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

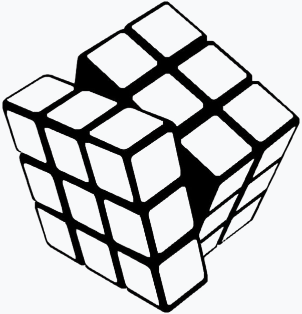
Assistant Professor of Applied Mathematics
and Fluid Dynamics

Research activities
Hydrodynamic instabilities
Transition to turbulence
Reduced-order modeling
Optimal control

Tools
High-performance computing
Numerical Linear Algebra
Convex optimization

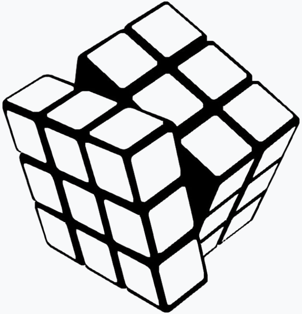
Non-exhaustive list of challenges

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

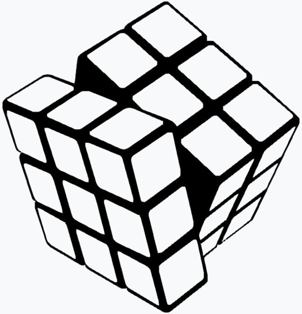
Modelling What kind of model?
Physics-based or data-driven?
How accurate does it need to be?
Can I quantify the uncertainties?
How expensive is it simulate?



Non-exhaustive list of challenges

Instrumentation

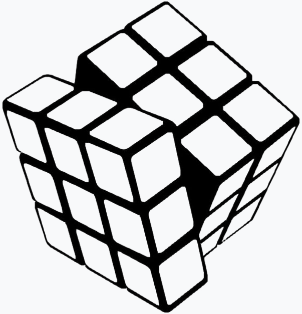
What is actually *observable*?
What kind of sensors can I use?
Where to place them and how many?
How to handle measurement noise?
What if one of my sensor fail?



Non-exhaustive list of challenges

Instrumentation

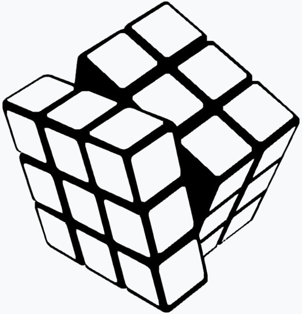
What is actually *controllable*?
What kind of actuators can I use?
Where to place them and how many?
How to handle disturbances?
What if one of my actuator fail?



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?



Linear System Identification

A brief overview

Known input sequence

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\boxed{\mathbf{u}}_t$$
$$\boxed{\mathbf{y}}_t = \mathbf{C}\mathbf{x}_t + \mathbf{D}\mathbf{u}_t.$$

Measured output sequence

Kalman decomposition

$$\begin{bmatrix} \mathbf{x}_{r\bar{o}} \\ \mathbf{x}_{ro} \\ \mathbf{x}_{\bar{r}\bar{o}} \\ \mathbf{x}_{\bar{r}o} \end{bmatrix}_{t+1} = \begin{bmatrix} \mathbf{A}_{r\bar{o}} & * & * & * \\ & \mathbf{A}_{ro} & * & * \\ & & \mathbf{A}_{\bar{r}\bar{o}} & * \\ & & & \mathbf{A}_{\bar{r}o} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{r\bar{o}} \\ \mathbf{x}_{ro} \\ \mathbf{x}_{\bar{r}\bar{o}} \\ \mathbf{x}_{\bar{r}o} \end{bmatrix}_{t} + \begin{bmatrix} \mathbf{B}_{r\bar{o}} \\ \mathbf{B}_{ro} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \mathbf{u}_t$$

$$\mathbf{y}_t = \begin{bmatrix} \mathbf{0} & \mathbf{C}_{ro} & \mathbf{0} & \mathbf{C}_{\bar{r}o} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{r\bar{o}} \\ \mathbf{x}_{ro} \\ \mathbf{x}_{\bar{r}\bar{o}} \\ \mathbf{x}_{\bar{r}o} \end{bmatrix}_{t} + \mathbf{D}\mathbf{u}_t$$

Kalman decomposition

$$\begin{bmatrix} \mathbf{x}_{r\bar{o}} \\ \mathbf{x}_{ro} \\ \mathbf{x}_{\bar{r}\bar{o}} \\ \mathbf{x}_{\bar{r}o} \end{bmatrix}_{t+1} = \begin{bmatrix} \mathbf{A}_{r\bar{o}} & * & * & * \\ & \boxed{\mathbf{A}_{ro}} & * & * \\ & & \mathbf{A}_{\bar{r}\bar{o}} & * \\ & & & \mathbf{A}_{\bar{r}o} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{r\bar{o}} \\ \mathbf{x}_{ro} \\ \mathbf{x}_{\bar{r}\bar{o}} \\ \mathbf{x}_{\bar{r}o} \end{bmatrix}_{t} + \begin{bmatrix} \mathbf{B}_{r\bar{o}} \\ \boxed{\mathbf{B}_{ro}} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \mathbf{u}_t$$

$$\mathbf{y}_t = \begin{bmatrix} \mathbf{0} & \boxed{\mathbf{C}_{ro}} & \mathbf{0} & \mathbf{C}_{\bar{r}o} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{r\bar{o}} \\ \mathbf{x}_{ro} \\ \mathbf{x}_{\bar{r}\bar{o}} \\ \mathbf{x}_{\bar{r}o} \end{bmatrix}_{t} + \boxed{\mathbf{D}}\mathbf{u}_t$$

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.

$$\begin{aligned} \mathbf{x}_{t+1} &= \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t \\ \mathbf{y}_t &= \mathbf{C}\mathbf{x}_t + \mathbf{D}\mathbf{u}_t \end{aligned} \qquad \begin{aligned} \mathbf{z}_{t+1} &= \mathbf{T}^{-1}\mathbf{A}\mathbf{T}\mathbf{z}_t + \mathbf{T}^{-1}\mathbf{B}\mathbf{u}_t \\ \mathbf{y}_t &= \mathbf{C}\mathbf{T}\mathbf{z}_t + \mathbf{D}\mathbf{u}_t \end{aligned}$$

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

Linear System Identification

Realization theory

Known input sequence

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\boxed{\mathbf{u}}_t$$
$$\boxed{\mathbf{y}}_t = \mathbf{C}\mathbf{x}_t + \mathbf{D}\mathbf{u}_t.$$

Measured output sequence

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

$$\mathbf{x}_2 = \mathbf{A}\mathbf{x}_1 + \mathbf{B}\mathbf{u}_1, \quad \mathbf{x}_3 = \mathbf{A}\mathbf{x}_2 + \mathbf{B}\mathbf{u}_2, \quad \mathbf{x}_4 = \mathbf{A}\mathbf{x}_3 + \mathbf{B}\mathbf{u}_3,$$

or alternatively

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

leading to

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

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

<u>Past data</u>

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

<u>Future data</u>

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

Future output

Past output

$$\boxed{\mathbf{Y}_f} - \mathbf{H}\,\boxed{\mathbf{U}_f} = \mathcal{O}\mathbf{A}^k\mathcal{O}^\dagger\,\boxed{\mathbf{Y}_p} + \mathcal{O}\left(\boldsymbol{\Delta} - \mathbf{A}^k\mathcal{O}^\dagger\mathbf{H}\right)\boxed{\mathbf{U}_p}$$

Future input

Past input

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{k-1} \end{bmatrix}$$

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

$$\Delta = \begin{bmatrix} \mathbf{A}^{k-1}\mathbf{B} & \mathbf{A}^{k-2}\mathbf{B} & \cdots & \mathbf{AB} & \mathbf{B} \end{bmatrix}$$

The problem we'd like to solve reads

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

It however is a non-convex problem.

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

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

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

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

Quadratic Program with Quadratic Equality constraints

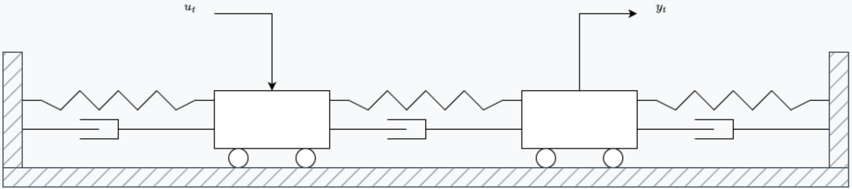
$$
\begin{aligned}
\underset{\mathcal{O},\boldsymbol{\Delta}}{\text{minimize}} \quad & \operatorname{Tr}\left(\begin{bmatrix} \mathcal{O}^\top & \boldsymbol{\Delta}^\top \end{bmatrix} \begin{bmatrix} \mathbf{0} & -\mathbf{M}\mathbf{Y}_f\mathbf{W}\mathbf{U}_p \\ -\mathbf{U}_p^\top\mathbf{W}\mathbf{Y}_f\mathbf{M}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

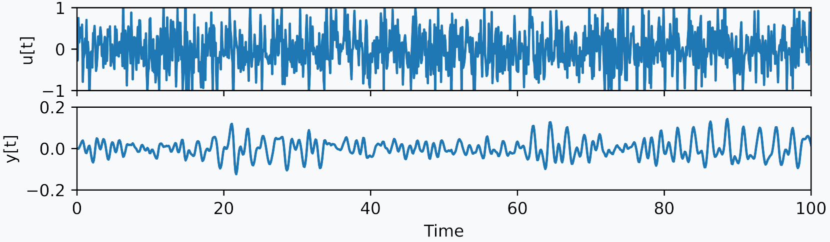
Trace minimization of a Hermitian matrix

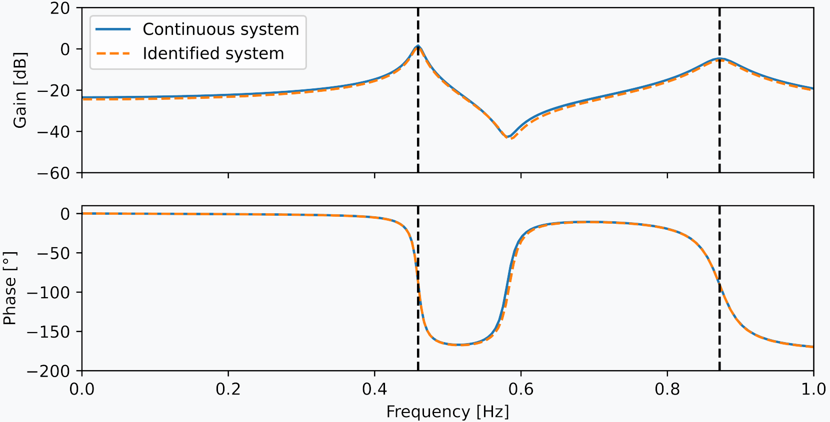
$$\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathcal{O} \\ \Delta \end{bmatrix} \Lambda = \begin{bmatrix} \mathbf{0} & -\mathbf{M}\mathbf{Y}_f\mathbf{W}\mathbf{U}_p \\ -\mathbf{U}_p^\top \mathbf{W}\mathbf{Y}_f\mathbf{M} & \mathbf{U}_p\mathbf{W}\mathbf{U}_p \end{bmatrix} \begin{bmatrix} \mathcal{O} \\ \Delta \end{bmatrix}$$

Generalized eigenvalue problem

	N4SID	MOESP	CVA
$\mathbf{M}$	$\mathbf{I}$	$\mathbf{I}$	$\left(\mathbf{Y}_f \Pi_{\mathbf{U}}^{\perp} \mathbf{Y}_f^T\right)^{-1}$
$\mathbf{W}$	?	$\Pi_{\mathbf{U}^{\perp}}$	$\Pi_{\mathbf{U}}^{\perp}$


u_t
y_t


1
0
−1
u[t]
0.2
0.0
−0.2
y[t]
0
20
40
60
80
100
Time


Continuous system
Identified system
Gain [dB]
20
0
−20
−40
−60
Phase [°]
0
−50
−100
−150
−200
0.0
0.2
0.4
0.6
0.8
1.0
Frequency [Hz]

Model Predictive Control

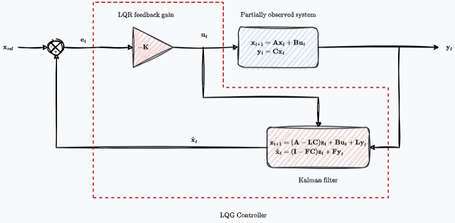
$$
\begin{aligned}
\text{minimize} \quad & \frac{1}{2}\sum_{t=0}^{\tau} \|\mathbf{y}_t\|_{\mathbf{Q}}^2 + \|\mathbf{u}_t\|_{\mathbf{R}}^2 \\
\text{subject to} \quad & \mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t \\
& \mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \mathbf{D}\mathbf{u}_t \\
& \mathbf{y}_t \in \mathcal{Y}_t \\
& \mathbf{u}_t \in \mathcal{U}_t
\end{aligned}
$$

- 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 behavorial approach

Single Input/Single Output setup



Hypothesis

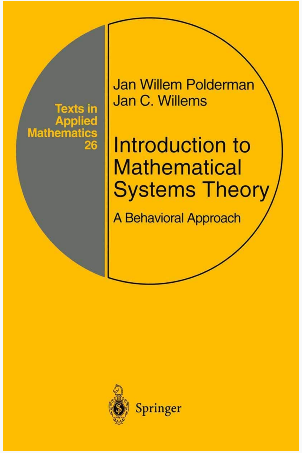
- The true system Σ is unknown.
- Its input-output dynamics can be well approximated by an LTI model.
- The rank of the model is unknown.
- Only input-output data are available.

A behavorial 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*.

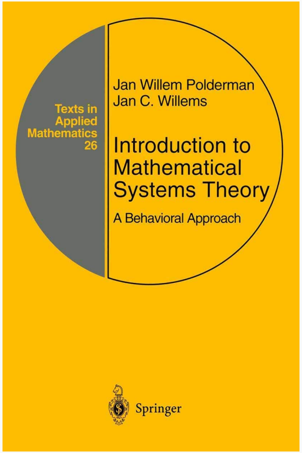


A behavorial view of LTI systems

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 behavorial view of LTI systems

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

$$\mathcal{B} \equiv \left\{ w \in \mathbb{W}^{\mathbb{Z}_+} \quad \middle| \quad R(\sigma) w = 0 \right\},$$

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 $R \in \mathbb{R}^{g \times q}[\sigma]$ is rank-deficient, the signal w can be partitioned as $w = \operatorname{col}(u, y) \in \mathcal{B}^u \times \mathcal{B}^y$, 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

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

$\begin{bmatrix} b_0 & a_0 & \cdots & b_\tau & a_\tau \end{bmatrix}$ 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* $\Sigma = \left(\mathbb{Z}_+, \mathbb{R}^{p+q}, \mathcal{B}\right)$,
- *a* T*-sample long trajectory* $\mathrm{col}(u_d, y_d) \in \mathcal{B}_T$ *where* u *is persistently exciting of order* $\tau + n$ *(prediction span + # states).*

Then

$$\mathrm{colspan}\left(\mathcal{H}_\tau\left(\begin{bmatrix} u \\ y \end{bmatrix}\right)\right) = \mathcal{B}_\tau.$$

$$\begin{aligned} \mathbf{x}_{t+1} &= \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t \\ \mathbf{y}_t &= \mathbf{C}\mathbf{x}_t + \mathbf{D}\mathbf{u}_t \end{aligned} \iff \text{colspan}\left(\begin{bmatrix} \begin{bmatrix} u_1 \\ y_1 \end{bmatrix} & \begin{bmatrix} u_2 \\ y_2 \end{bmatrix} & \begin{bmatrix} u_3 \\ y_3 \end{bmatrix} & \cdots \\ \begin{bmatrix} u_2 \\ y_2 \end{bmatrix} & \begin{bmatrix} u_3 \\ y_3 \end{bmatrix} & \begin{bmatrix} u_4 \\ y_4 \end{bmatrix} & \cdots \\ \begin{bmatrix} u_3 \\ y_3 \end{bmatrix} & \begin{bmatrix} u_4 \\ y_4 \end{bmatrix} & \begin{bmatrix} u_5 \\ y_5 \end{bmatrix} & \cdots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}\right)$$

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

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

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

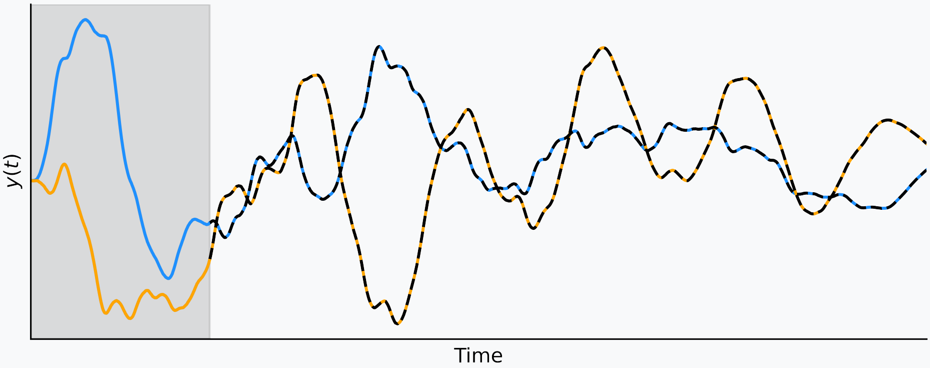
If $T_{\mathrm{ini}} \geqslant$ lag of the system then y is unique.

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 y is then given by

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


$y(t)$
Time

Data-Enabled Predictive Control (DeePC)

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

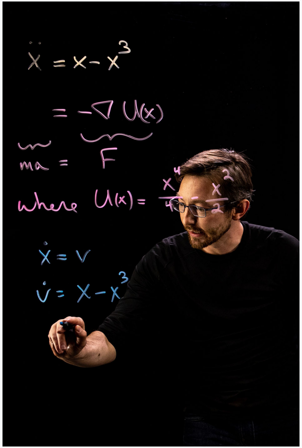
- Non-parametric model for *prediction* and *estimation*.
- Very flexible framework w/ excellent performances.
- If $\mathcal{Y}_t$ and $\mathcal{U}_t$ are *convex sets*, the whole optimization problem is convex.
- Equivalent to MPC if Σ is a deterministic LTI system.

- 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

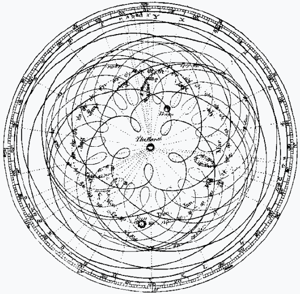
What is SINDy?



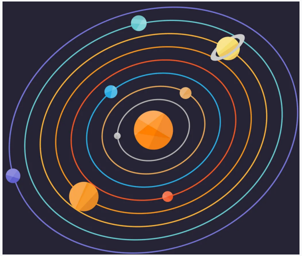
Steven Brunton (UW, USA)

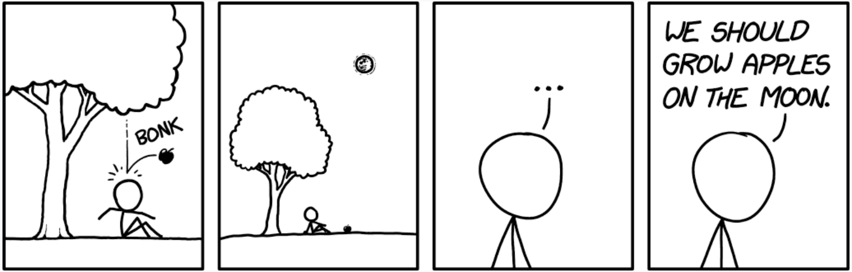
- Stands for *Sparse Identification of Nonlinear Dynamics*.
- First paper published in 2015 in PNAS.
- Framework for identifying equations from data leveraging *sparse regression* algorithms.
- Developed into a complete ecosystem since the seminal work of Steve.

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.




BONK
...
WE SHOULD GROW APPLES ON THE MOON.

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

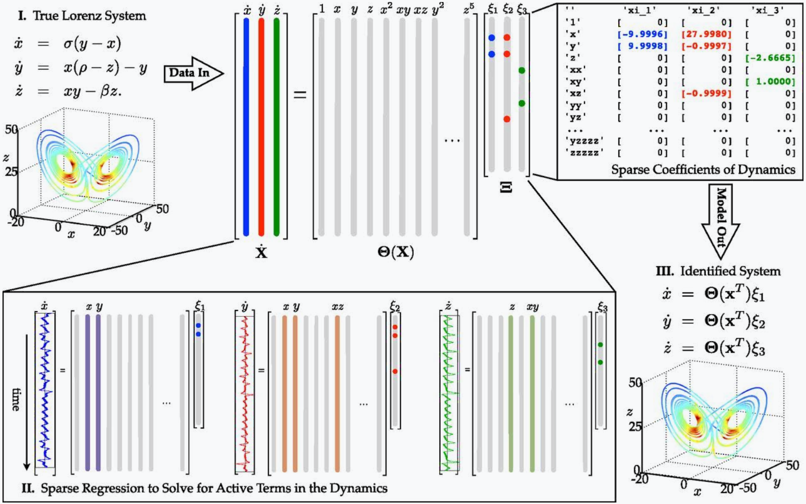
README License

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).

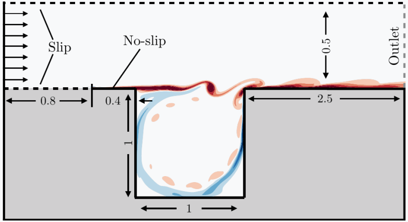
SINDy for Ordinary Diff. Eq.



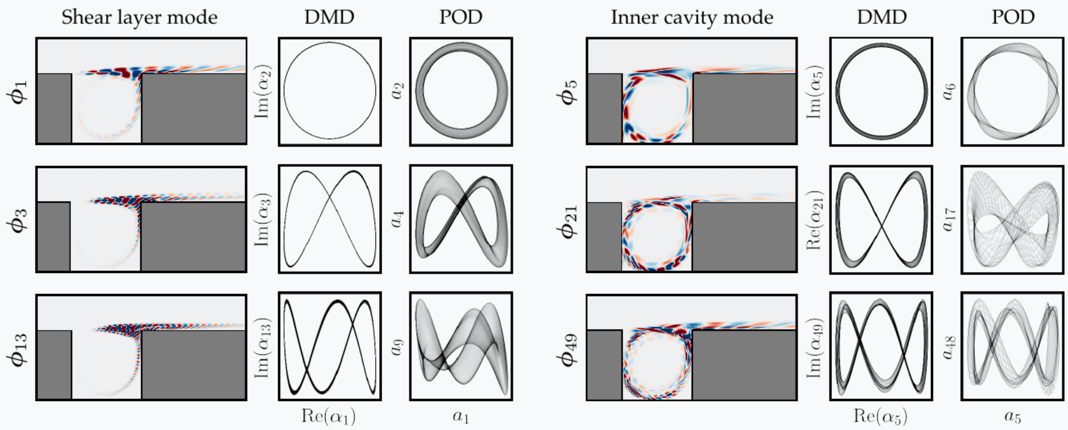
$$
\begin{aligned}
\underset{\boldsymbol{\alpha}}{\text{minimize}} \quad & \|\boldsymbol{\alpha}\|_0 \\
\text{subject to} \quad & \int_0^T (\dot{\mathbf{x}} - f(\mathbf{x}))\, dt = \mathbf{0} \\
& f(\mathbf{x}) - \sum_{i=1}^{n} \vartheta_i(\mathbf{x}) \alpha_i = \mathbf{0}, \\
& h(\mathbf{x}, \boldsymbol{\alpha}) = \mathbf{0} \\
& g(\mathbf{x}, \boldsymbol{\alpha}) \preccurlyeq \mathbf{0}.
\end{aligned}
$$

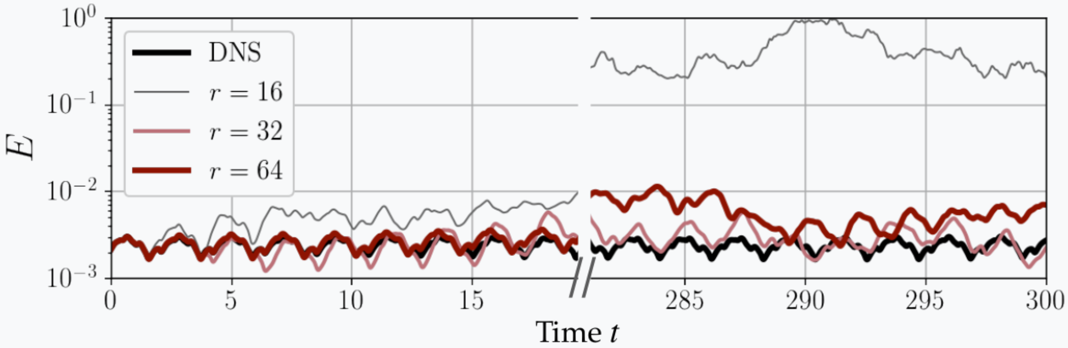
$$
\begin{aligned}
\underset{\boldsymbol{\alpha}}{\text{minimize}} \quad & \|\boldsymbol{\alpha}\|_1 \\
\text{subject to} \; & \left\|\dot{\mathbf{X}} - \boldsymbol{\Theta}(\mathbf{X})\boldsymbol{\alpha}\right\|_F^2 \leqslant \varepsilon \\
& h(\mathbf{x}, \boldsymbol{\alpha}) = 0 \\
& g(\mathbf{x}, \boldsymbol{\alpha}) \preccurlyeq 0.
\end{aligned}
$$

Identifying normal forms

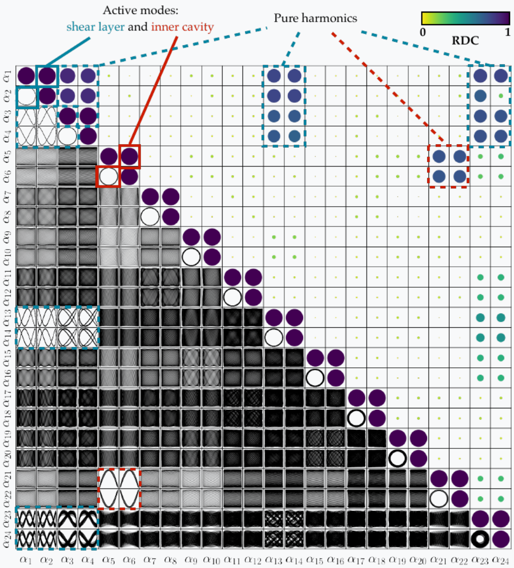


- Classical example in flow control.
- At $\mathrm{Re} = 7500$, the dynamics are *quasiperiodic*.
- 64 POD modes required to capture 99% of the fluctuations.


Shear layer mode
DMD
POD
Inner cavity mode
DMD
POD
ϕ_1
ϕ_3
ϕ_{13}
ϕ_5
ϕ_{21}
ϕ_{49}
$\mathrm{Im}(\alpha_2)$
$\mathrm{Im}(\alpha_3)$
$\mathrm{Im}(\alpha_{13})$
a_2
a_4
a_9
$\mathrm{Im}(\alpha_5)$
$\mathrm{Re}(\alpha_{21})$
$\mathrm{Im}(\alpha_{49})$
a_6
a_{17}
a_{48}
$\mathrm{Re}(\alpha_1)$
a_1
$\mathrm{Re}(\alpha_5)$
a_5



DNS
$r = 16$
$r = 32$
$r = 64$
E
10^0
10^{-1}
10^{-2}
10^{-3}
0
5
10
15
285
290
295
300
Time t

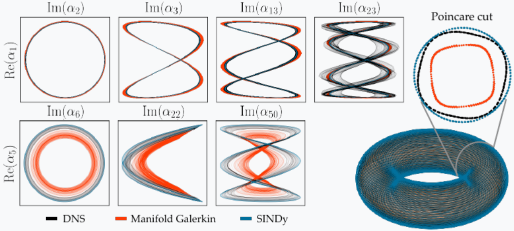


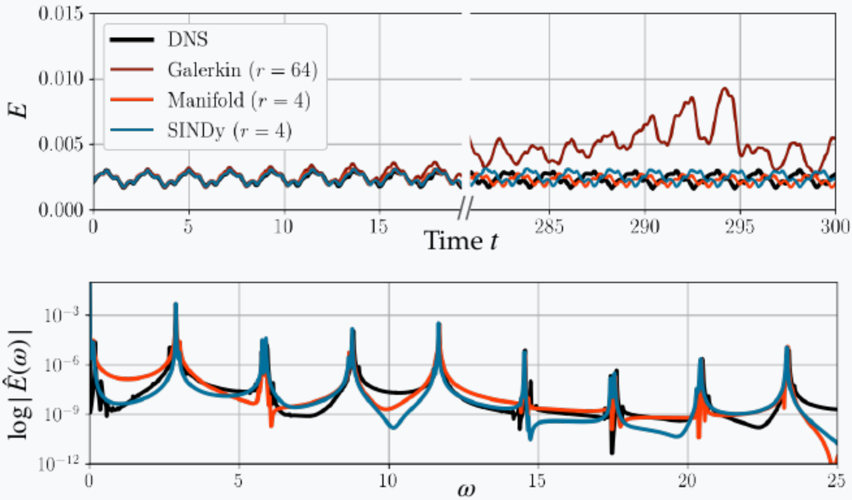
- Many modes are artifacts coming from representing a low-dimensional manifold in a large Euclidean space.
- Need to find a way to break the *Kolmogorov n-width*.

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

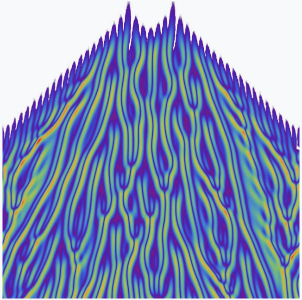
$$\dot{x} = \lambda_1 x - \mu_1 |x|^2 x$$
$$\dot{y} = \lambda_2 y - \mu_2 |y|^2 y.$$

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

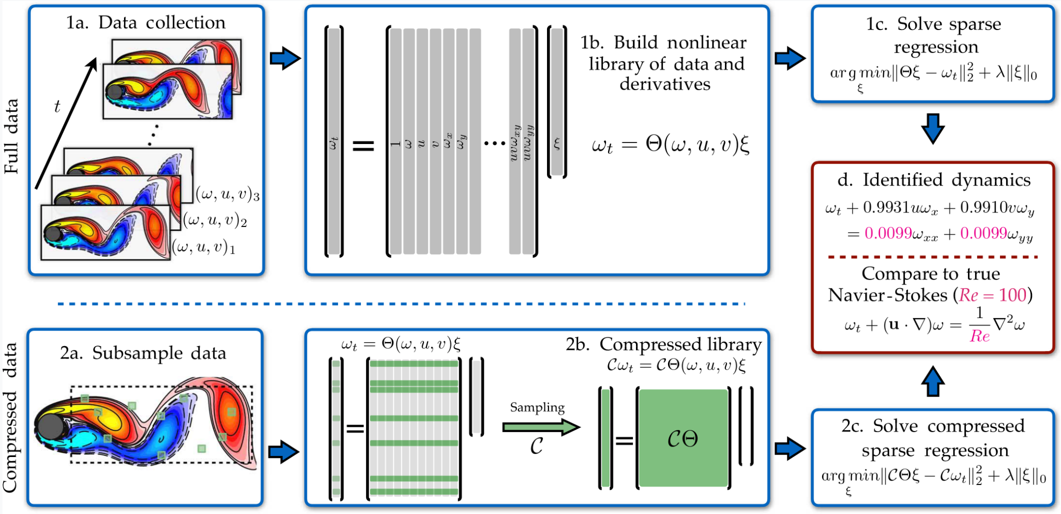



0.015
0.010
0.005
0.000
E
DNS
Galerkin $(r = 64)$
Manifold $(r = 4)$
SINDy $(r = 4)$
0
5
10
15
285
290
295
300
Time t
10^{-3}
10^{-6}
10^{-9}
10^{-12}
$\log|\hat{E}(\omega)|$
0
5
10
15
20
25
ω

SINDy for Partial Diff. Eq.

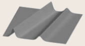
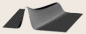
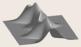
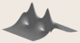
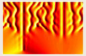
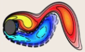


- Extending SINDy to PDE is straightforward.
- Massively over-determined constrained least-squares problem.



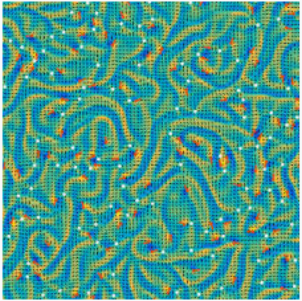
Full data
1a. Data collection
t
$(\omega, u, v)_3$
$(\omega, u, v)_2$
$(\omega, u, v)_1$
ω_t
1 ω u v ω_x ω_y $\cdots$ $uv\omega_{xy}$ $uv\omega_{yy}$
ξ
1b. Build nonlinear library of data and derivatives
$\omega_t = \Theta(\omega, u, v)\xi$
1c. Solve sparse regression
$\arg\min_\xi \|\Theta\xi - \omega_t\|_2^2 + \lambda\|\xi\|_0$
d. Identified dynamics
$\omega_t + 0.9931 u\omega_x + 0.9910 v\omega_y = 0.0099\omega_{xx} + 0.0099\omega_{yy}$
Compare to true Navier-Stokes ($Re = 100$)
$\omega_t + (\mathbf{u}\cdot\nabla)\omega = \frac{1}{Re}\nabla^2\omega$
Compressed data
2a. Subsample data
$\omega_t = \Theta(\omega, u, v)\xi$
Sampling
$\mathcal{C}$
2b. Compressed library
$\mathcal{C}\omega_t = \mathcal{C}\Theta(\omega, u, v)\xi$
$\mathcal{C}\Theta$
2c. Solve compressed sparse regression
$\arg\min_\xi \|\mathcal{C}\Theta\xi - \mathcal{C}\omega_t\|_2^2 + \lambda\|\xi\|_0$

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.

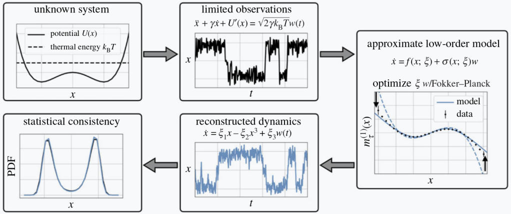
PDE	Form	Error (no noise, noise)	Discretization
 KdV	$u_t + 6uu_x + u_{xxx} = 0$	$1 \pm 0.2\%, 7 \pm 5\%$	$x \in [-30, 30], n = 512,\ t \in [0, 20], m = 201$
 Burgers	$u_t + uu_x - \epsilon u_{xx} = 0$	$0.15 \pm 0.06\%, 0.8 \pm 0.6\%$	$x \in [-8, 8], n = 256,\ t \in [0, 10], m = 101$
 Schrödinger	$iu_t + \frac{1}{2}u_{xx} - \frac{x^2}{2}u = 0$	$0.25 \pm 0.01\%, 10 \pm 7\%$	$x \in [-7.5, 7.5], n = 512,\ t \in [0, 10], m = 401$
 NLS	$iu_t + \frac{1}{2}u_{xx} + \lvert u\rvert^2 u = 0$	$0.05 \pm 0.01\%, 3 \pm 1\%$	$x \in [-5, 5], n = 512,\ t \in [0, \pi], m = 501$
 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	$\omega_t + (\mathbf{u} \cdot \nabla)\omega = \frac{1}{Re}\nabla^2\omega$	$1 \pm 0.2\%\ ,\ 7 \pm 6\%$	$x \in [0, 9], n_x = 449,\ y \in [0, 4], n_y = 199,$ $t \in [0, 30], m = 151,$ 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.



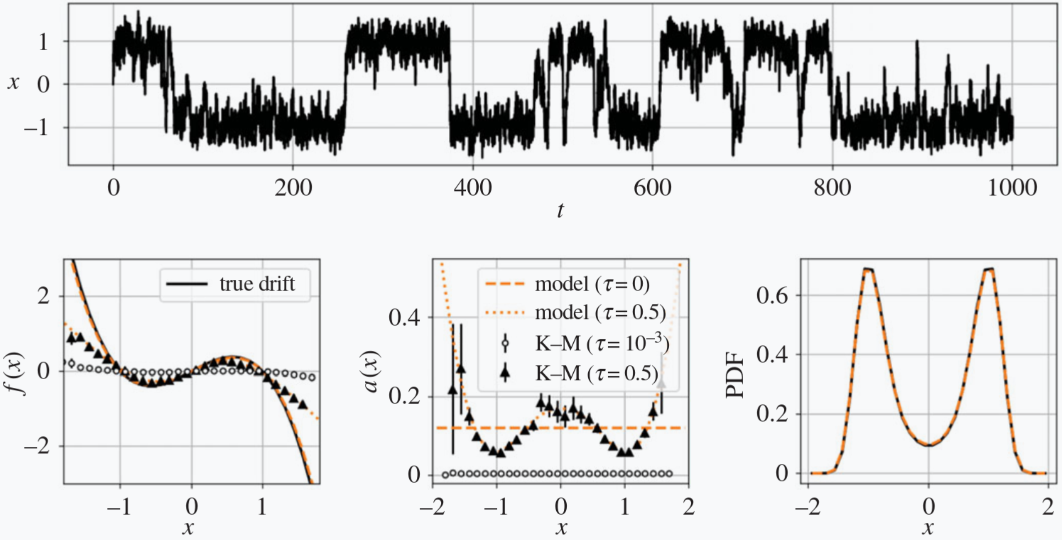
SINDy for Stochastic Diff. Eq.

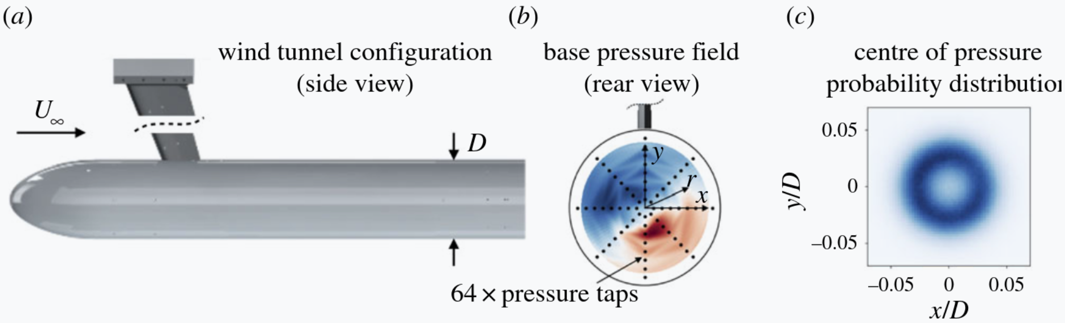


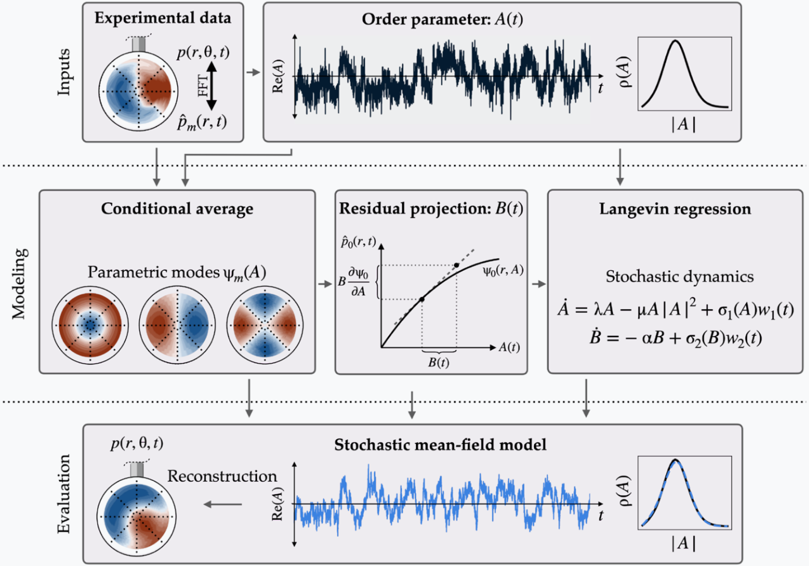
- Many systems in physics can be described by *Langevin equations*

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

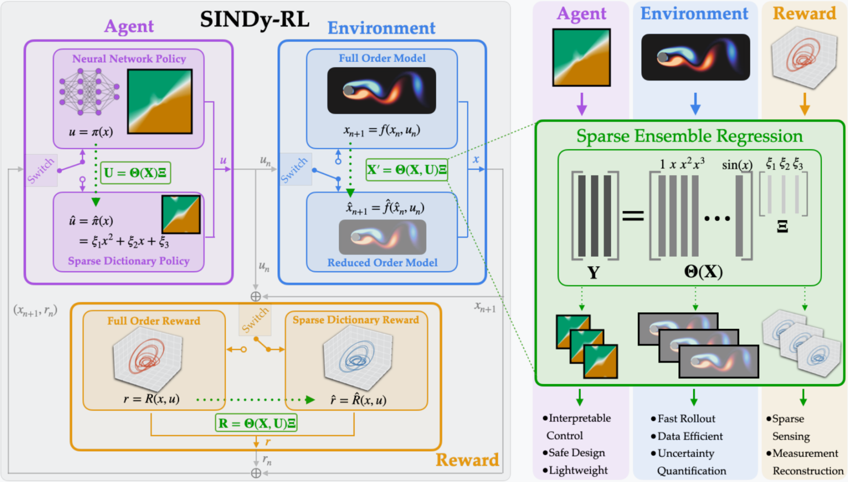
- Identifying the drift and diffusion terms are is slightly more involved than vanilla SINDy.

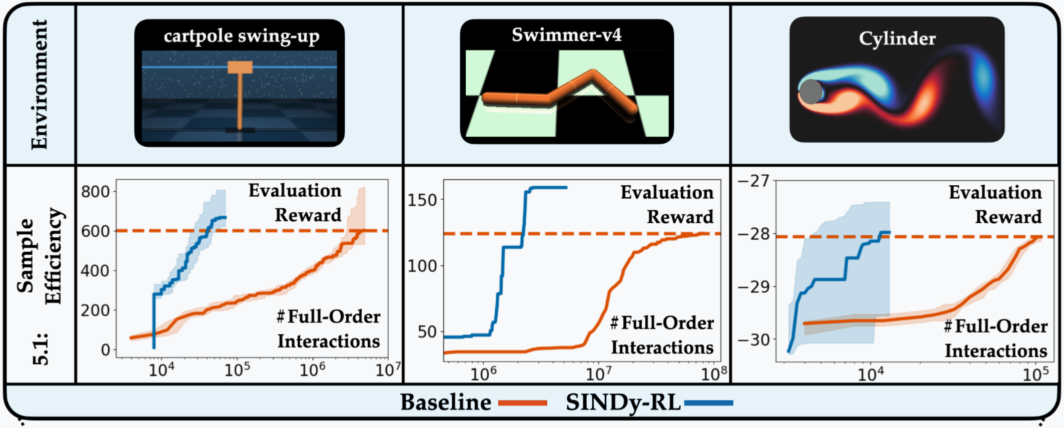

x
1
0
−1
0
200
400
600
800
1000
t
true drift
f(x)
2
0
−2
−1
0
1
x
model (τ = 0)
model (τ = 0.5)
K–M (τ = 10^−3)
K–M (τ = 0.5)
a(x)
0.4
0.2
0
−2
−1
0
1
2
x
PDF
0.6
0.4
0.2
0
−2
0
2
x


(a)
wind tunnel configuration
(side view)
U_∞
D
(b)
base pressure field
(rear view)
y
r
x
64 × pressure taps
(c)
centre of pressure
probability distributio
0.05
0
–0.05
y/D
–0.05
0
0.05
x/D


Inputs
Experimental data
$p(r,\theta,t)$
FFT
$\hat{p}_m(r,t)$
Order parameter: $A(t)$
Re(A)
t
$\rho(A)$
$|A|$
Modeling
Conditional average
Parametric modes $\psi_m(A)$
Residual projection: $B(t)$
$\hat{p}_0(r,t)$
$B\frac{\partial\psi_0}{\partial A}$
$\psi_0(r,A)$
$A(t)$
$B(t)$
Langevin regression
Stochastic dynamics
$\dot{A} = \lambda A - \mu A|A|^2 + \sigma_1(A)w_1(t)$
$\dot{B} = -\alpha B + \sigma_2(B)w_2(t)$
Evaluation
$p(r,\theta,t)$
Reconstruction
Stochastic mean-field model
Re(A)
t
$\rho(A)$
$|A|$

SINDy for Reinf. Learning




Environment
cartpole swing-up
Swimmer-v4
Cylinder
5.1: Sample Efficiency
Evaluation Reward
Full-Order Interactions
800
600
400
200
0
10^4
10^5
10^6
10^7
150
100
50
10^6
10^7
10^8
−27
−28
−29
−30
10^4
10^5
Baseline
SINDy-RL

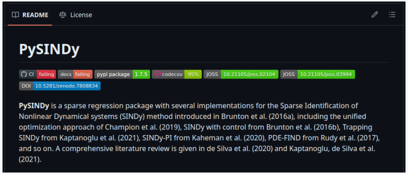
Conclusion

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


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).

- 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