B. (2021): Learning reduced-order dynamics for parametrized shallow water equations from data. Int. J. Numer. Meth. Eng., 93(8):2803–2821.

Benner, P., Goyal, P., Heiland, J., Pontes Duff, I. (2022): Operator inference and physics-informed learning of low-dimensional models for incompressible flows. Elec. Trans. Numer. Anal., 56:28–51.

Goyal, P., Pontes Duff, I., Benner, P. (2023): Inference of continuous linear systems from data with guaranteed stability. arXiv:2301.10060

Goyal, P., Pontes Duff, I., Benner, P. (2023): Guaranteed stable quadratic models and their applications in SINDy and operator inference. arXiv:2308.13819

Pontes Duff, I., Goyal, P., Benner, P. (2024): Stability-certified learning of control systems with quadratic nonlinearities. Proc. MTNS 2024 / IFAC-PapersOnLine, 58(17):151–156, 2024.

Gkimisis, L., Pontes Duff, I., Goyal, P., Benner, P. (2025): On the representation of energy-preserving quadratic operators with application to operator inference. arXiv: 2503.10824