

Linear and nonlinear system identification

Theoretical principles

*CEA-EDF-INRIA Summer School
June 2025, EDF Lab Saclay*

Jean-Christophe Loiseau

✉ jean-christophe.loiseau@ensam.eu
🌐 loiseaujc.github.io



Assistant Professor of Applied Mathematics
and Fluid Dynamics

Research activities

Hydrodynamic instabilities
Transition to turbulence
Reduced-order modeling
Optimal control

Tools

High-performance computing
Numerical Linear Algebra
Convex optimization

Non-exhaustive list of challenges

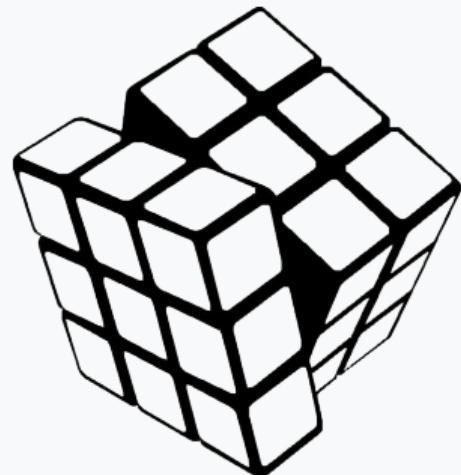
- Physics**
- What is physically realizable?
 - What are the operating conditions?
 - Which mechanisms to leverage?
 - How to identify them?
 - How generic are they?



Non-exhaustive list of challenges

Instrumentation

- What is actually *observable*?
- What kind of sensors can I use?
- Where to place them and how many?
- How to handle measurement noise?
- What if one of my sensor fail?



Non-exhaustive list of challenges

Instrumentation

- What is actually *controllable*?
- What kind of actuators can I use?
- Where to place them and how many?
- How to handle disturbances?
- What if one of my actuator fail?



Non-exhaustive list of challenges

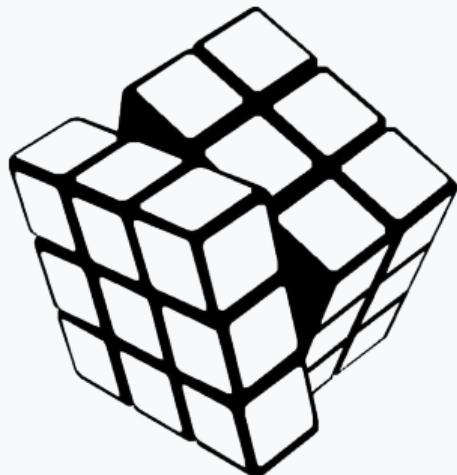
- Control**
- What are the specifications?
 - How robust does it need to be?
 - Do I have hard operating constraints?
 - On-board or external computations?
 - Do I need any form of certification?



Non-exhaustive list of challenges

Modelling

- What kind of model?
- Physics-based or data-driven?
- How accurate does it need to be?
- Can I quantify the uncertainties?
- How expensive is it simulate?



Linear System Identification

A brief overview

$$\begin{aligned} & \text{Known input sequence} \\ & \downarrow \\ & \mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t \\ & \mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \mathbf{D}\mathbf{u}_t. \\ & \uparrow \\ & \text{Measured output sequence} \end{aligned}$$

Kalman decomposition

$$\begin{bmatrix} \mathbf{x}_{r\bar{o}} \\ \mathbf{x}_{ro} \\ \mathbf{x}_{\bar{r}o} \\ \mathbf{x}_{\bar{\bar{r}}o} \end{bmatrix}_{t+1} = \begin{bmatrix} \mathbf{A}_{r\bar{o}} & * & * & * \\ & \mathbf{A}_{ro} & * & * \\ & & \mathbf{A}_{\bar{r}o} & * \\ & & & \mathbf{A}_{\bar{\bar{r}}o} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{r\bar{o}} \\ \mathbf{x}_{ro} \\ \mathbf{x}_{\bar{r}o} \\ \mathbf{x}_{\bar{\bar{r}}o} \end{bmatrix}_t + \begin{bmatrix} \mathbf{B}_{r\bar{o}} \\ \mathbf{B}_{ro} \\ \mathbf{o} \\ \mathbf{o} \end{bmatrix} u_t$$

$$y_t = [0 \quad C_{ro} \quad 0 \quad C_{\bar{r}o}] \begin{bmatrix} x_{r\bar{o}} \\ x_{ro} \\ x_{\bar{r}o} \\ x_{\bar{\bar{r}}o} \end{bmatrix}_t + Du_t$$

Kalman decomposition

$$\begin{bmatrix} \mathbf{x}_{r\bar{o}} \\ \mathbf{x}_{ro} \\ \mathbf{x}_{\bar{r}o} \\ \mathbf{x}_{\bar{\bar{r}}o} \end{bmatrix}_{t+1} = \begin{bmatrix} \mathbf{A}_{r\bar{o}} & * & * & * \\ * & \mathbf{A}_{ro} & * & * \\ \mathbf{A}_{\bar{r}o} & * & \mathbf{A}_{\bar{o}} & * \\ \mathbf{A}_{\bar{\bar{r}}o} & * & * & \mathbf{A}_{\bar{\bar{o}}} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{r\bar{o}} \\ \mathbf{x}_{ro} \\ \mathbf{x}_{\bar{r}o} \\ \mathbf{x}_{\bar{\bar{r}}o} \end{bmatrix}_t + \begin{bmatrix} \mathbf{B}_{r\bar{o}} \\ \mathbf{B}_{ro} \\ \mathbf{B}_{\bar{r}o} \\ \mathbf{B}_{\bar{\bar{r}}o} \end{bmatrix} \mathbf{u}_t$$

In practice, only a realization of the jointly observable and controllable subsystem $(\mathbf{A}_{ro}, \mathbf{B}_{ro}, \mathbf{C}_{ro}, \mathbf{D})$ can be obtained from data.

The realization of the jointly controllable and observable subsystem is not unique.

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t$$

$$\mathbf{y}_t = \mathbf{Cx}_t + \mathbf{Du}_t$$

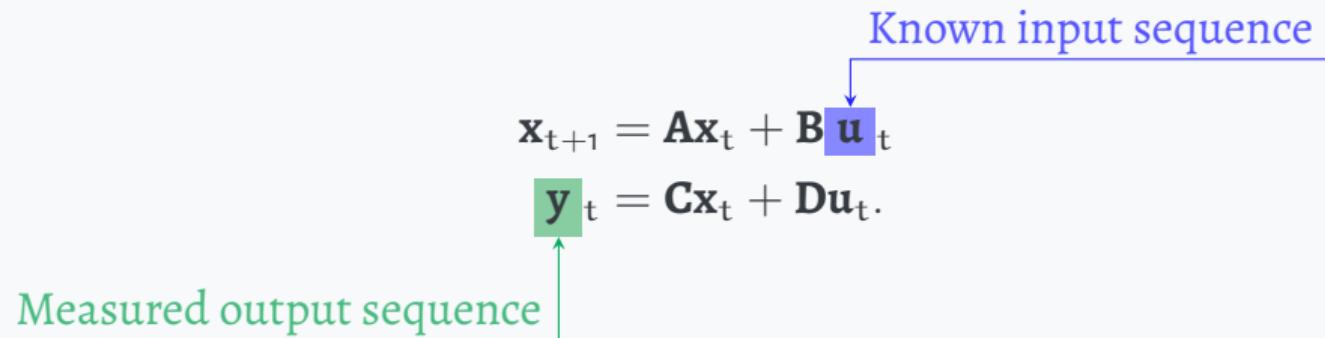
$$\mathbf{z}_{t+1} = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}\mathbf{z}_t + \mathbf{T}^{-1}\mathbf{B}\mathbf{u}_t$$

$$\mathbf{y}_t = \mathbf{C}\mathbf{T}\mathbf{z}_t + \mathbf{D}\mathbf{u}_t$$

Both models have the same I/O behavior and are thus equivalent representations of the same system.

Linear System Identification

Realization theory



Given the (yet unknown) states at time $t = 1$ and $t = 2$, we can write

$$\mathbf{x}_2 = \mathbf{Ax}_1 + \mathbf{Bu}_1, \quad \mathbf{x}_3 = \mathbf{Ax}_2 + \mathbf{Bu}_2, \quad \mathbf{x}_4 = \mathbf{Ax}_3 + \mathbf{Bu}_3,$$

or alternatively

$$\mathbf{x}_3 = \mathbf{A}^2\mathbf{x}_1 + \mathbf{ABu}_1 + \mathbf{Bu}_2, \quad \mathbf{x}_4 = \mathbf{A}^2\mathbf{x}_2 + \mathbf{ABu}_2 + \mathbf{Bu}_3$$

leading to

$$\begin{bmatrix} \mathbf{x}_3 & \mathbf{x}_4 \end{bmatrix} = \mathbf{A}^2 \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix} + [\mathbf{AB} \quad \mathbf{B}] \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 \\ \mathbf{u}_2 & \mathbf{u}_3 \end{bmatrix}.$$

$$\begin{bmatrix} \mathbf{y}_k & \mathbf{y}_{k+1} \\ \mathbf{y}_{k+1} & \mathbf{y}_{k+2} \end{bmatrix} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k & \mathbf{x}_{k+1} \end{bmatrix} + \begin{bmatrix} \mathbf{D} \\ \mathbf{CB} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{u}_k & \mathbf{u}_{k+1} \\ \mathbf{u}_{k+1} & \mathbf{u}_{k+2} \end{bmatrix}$$

Past data

$$\mathbf{X}_p = \mathcal{O}^\dagger (\mathbf{Y}_p - \mathbf{H}\mathbf{U}_p)$$

Future data

$$\mathcal{O}\mathbf{X}_f = \mathbf{Y}_f - \mathbf{H}\mathbf{U}_f$$

$$\underline{\text{Future output}} \quad \underline{\text{Past output}}$$
$$\mathbf{Y}_f - \mathbf{H} \mathbf{U}_f = \mathcal{O} \mathbf{A}^k \mathcal{O}^\dagger \mathbf{Y}_p + \mathcal{O} (\Delta - \mathbf{A}^k \mathcal{O}^\dagger \mathbf{H}) \mathbf{U}_p$$
$$\begin{matrix} \text{Future input} \\ \uparrow \\ \mathbf{U}_f \end{matrix} \quad \begin{matrix} \text{Past input} \\ \uparrow \\ \mathbf{U}_p \end{matrix}$$

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{k-1} \end{bmatrix} \quad \mathbf{H} = \begin{bmatrix} \mathbf{D} & & & & \\ \mathbf{CB} & \mathbf{D} & & & \\ \mathbf{CAB} & \mathbf{CB} & \mathbf{D} & & \\ \mathbf{CA}^2\mathbf{B} & \mathbf{CAB} & \mathbf{CB} & \mathbf{D} & \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ \mathbf{CA}^{k-1}\mathbf{B} & \mathbf{CA}^{k-2}\mathbf{B} & \cdots & \mathbf{CAB} & \mathbf{CB} & \mathbf{D} \end{bmatrix}$$

$$\Delta = [\mathbf{A}^{k-1}\mathbf{B} \quad \mathbf{A}^{k-2}\mathbf{B} \quad \cdots \quad \mathbf{AB} \quad \mathbf{B}]$$

The problem we'd like to solve reads

$$\text{minimize} \quad \|\mathbf{Y}_f - \mathbf{H}\mathbf{U}_f - \mathcal{O}\mathbf{A}^k\mathcal{O}^\dagger\mathbf{Y}_p + \mathcal{O}(\Delta - \mathbf{A}^k\mathcal{O}^\dagger\mathbf{H})\mathbf{U}_p\|_F^2$$

It however is a non-convex problem.

Let $\mathbf{W} = \mathbf{I} - \mathbf{U}_f^T (\mathbf{U}_f \mathbf{U}_f^T)^{-1} \mathbf{U}_f$ and k large enough, the problem reduces to

$$\begin{aligned} & \text{minimize } \left\| \mathbf{M}^{\frac{1}{2}} (\mathbf{Y}_f - \mathcal{O}\Delta \mathbf{U}_p) \mathbf{W}^{\frac{1}{2}} \right\|_F^2 \\ & \text{subject to } \text{rank}(\mathcal{O}\Delta) = r \end{aligned}$$

which can be cast as a *Quadratic Program with Quadratic equality constraints* admitting a closed-form solution.

$$\begin{aligned} & \underset{\mathcal{O}, \Delta}{\text{minimize}} \quad \frac{1}{2} \text{Tr} (\Delta^T \mathbf{U}_p^T \mathbf{W} \mathbf{U}_p \Delta) - \text{Tr} (\mathcal{O}^T \mathbf{M} \mathbf{Y}_f^T \mathbf{W} \mathbf{U}_p \Delta) \\ & \text{subject to} \quad \mathcal{O}^T \mathbf{M} \mathcal{O} = \mathbf{I}_r \end{aligned}$$

Quadratic Program with Quadratic Equality constraints

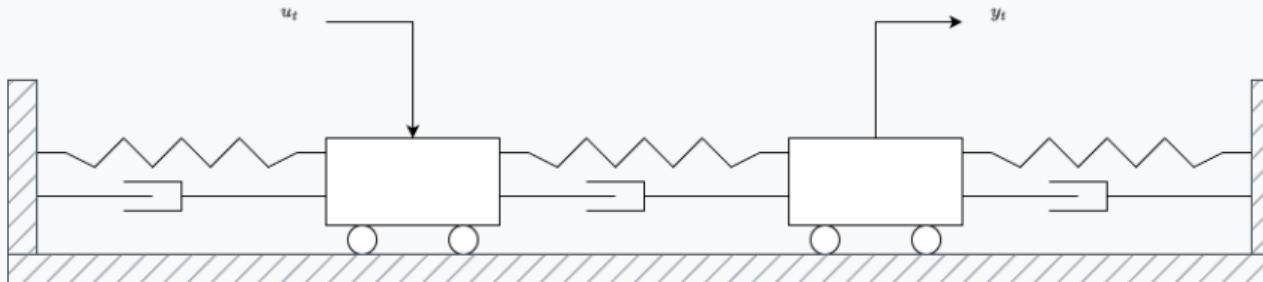
$$\begin{aligned}
 & \underset{\boldsymbol{\theta}, \Delta}{\text{minimize}} \quad \text{Tr} \left(\begin{bmatrix} \boldsymbol{\theta}^\top & \Delta^\top \end{bmatrix} \begin{bmatrix} \mathbf{0} & -\mathbf{M}\mathbf{Y}_f\mathbf{W}\mathbf{U}_p \\ -\mathbf{U}_p^\top \mathbf{W}\mathbf{Y}_f\mathbf{M} & \mathbf{U}_p\mathbf{W}\mathbf{U}_p \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta} \\ \Delta \end{bmatrix} \right) \\
 & \text{subject to } \begin{bmatrix} \boldsymbol{\theta}^\top & \Delta^\top \end{bmatrix} \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta} \\ \Delta \end{bmatrix} = \mathbf{I}_r
 \end{aligned}$$

Trace minimization of a Hermitian matrix

$$\begin{bmatrix} \mathbf{M} & \mathbf{o} \\ \mathbf{o} & \mathbf{o} \end{bmatrix} \begin{bmatrix} \mathcal{O} \\ \Delta \end{bmatrix} \boldsymbol{\Lambda} = \begin{bmatrix} \mathbf{o} & -\mathbf{M}\mathbf{Y}_f\mathbf{W}\mathbf{U}_p \\ -\mathbf{U}_p^T\mathbf{W}\mathbf{Y}_f\mathbf{M} & \mathbf{U}_p\mathbf{W}\mathbf{U}_p \end{bmatrix} \begin{bmatrix} \mathcal{O} \\ \Delta \end{bmatrix}$$

Generalized eigenvalue problem

	N4SID	MOESP	CVA
M	I	I	$(\mathbf{Y}_f \Pi_U^\perp \mathbf{Y}_f^\top)^{-1}$
W	?	Π_{U^\perp}	Π_U^\perp



LINEAR SYSTEM IDENTIFICATION
oooooooooooooooooooo●oooooooooooooooooooo

NONLINEAR SYSTEM IDENTIFICATION
oooooo

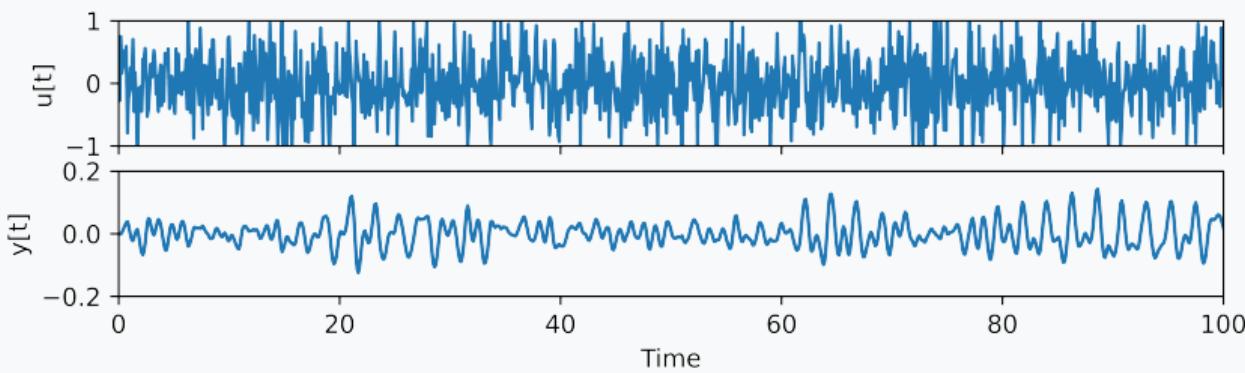
SINDY FOR ODE
oooooooooooo

FINDING PDES
oooo

SINDY FOR SDE
oooo

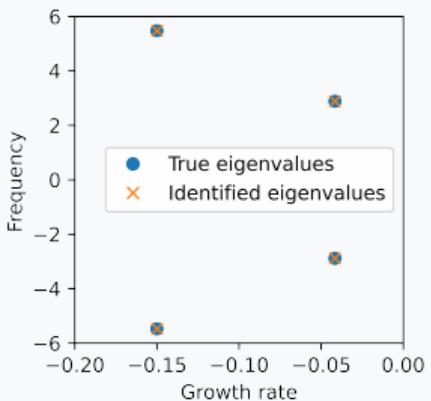
REINF. LEARNING
oo

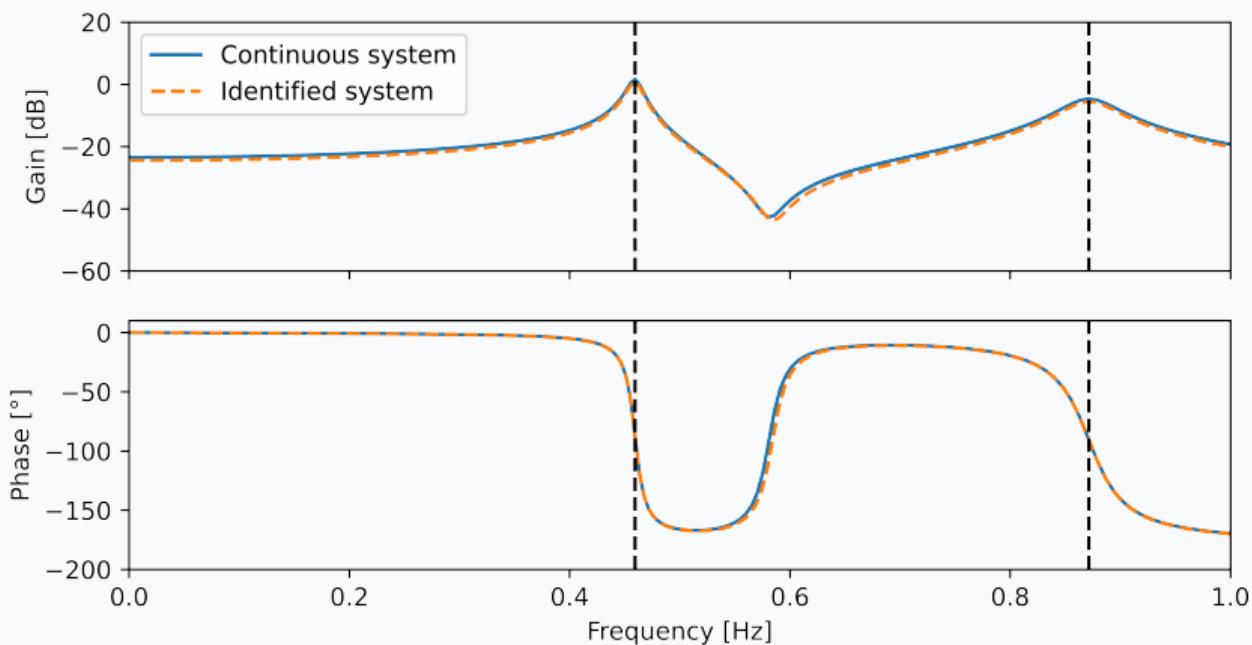
CONCLUSION
ooo



$$\mathbf{A} = \begin{bmatrix} 0.95 & -0.28 & -0.01 & 0.03 \\ 0.28 & 0.95 & -0.03 & 0.005 \\ -0.007 & 0.017 & 0.84 & 0.52 \\ -0.01 & 0.002 & -0.49 & 0.84 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0.054 \\ -0.03 \\ 0.05 \\ 0.03 \end{bmatrix}, \quad \mathbf{C} = [0.1 \quad 0.07 \quad 0.2 \quad -0.12]$$





Model Predictive Control

minimize $\frac{1}{2} \sum_{t=0}^{\tau} \|y_t\|_Q^2 + \|u_t\|_R^2$

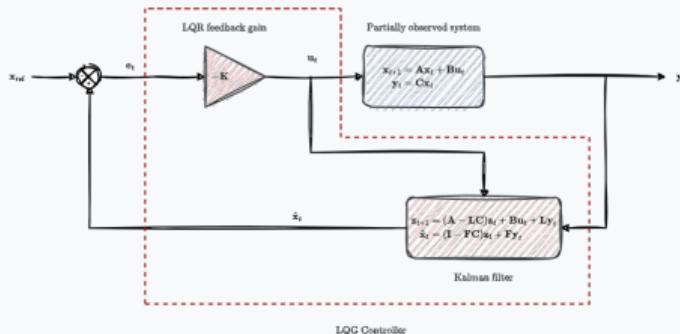
subject to $\begin{aligned}x_{t+1} &= Ax_t + Bu_t \\y_t &= Cx_t + Du_t \\y_t &\in \mathcal{Y}_t \\u_t &\in \mathcal{U}_t\end{aligned}$

- Golden standard in industry-grade control.
- Very flexible framework w/ excellent performances.
- If \mathcal{Y}_t and \mathcal{U}_t are *convex sets*, the whole optimization problem is convex.

Linear System Identification

A behavioral approach

Single Input/Single Output setup



Hypothesis

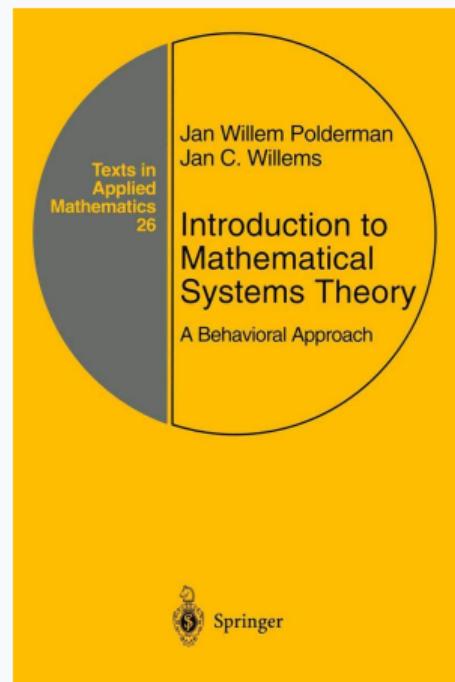
- The true system Σ is unknown.
- Its input-output dynamics can be well approximated by an LTI model.
- The rank of the model is unknown.
- Only input-output data are available.

A behavioral view of LTI systems

Definition

A discrete-time **dynamical system** is defined by a 3-tuple $\Sigma = (\mathbb{Z}_+, \mathbb{W}, \mathcal{B})$ where

1. \mathbb{Z}_+ is the discrete-time axis.
2. \mathbb{W} is the signal space.
3. $\mathcal{B} \subseteq \mathbb{W}^{\mathbb{Z}_+}$ is the *behavior*.

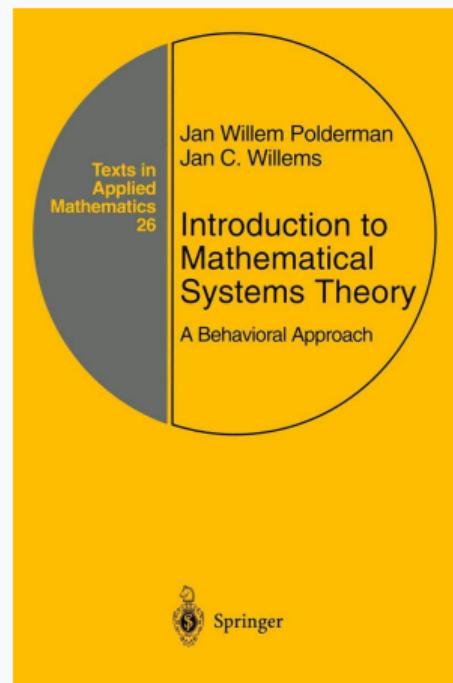


A behavioral view of LTI systems

Definition

The dynamical system $\Sigma = (\mathbb{Z}_+, \mathbb{W}, \mathcal{B})$ is

1. **linear** if \mathbb{W} is a vector space and $\mathcal{B} \subseteq \mathbb{W}^{\mathbb{Z}_+}$ is a subspace.
2. **time-invariant** if $\sigma\mathcal{B} \subseteq \mathcal{B}$ where $\sigma w_t = w_{t+1}$.
3. **complete** if \mathcal{B} is closed $\leftrightarrow \mathbb{W}$ is finite dimensional.



A behavioral view of LTI systems

Informally, the behavior \mathcal{B} is the solution set of the linear difference equation describing the system, *i.e.*

$$\mathcal{B} \equiv \{w \in \mathbb{W}^{\mathbb{Z}_+} \mid R(\sigma)w = o\},$$

where $R \in \mathbb{R}^{g \times q}[\sigma]$ is a *polynomial matrix* (if $\mathbb{W} = \mathbb{R}^q$). This is known as the *kernal representation* of the system.

Input-Output LTI systems

If the polynomial matrix $R \in \mathbb{R}^{q \times q}[\sigma]$ is rank-deficient, the signal w can be partitioned as $w = \text{col}(u, y) \in \mathcal{B}^u \times \mathcal{B}^y$, where u is the input and y the output. Then

$$\begin{bmatrix} -Q(\sigma) & P(\sigma) \end{bmatrix} \begin{bmatrix} u \\ y \end{bmatrix} = 0 \quad \leftrightarrow \quad a_0 y_t + \cdots + a_n y_{t+n} = b_0 u_t + \cdots + b_n u_{t+n}.$$

The kernel representation of the I/O system is an *AutoRegressive with Moving Average* (ARMA) model.

Input-Output LTI systems

$$\mathcal{H}_\tau \begin{pmatrix} u \\ y \end{pmatrix} = \begin{bmatrix} \begin{bmatrix} u_1 \\ y_1 \end{bmatrix} & \begin{bmatrix} u_2 \\ y_2 \end{bmatrix} & \begin{bmatrix} u_3 \\ y_3 \end{bmatrix} & \cdots & \begin{bmatrix} u_{T-\tau+1} \\ y_{T-\tau+1} \end{bmatrix} \\ \begin{bmatrix} u_2 \\ y_2 \end{bmatrix} & \begin{bmatrix} u_3 \\ y_3 \end{bmatrix} & \begin{bmatrix} u_4 \\ y_4 \end{bmatrix} & \cdots & \vdots \\ \begin{bmatrix} u_3 \\ y_3 \end{bmatrix} & \begin{bmatrix} u_4 \\ y_4 \end{bmatrix} & \begin{bmatrix} u_5 \\ y_5 \end{bmatrix} & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \begin{bmatrix} u_\tau \\ y_\tau \end{bmatrix} & \cdots & \cdots & \cdots & \begin{bmatrix} u_T \\ y_T \end{bmatrix} \end{bmatrix}$$

$[b_0 \ b_1 \ \dots \ b_\tau \ a_0 \ a_1 \ \dots \ a_\tau]$ spans the left nullspace of this **Hankel** matrix.

Fundamental Lemma

Lemma (Willems et al., '05)

Let $T, \tau \in \mathbb{Z}_+$. Consider

- a controllable LTI system $\Sigma = (\mathbb{Z}_+, \mathbb{R}^{p+q}, \mathcal{B})$,
- a T -sample long trajectory $\text{col}(u_d, y_d) \in \mathcal{B}_T$ where u is persistently exciting of order $\tau + n$ (prediction span + # states).

Then

$$\text{colspan} \left(\mathcal{H}_\tau \begin{pmatrix} [u] \\ [y] \end{pmatrix} \right) = \mathcal{B}_\tau.$$

$$\begin{aligned} \mathbf{x}_{t+1} &= \mathbf{Ax}_t + \mathbf{Bu}_t \\ \mathbf{y}_t &= \mathbf{Cx}_t + \mathbf{Du}_t \end{aligned}$$

\iff

$$\text{colspan} \left(\begin{bmatrix} \begin{bmatrix} u_1 \\ y_1 \end{bmatrix} & \begin{bmatrix} u_2 \\ y_2 \end{bmatrix} & \begin{bmatrix} u_3 \\ y_3 \end{bmatrix} & \dots \\ \begin{bmatrix} u_2 \\ y_2 \end{bmatrix} & \begin{bmatrix} u_3 \\ y_3 \end{bmatrix} & \begin{bmatrix} u_4 \\ y_4 \end{bmatrix} & \dots \\ \begin{bmatrix} u_3 \\ y_3 \end{bmatrix} & \begin{bmatrix} u_4 \\ y_4 \end{bmatrix} & \begin{bmatrix} u_5 \\ y_5 \end{bmatrix} & \dots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix} \right)$$

Parametric state-space
model

Non-parametric model from raw data

Data-driven simulation

Problem – Predict future output $y \in \mathbb{R}^{q \cdot \tau}$ based on

- Training data $\text{col}(u^d, y^d) \in \mathcal{B}_{T_{\text{data}}} \rightarrow$ Form Hankel matrix.
- Initial trajectory $\text{col}(u_{\text{ini}}, y_{\text{ini}}) \in \mathbb{R}^{(p+q) \cdot T_{\text{ini}}} \rightarrow$ Estimate initial condition x_{ini} .
- Input signal $u \in \mathbb{R}^{p \cdot \tau} \rightarrow$ Predict forward.

Solution – Given $(u_1, u_2, \dots, u_\tau)$ and $\text{col}(u_{\text{ini}}, y_{\text{ini}})$, compute \mathbf{g} and $(y_1, y_2, \dots, y_\tau)$ from

$$\begin{bmatrix} \mathbf{U}_p \\ \mathbf{Y}_p \\ \mathbf{U}_f \\ \mathbf{Y}_f \end{bmatrix} \mathbf{g} = \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \\ y \end{bmatrix}.$$

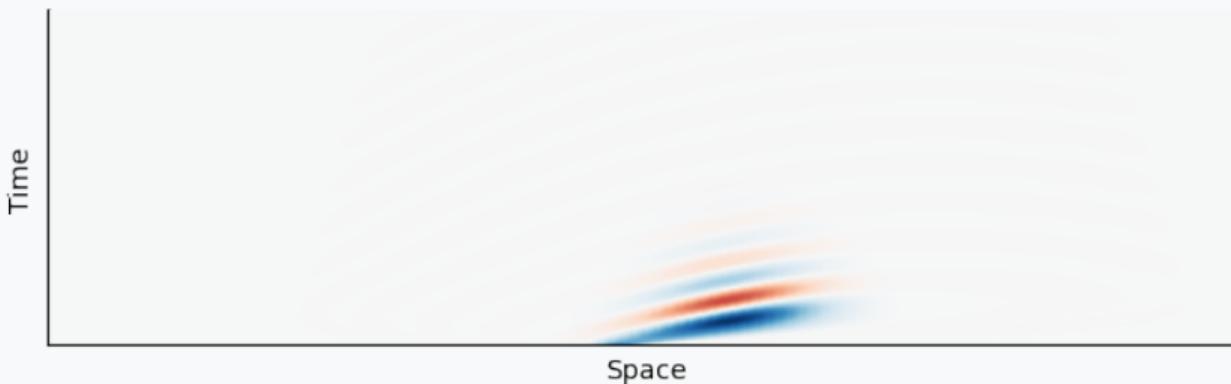
If $T_{\text{ini}} \geqslant \text{lag of the system}$ then y is unique.

Step 1 – The vector \mathbf{g} is given by

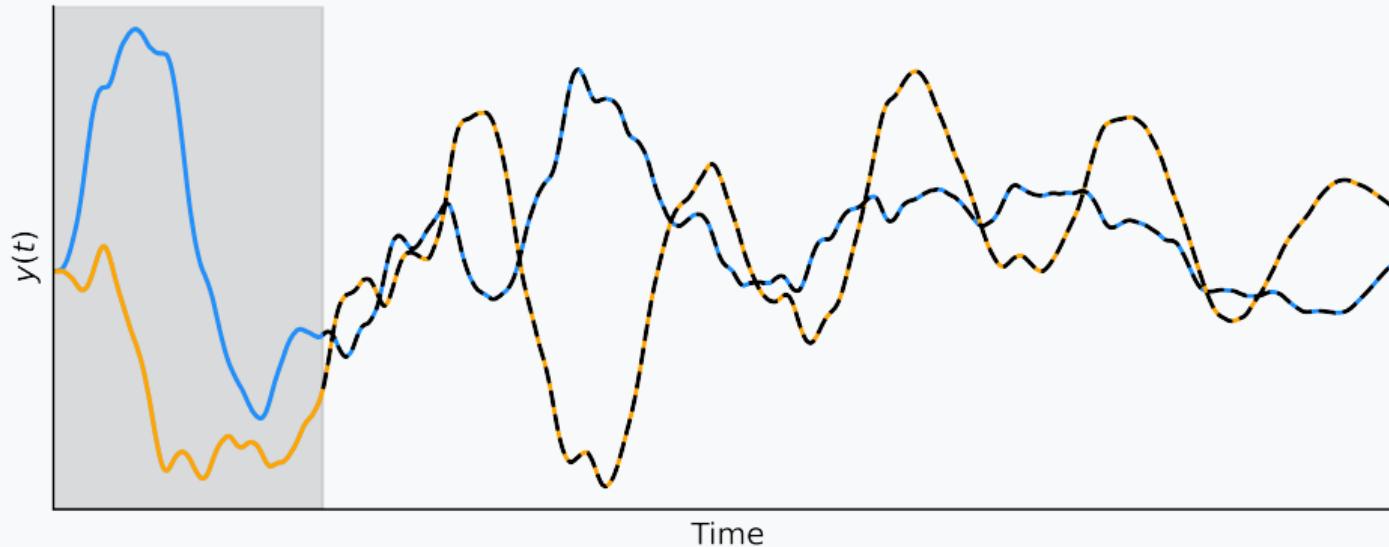
$$\mathbf{g} = \begin{bmatrix} \mathbf{U}_p \\ \mathbf{Y}_p \\ \mathbf{U}_f \end{bmatrix}^\dagger \begin{bmatrix} u_{\text{ini}} \\ y_{\text{ini}} \\ u \end{bmatrix}.$$

Step 2 - The future output \mathbf{y} is then given by

$$\mathbf{y} = \mathbf{Y}_f \mathbf{g}.$$



$$\frac{\partial u}{\partial t} = \left(-\nu \frac{\partial}{\partial x} + \gamma \frac{\partial^2}{\partial x^2} + \mu(x) \right) u + f$$



Data-Enabled Predictive Control (DeePC)

$$\begin{aligned} & \underset{\mathbf{g}, \mathbf{y}, \mathbf{u}}{\text{minimize}} && \frac{1}{2} \sum_{t=0}^{\tau} \|\mathbf{y}_t\|_{\mathbf{Q}}^2 + \|\mathbf{u}_t\|_{\mathbf{R}}^2 \\ & \text{subject to} && \begin{bmatrix} \mathbf{U}_p \\ \mathbf{Y}_p \\ \mathbf{U}_p \\ \mathbf{U}_f \end{bmatrix} \mathbf{g} = \begin{bmatrix} \mathbf{u}_{\text{ini}} \\ \mathbf{y}_{\text{ini}} \\ \mathbf{u} \\ \mathbf{y} \end{bmatrix}, \\ & && \mathbf{y}_t \in \mathcal{Y}_t, \\ & && \mathbf{u}_t \in \mathcal{U}_t. \end{aligned}$$

- Non-parametric model for *prediction* and *estimation*.
 - Very flexible framework w/ excellent performances.
 - If \mathcal{Y}_t and \mathcal{U}_t are *convex sets*, the whole optimization problem is convex.
 - Equivalent to MPC if Σ is a deterministic LTI system.

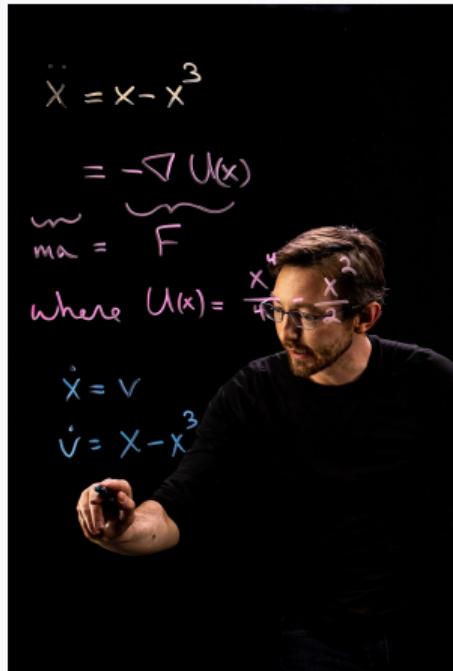
- Parameterization of the IO behavior is irrelevant as long as it captures $\mathcal{B} = \mathcal{B}^u \times \mathcal{B}^y$ correctly.
- Even when the maths are well understood, some things fundamentally stay out of reach, no matter how fancy of a data-driven algorithm you use.
- A proper choice of sensors and actuators is thus crucial to maximize the amount of information one can extract from data.



Sparse Identification of Nonlinear Dynamics

A practical approach

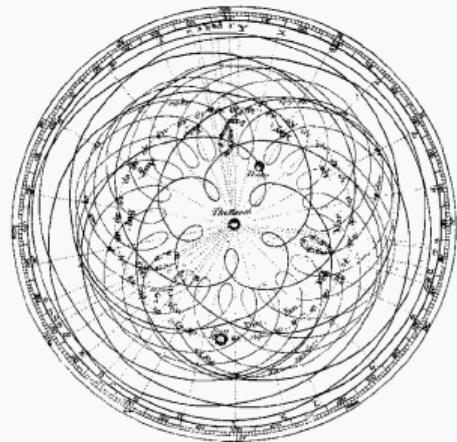
What is SINDy?



Steven Brunton (UW, USA)

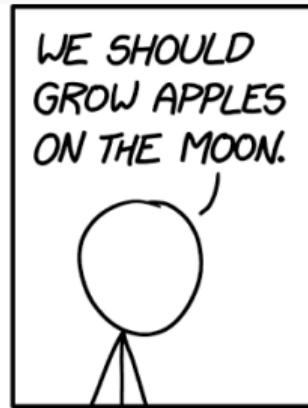
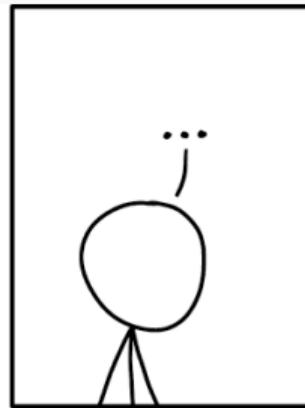
- Stands for *Sparse Identification of Nonlinear Dynamics*.
- First paper published in 2015 in PNAS.
- Framework for identifying equations from data leveraging *sparse regression* algorithms.
- Developed into a complete ecosystem since the seminal work of Steve.

Observation – Using suitable coordinates, many systems in the physical sciences are described by a set of **sparse** equations.



Observation – Using suitable coordinates, many systems in the physical sciences are described by a set of **sparse** equations.





- Vanilla SINDy
- Constrained SINDy
- Weak SINDy
- Ensemble SINDy
- SINDy for control
- SINDy-MPC
- SINDy-PI
- MANDy
- Langevin Regression
- Bayesian SINDy
- CINDy
- ...

[README](#) [License](#)



PySINDy

[CI failing](#) [docs failing](#) [pypi package 1.7.5](#) [codecov 95%](#) [JOSS 10.21105/joss.02104](#) [JOSS 10.21105/joss.03994](#)

[DOI 10.5281/zenodo.7808834](#)

PySINDy is a sparse regression package with several implementations for the Sparse Identification of Nonlinear Dynamical systems (SINDy) method introduced in Brunton et al. (2016a), including the unified optimization approach of Champion et al. (2019), SINDy with control from Brunton et al. (2016b), Trapping SINDy from Kaptanoglu et al. (2021), SINDy-PI from Kaheman et al. (2020), PDE-FIND from Rudy et al. (2017), and so on. A comprehensive literature review is given in de Silva et al. (2020) and Kaptanoglu, de Silva et al. (2021).

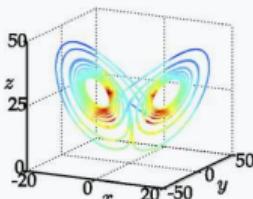
SINDy for Ordinary Diff. Eq.

I. True Lorenz System

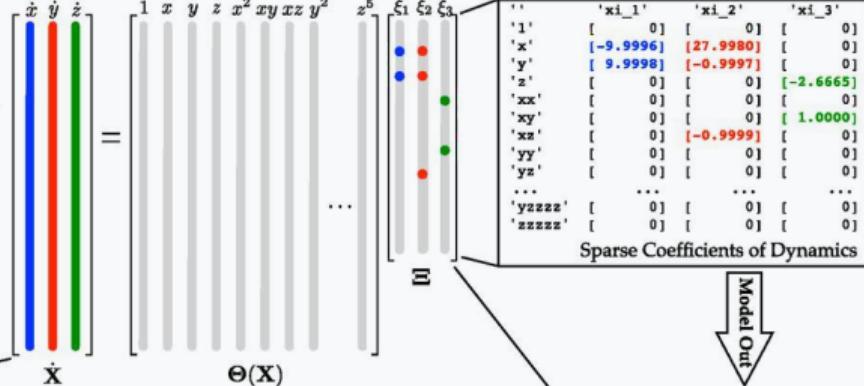
$$\dot{x} = \sigma(y - x)$$

$$\dot{y} = x(\rho - z) - y$$

$$\dot{z} = xy - \beta z.$$



Data In

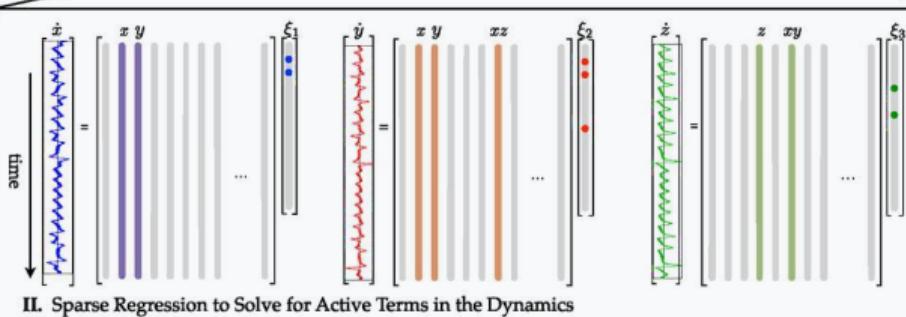
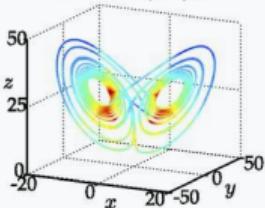


III. Identified System

$$\dot{x} = \Theta(x^T)\xi_1$$

$$\dot{y} = \Theta(x^T)\xi_2$$

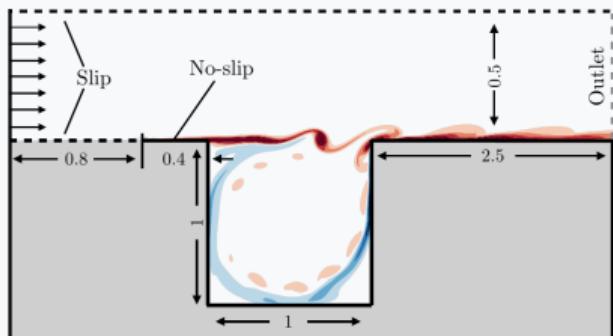
$$\dot{z} = \Theta(x^T)\xi_3$$



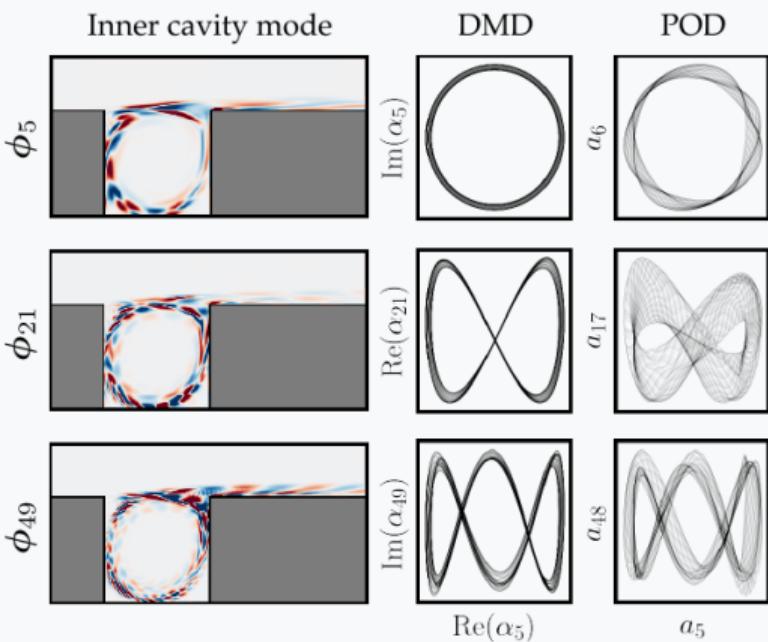
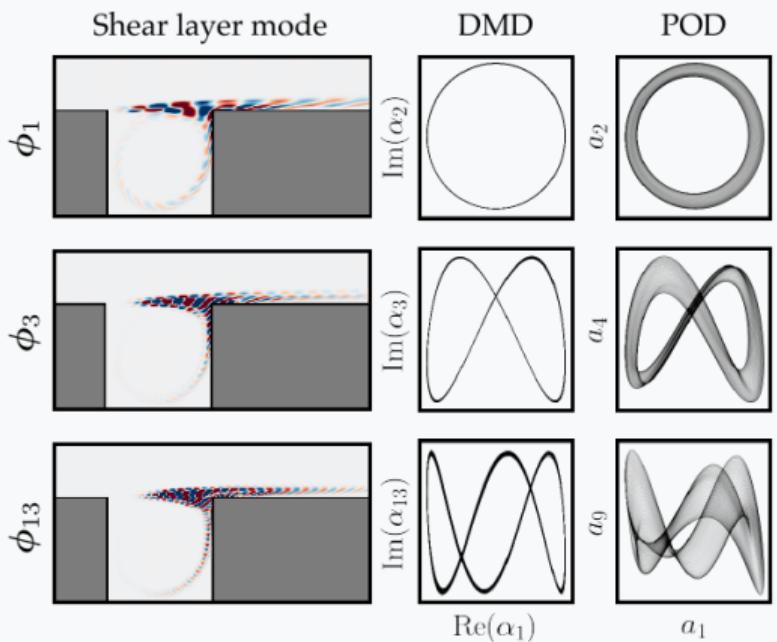
$$\begin{aligned} & \underset{\alpha}{\text{minimize}} \quad \|\alpha\|_0 \\ & \text{subject to } \int_0^T (\dot{\mathbf{x}} - f(\mathbf{x})) dt = \mathbf{o} \\ & \quad f(\mathbf{x}) - \sum_{i=1}^n \vartheta_i(\mathbf{x}) \alpha_i = \mathbf{o}, \\ & \quad h(\mathbf{x}, \alpha) = \mathbf{o} \\ & \quad g(\mathbf{x}, \alpha) \leq \mathbf{o}. \end{aligned}$$

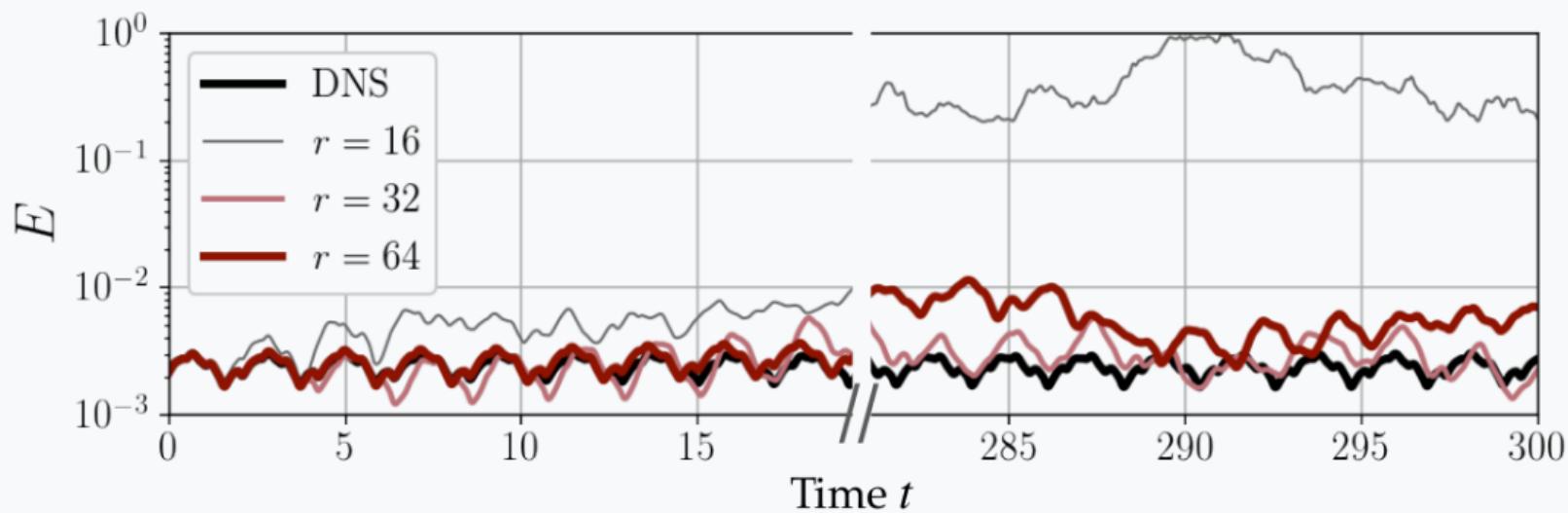
$$\begin{aligned} & \underset{\alpha}{\text{minimize}} \quad \|\alpha\|_1 \\ & \text{subject to } \|\dot{\mathbf{X}} - \Theta(\mathbf{X})\alpha\|_{\text{F}}^2 \leq \varepsilon \\ & h(\mathbf{x}, \alpha) = 0 \\ & g(\mathbf{x}, \alpha) \asymp 0. \end{aligned}$$

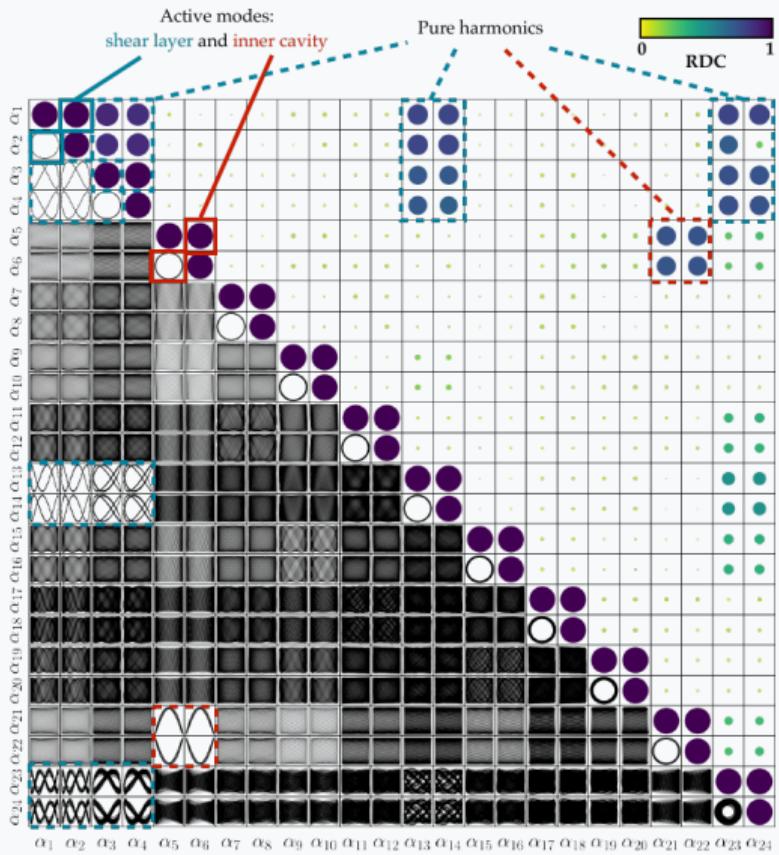
Identifying normal forms



- Classical example in flow control.
- At $Re = 7500$, the dynamics are *quasiperiodic*.
- 64 POD modes required to capture 99% of the fluctuations.







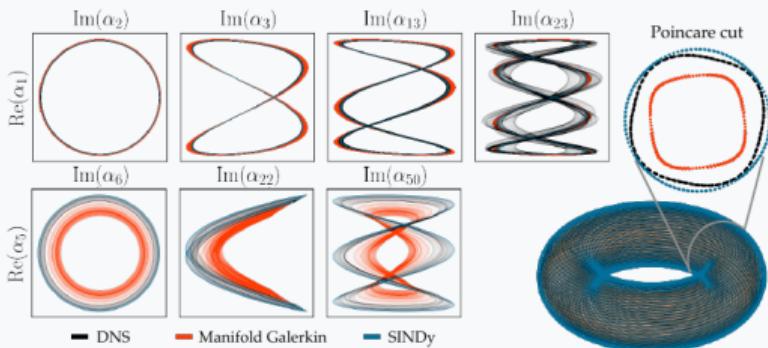
- Many modes are artifacts coming from representing a low-dimensional manifold in a large Euclidean space.
- Need to find a way to break the *Kolmogorov n -width*.

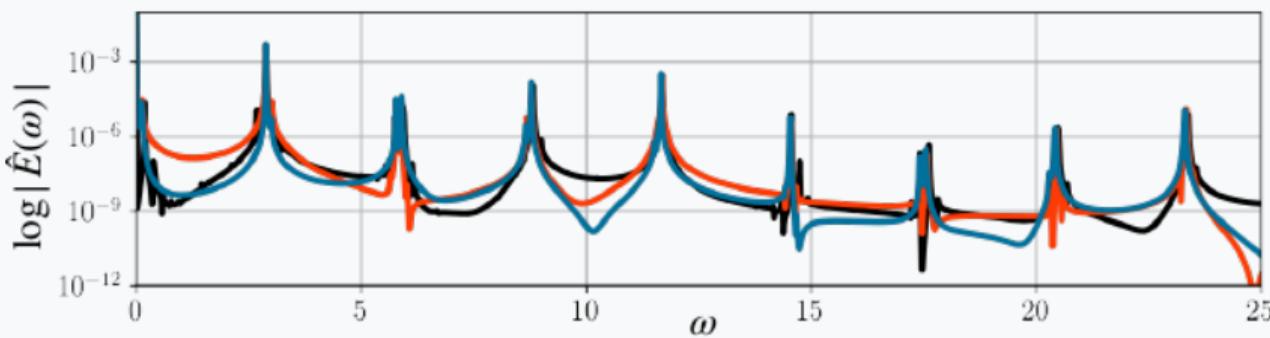
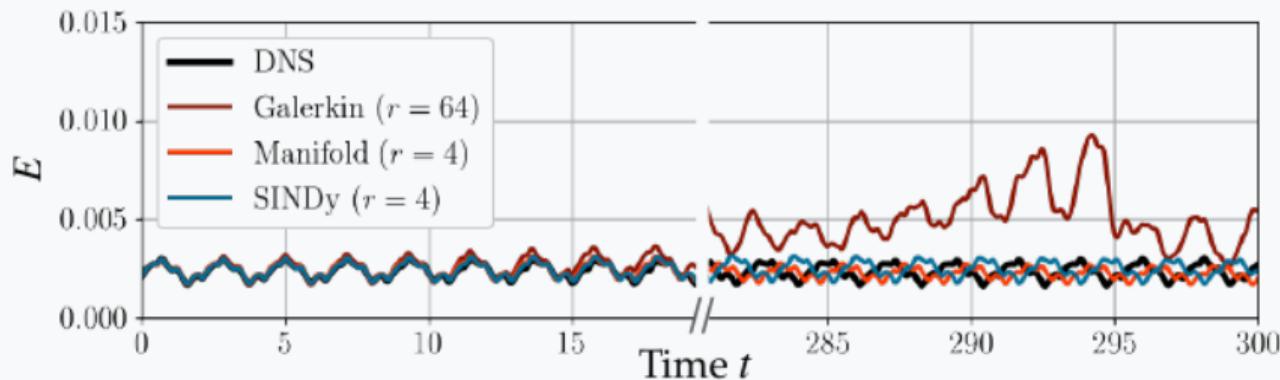
- Using only the four active degrees of freedom, SINDy identifies

$$\dot{x} = \lambda_1 x - \mu_1 |x|^2 x$$

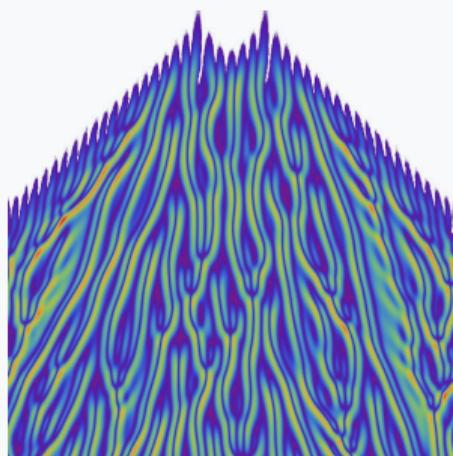
$$\dot{y} = \lambda_2 y - \mu_2 |y|^2 y.$$

- ROM consistent with the known bifurcation diagram of the problem.





SINDy for Partial Diff. Eq.



- Extending SINDy to PDE is straightforward.
- Massively over-determined constrained least-squares problem.

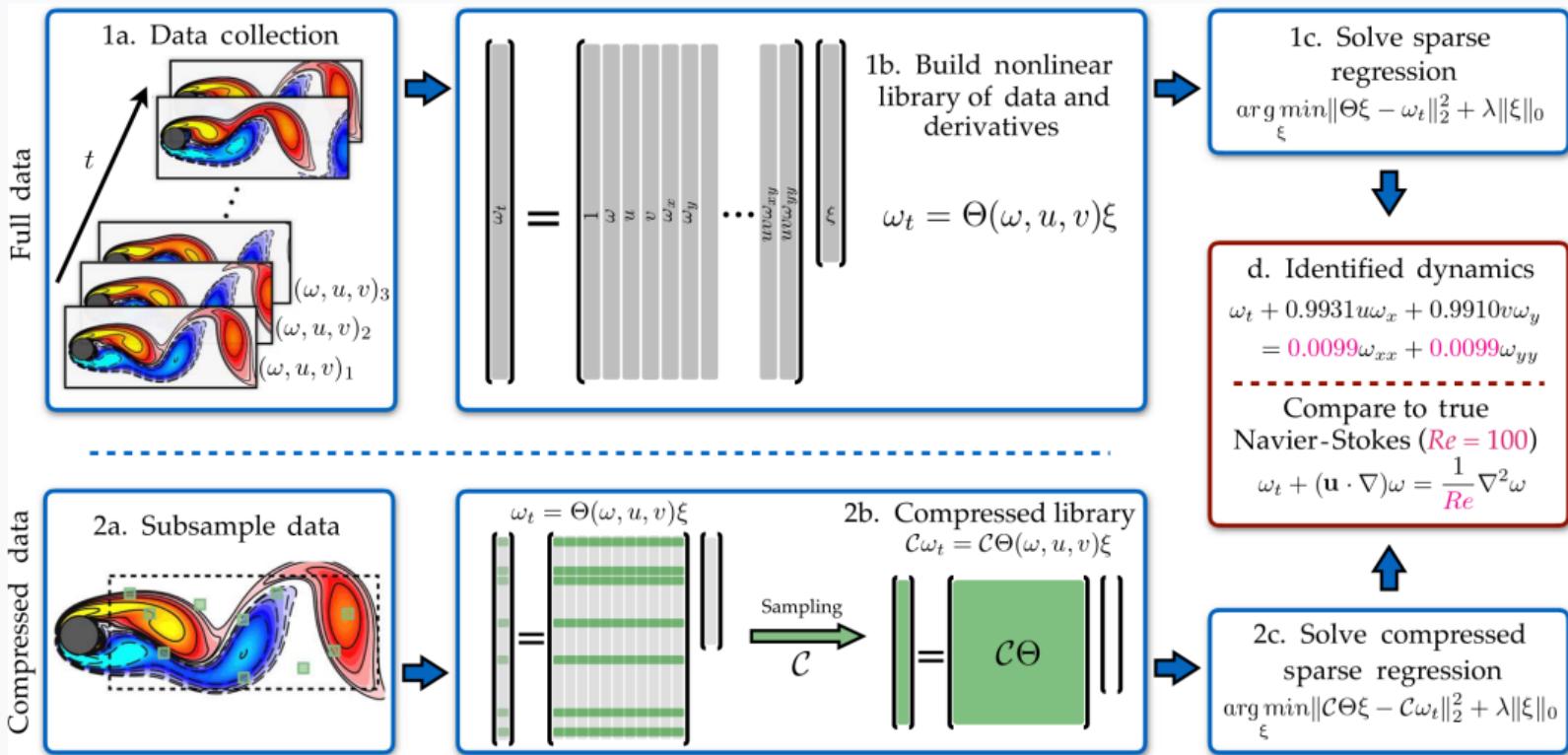
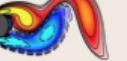
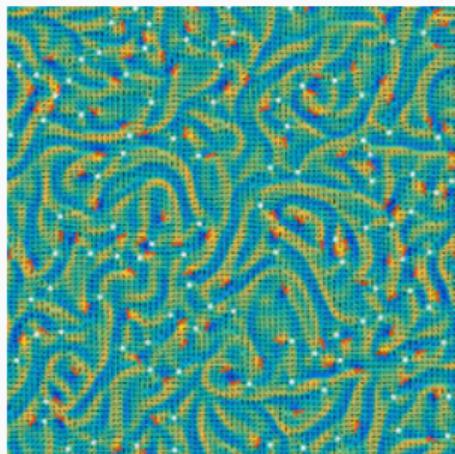


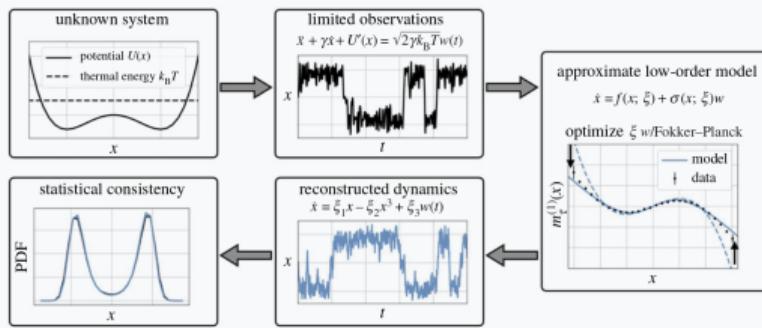
Table 1. Summary of regression results for a wide range of canonical models of mathematical physics. In each example, the correct model structure is identified using PDE-FIND. The spatial and temporal sampling of the numerical simulation data used for the regression is given along with the error produced in the parameters of the model for both no noise and 1% noise. In the reaction-diffusion system, 0.5% noise is used. For Navier-Stokes and reaction-diffusion, the percent of data used in subsampling is also given. NLS, nonlinear Schrödinger; KS, Kuramoto-Sivashinsky.

PDE	Form	Error (no noise, noise)	Discretization
 KdV	$u_t + 6uu_x + u_{xxx} = 0$	$1 \pm 0.2\%, 7 \pm 5\%$	$x \in [-30, 30], n = 512, t \in [0, 20], m = 201$
 Burgers	$u_t + uu_x - \epsilon u_{xx} = 0$	$0.15 \pm 0.06\%, 0.8 \pm 0.6\%$	$x \in [-8, 8], n = 256, t \in [0, 10], m = 101$
 Schrödinger	$iu_t + \frac{1}{2}u_{xx} - \frac{x^2}{2}u = 0$	$0.25 \pm 0.01\%, 10 \pm 7\%$	$x \in [-7.5, 7.5], n = 512, t \in [0, 10], m = 401$
 NLS	$iu_t + \frac{1}{2}u_{xx} + u ^2u = 0$	$0.05 \pm 0.01\%, 3 \pm 1\%$	$x \in [-5, 5], n = 512, t \in [0, \pi], m = 501$
 KS	$u_t + uu_x + u_{xx} + u_{xxxx} = 0$	$1.3 \pm 1.3\%, 52 \pm 1.4\%$	$x \in [0, 100], n = 1024, t \in [0, 100], m = 251$
 Reaction Diffusion	$u_t = 0.1\nabla^2 u + \lambda(A)u - \omega(A)v$ $v_t = 0.1\nabla^2 v + \omega(A)u + \lambda(A)v$ $A^2 = u^2 + v^2, \omega = -\beta A^2, \lambda = 1 - A^2$	$0.02 \pm 0.01\%, 3.8 \pm 2.4\%$	$x, y \in [-10, 10], n = 256, t \in [0, 10], m = 201$ subsample 1.14%
 Navier-Stokes	$\omega_t + (\mathbf{u} \cdot \nabla)\omega = \frac{1}{Re}\nabla^2\omega$	$1 \pm 0.2\%, 7 \pm 6\%$	$x \in [0, 9], n_x = 449, y \in [0, 4], n_y = 199,$ $t \in [0, 30], m = 151, \text{subsample } 2.22\%$

- Physical assumptions of *smoothness*, *locality* and *symmetry* can be used to design an admissible dictionary.
- Recent extension of incorporate assumptions of *smoothness*, *locality* and *symmetry* to design *a priori* a physically admissible dictionary.



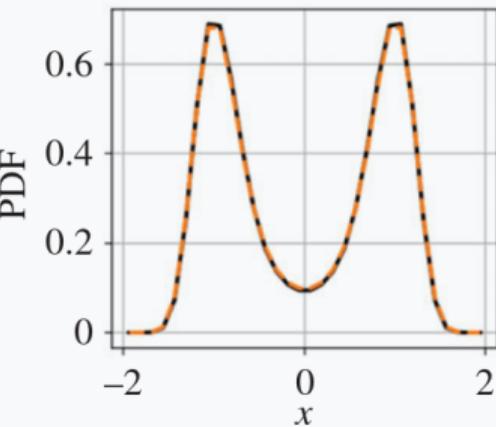
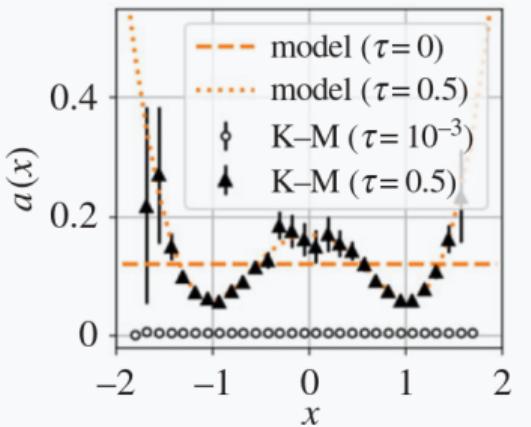
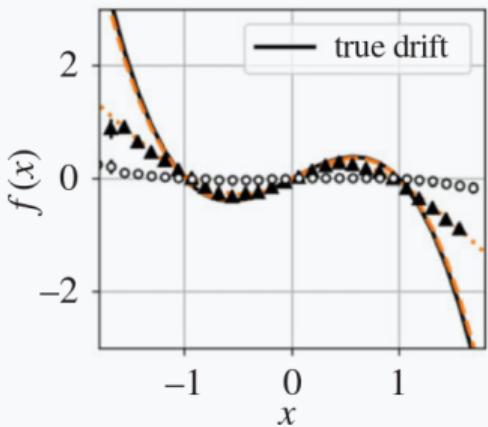
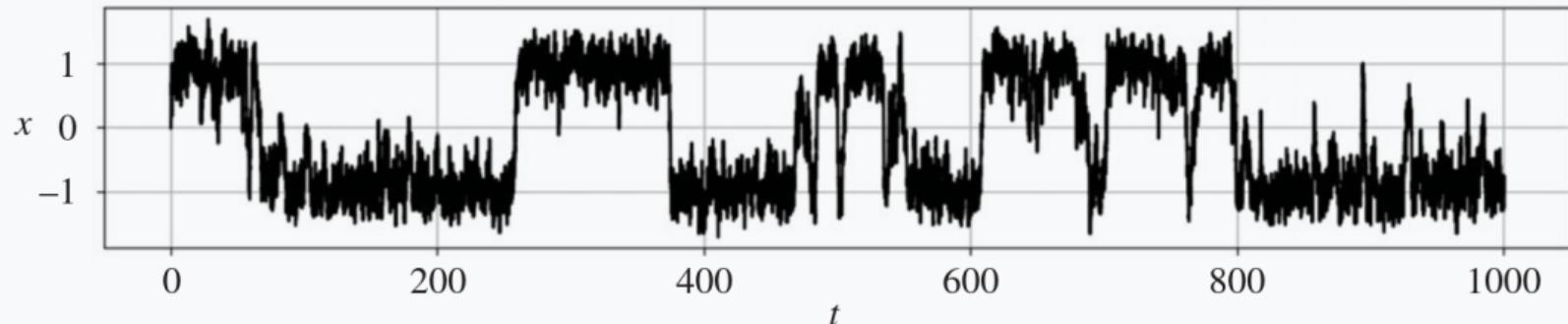
SINDy for Stochastic Diff. Eq.

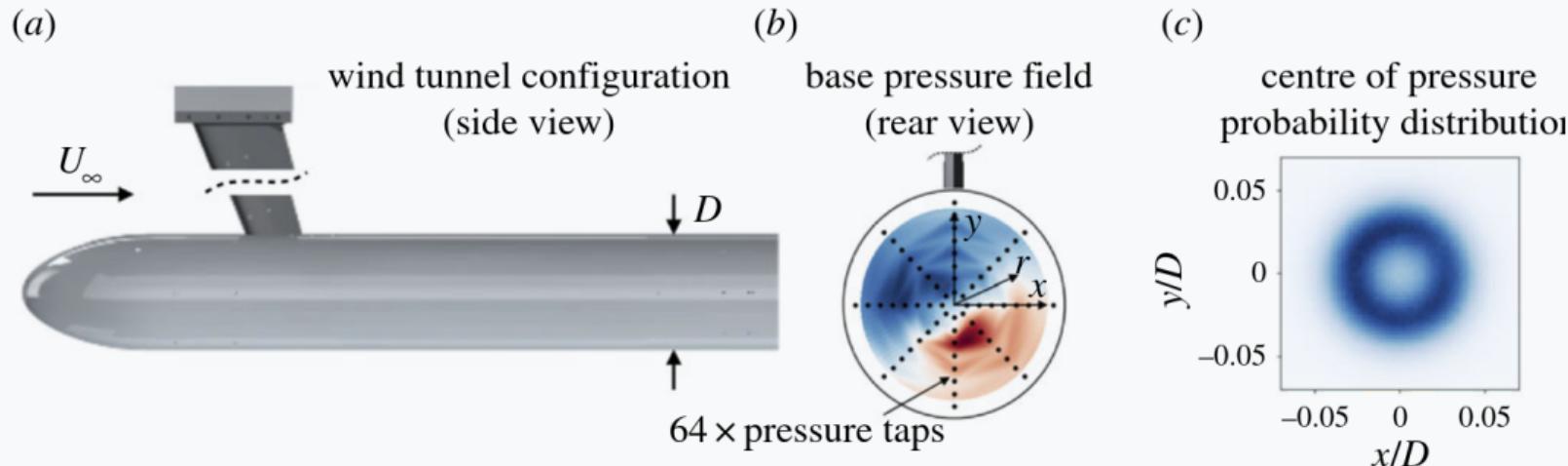


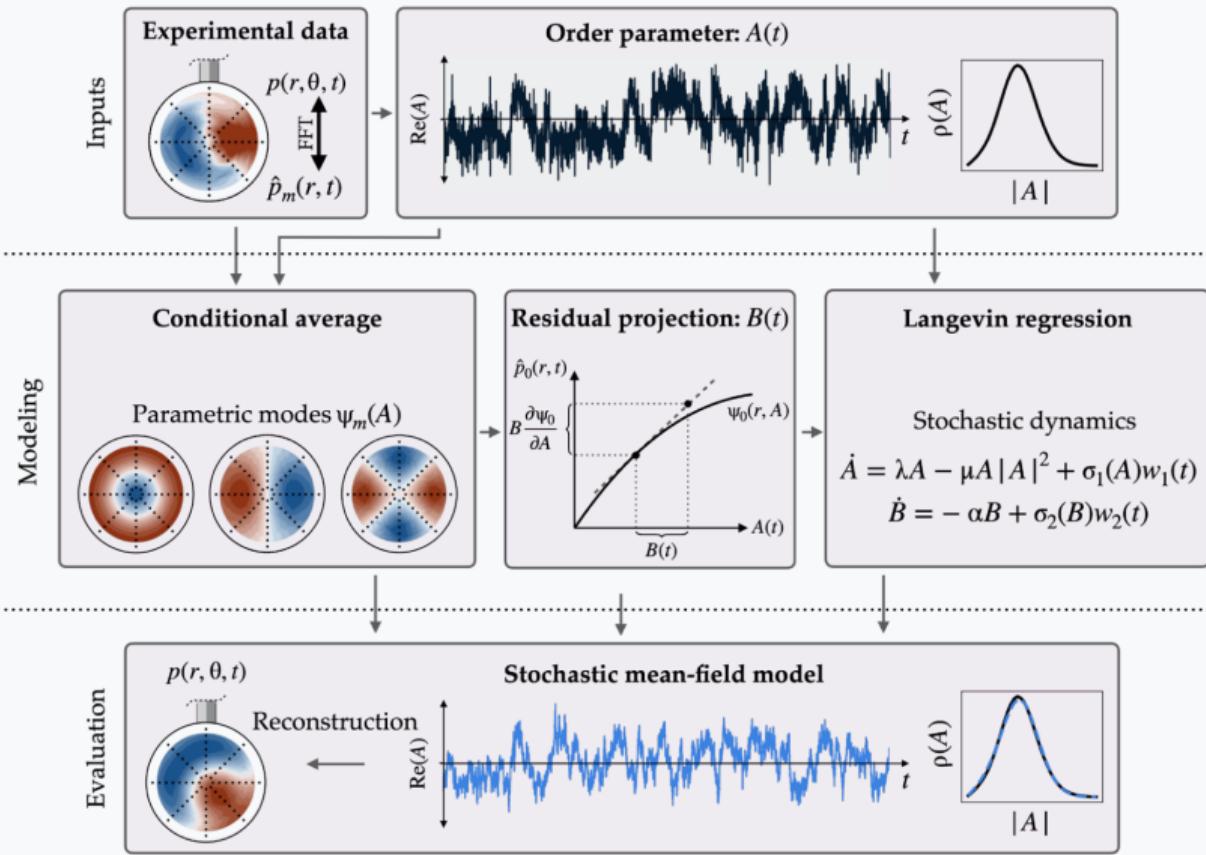
- Many systems in physics can be described by *Langevin equations*

$$\dot{x} = f(x) + \sigma(x)\eta.$$

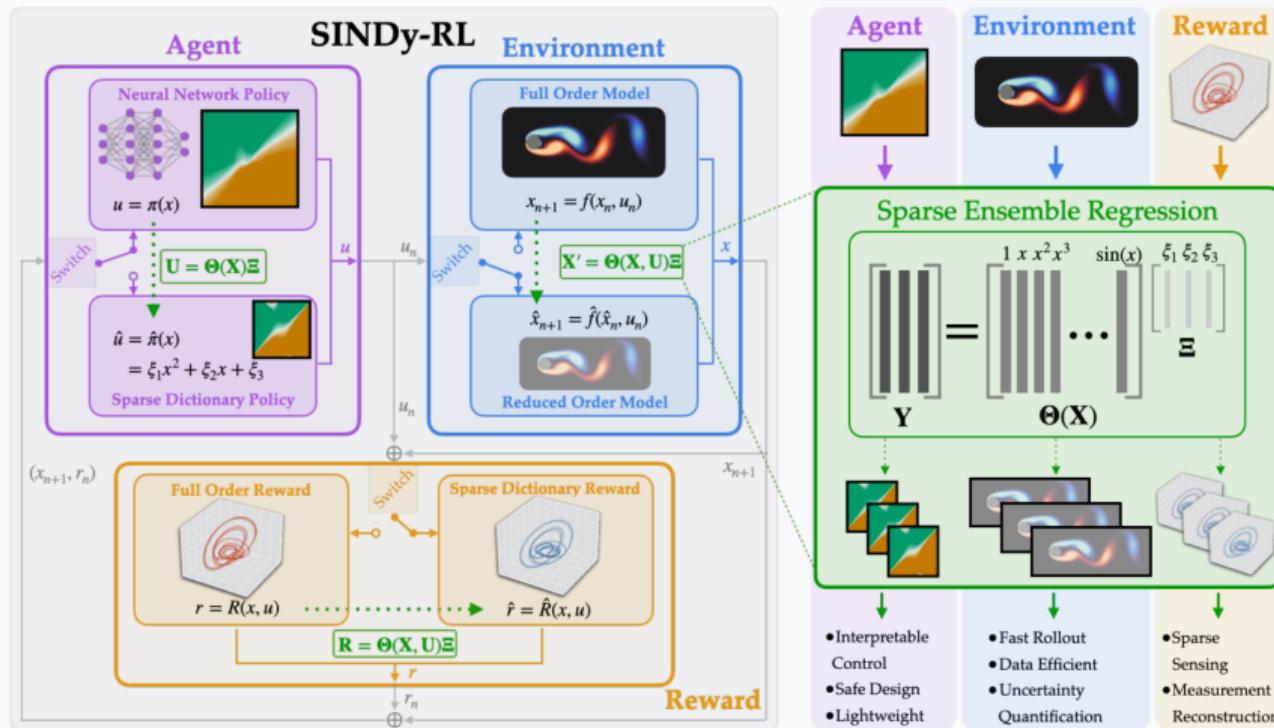
- Identifying the drift and diffusion terms are slightly more involved than vanilla SINDy.

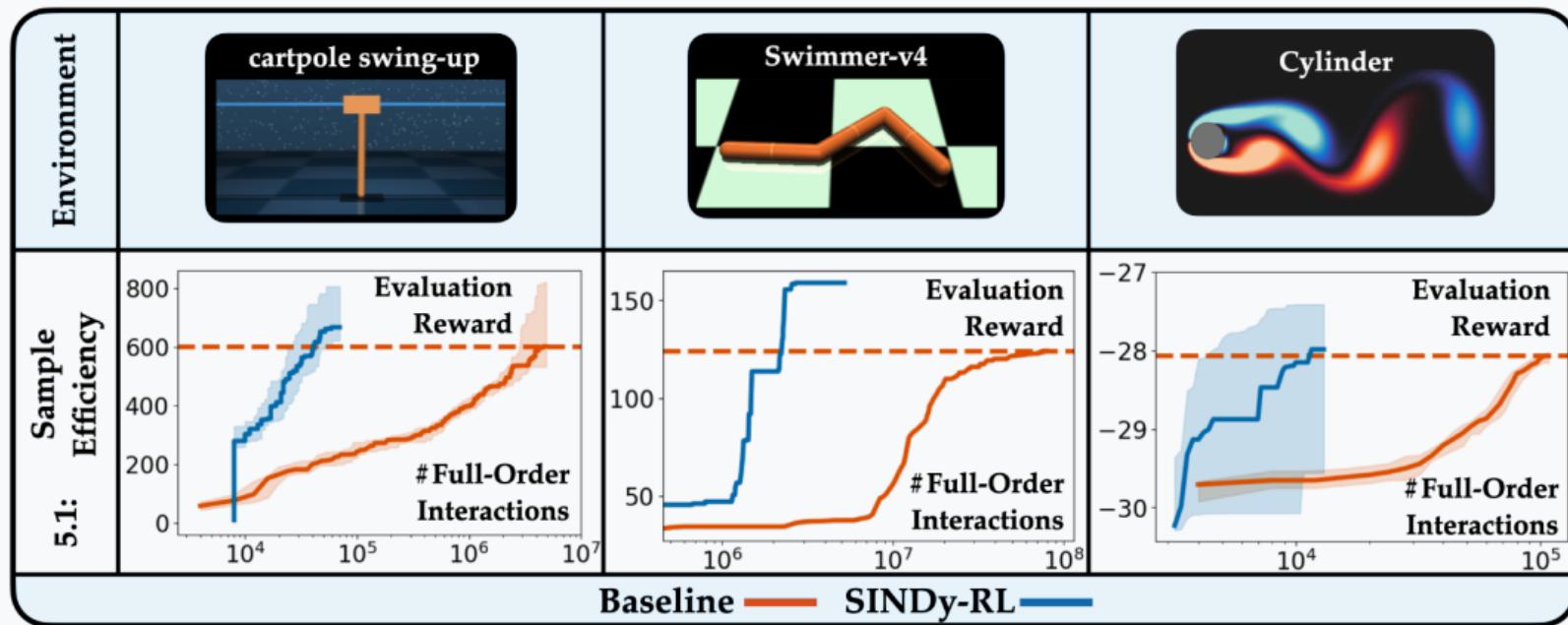






SINDy for Reinf. Learning





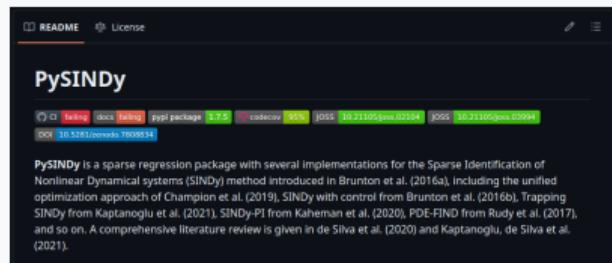
Conclusion

Conclusion

- Since 2015, **SINDy** has evolved into a mature ecosystem.
 - Ordinary diff. eq., partial diff. eq., control systems, etc.
- Despite its versatility, **SINDy** is not a silver bullet.
 - Requires quite a bit of domain expertise.



PySINDy



- Open-source Python package with a simple and scikit-learn compatible API.
- We're always on the look for new contributors!
 - More computationally efficient algorithms.
 - New/better variants of **SINDy**.

This presentation wouldn't have been possible without many collaborators, including (but not limited to):

Steven Brunton, Bing Brunton, Nathan Kutz, Jared Callaham, Kathleen Champion, Brian da Silva, Alan Kaptanoglu, Kadierdan Kaheman, Urban Fasel, Sam Rudy, Zachary Nicolaou, Georgios Rigas, Nicholas Zolman and many others.

Thank you for your attention!

Any questions?



loiseaujc.github.io