SINDy – Sparse Identification of Nonlinear Dynamics

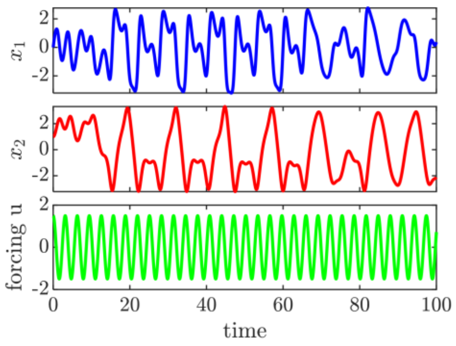
Filton workshop 2024

Urban Fasel

Imperial College London

SINDy

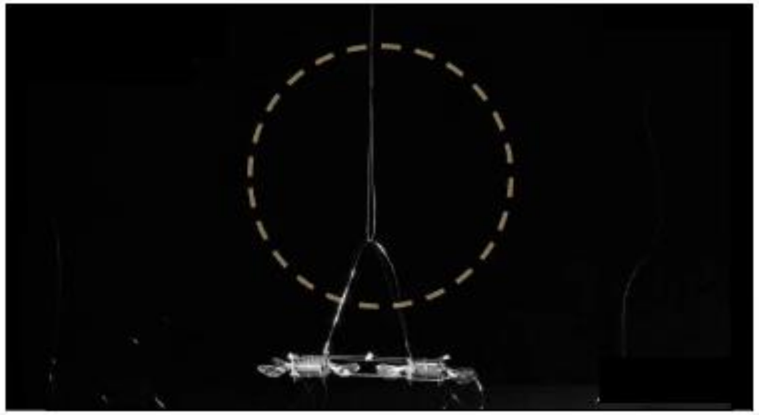
Data → *Dynamics (assumptions)* → *Model structure & coefficients*

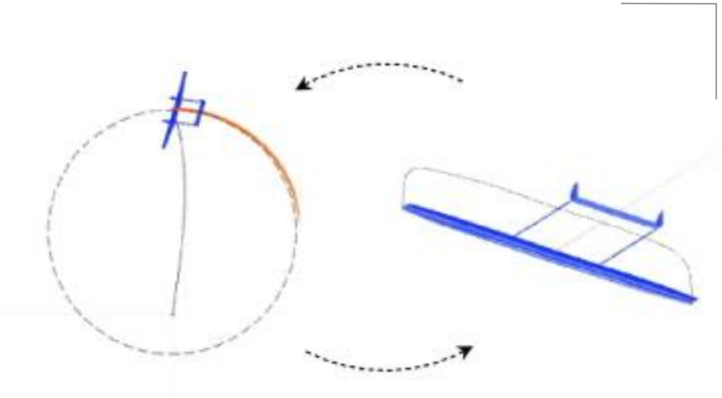


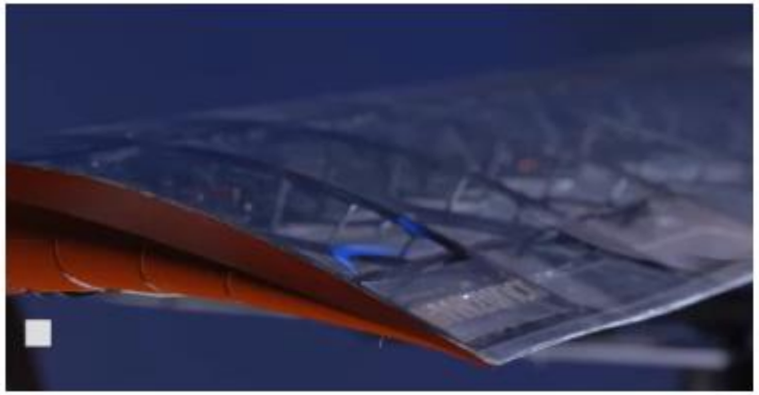
$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x})$$

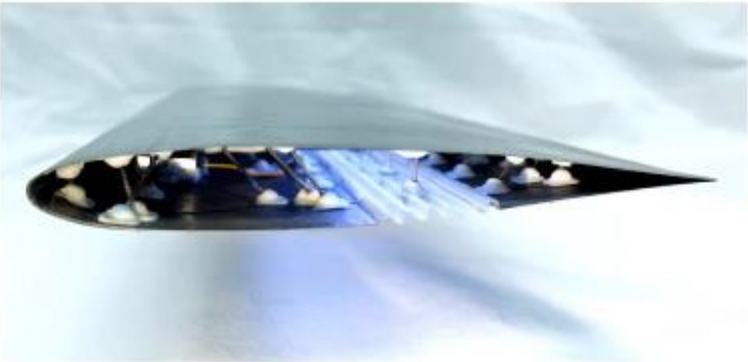
$$\begin{aligned} \dot{x}_1 &= -\delta x_1 - \alpha x_2 - \beta x_2^3 + u \\ \dot{x}_2 &= x_1 \\ u &= \gamma \cos(\omega t) \end{aligned}$$

My work









(Adaptive) flight systems

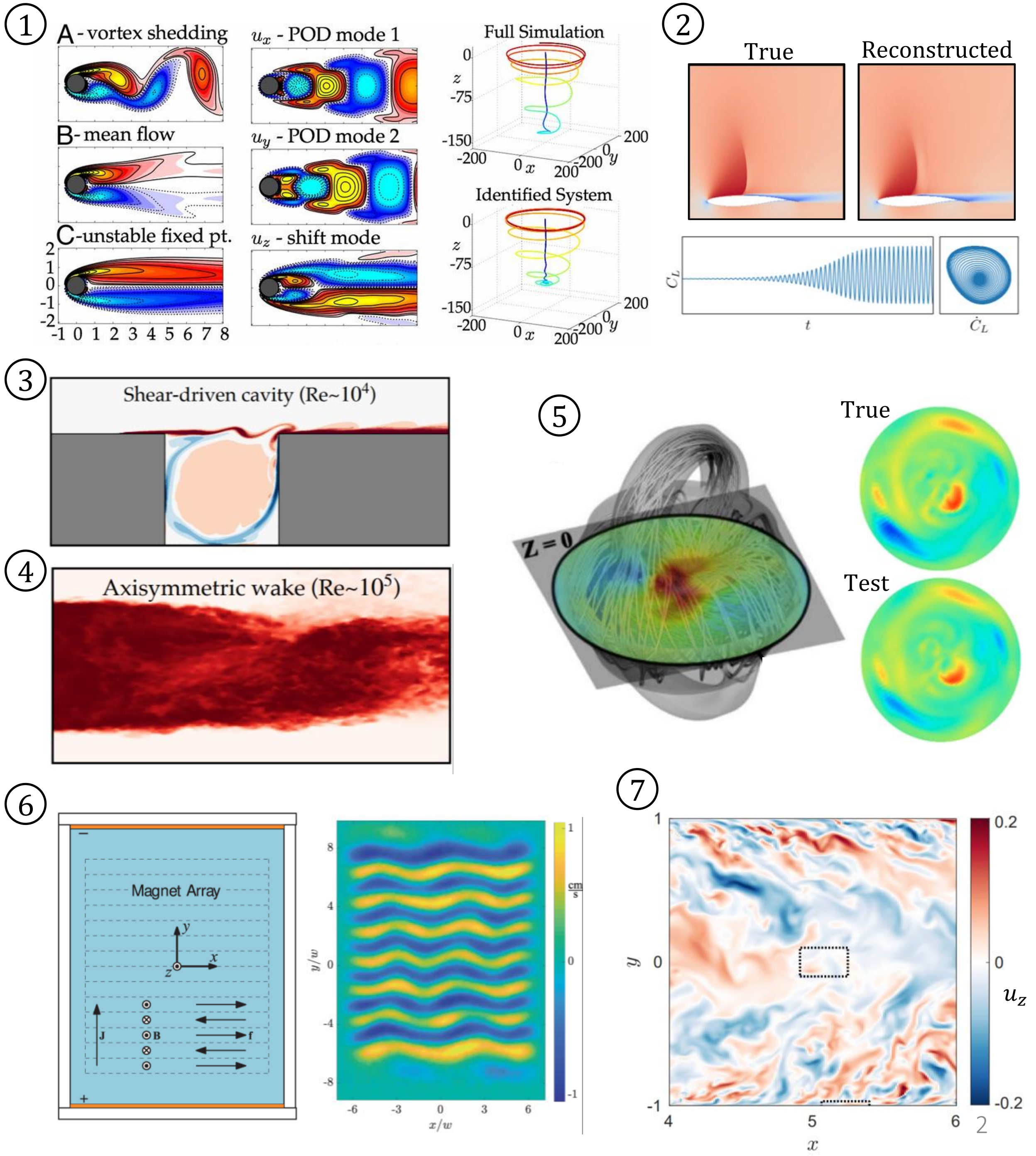
1. Flapping wing MAV
2. Renewable energy systems
3. Composite additive manufacturing morphing wing drones
4. Morphing wings
5. Airborne wind energy

Methods

1. Co-design optimization
2. **Data driven modeling & control**
   - DMDc, SINDyC, E-SINDy

SINDy – applications

1. Vortex shedding past a cylinder
    - *Time history of POD coefficients:*
        - $\dot{x} = \mu x - \omega y + Axz$
        - $\dot{y} = \omega x + \mu y + Ayz$
        - $\dot{z} = -\lambda(z - x^2 - y^2)$
2. Shock wave dynamics 2D airfoil transonic buffet conditions
    - *Parametric c_L model for different α*
        - $c_L(r,\phi) = c_0 + c_1 r + c_2 r\cos(\varphi) + c_3 r\sin(\varphi) + c_4 r^2\cos(2\varphi) + c_5 r^2\sin(2\varphi)$
3. Cavity flow
    - *Coefficients of 2 active DMD modes*
        - $\dot{\alpha}_1 = \lambda_1\alpha_1 - \mu_1\alpha_1|\alpha_1|^2$
        - $\dot{\alpha}_5 = \lambda_5\alpha_5 - \mu_5\alpha_5|\alpha_5|^2$
4. Experimental measurements turbulent bluff body wake
    - *Statistical behavior of the CoP (learning drift and diffusion of SDE)*
        - $\dot{r} = \lambda r - \mu r^3 + \frac{\sigma^2}{2r} + \left(\sigma_0 + \sigma_1^2\right)w(t)$
5. Plasma dynamics (magnetohydrodynamics): 3D spheromak sim
    - *Dominant POD coefficient dynamics*
        - $\dot{a}_1 = 0.091\mathrm{a}_2 + 0.009\mathrm{a}_5$
        - $\dot{a}_2 = -0.091\mathrm{a}_1 + 0.008\mathrm{a}_5 - 0.011\mathrm{a}_6$
        - …
6. Experimental weakly turbulent fluid flow in a thin electrolyte layer
    - *Measured velocity field, identify PDE: form similar to N-S*
        - $\partial_t\mathbf{u} = c_1(\mathbf{u}\cdot\nabla)\mathbf{u} + c_2\nabla^2\mathbf{u} + c_3\mathbf{u} - \rho^{-1}\nabla p + \rho^{-1}\mathbf{f}$
7. Turbulent 3D channel flow ($Re = 1000$) Johns Hopkins database
    - *Identify PDEs: N-S, continuity equation, boundary conditions*
        - $\partial_t\mathbf{u} = -(\mathbf{u}\cdot\nabla)\mathbf{u} - 0.995\nabla p + 4.93\cdot 10^{-5}\nabla^2\mathbf{u}$



Workshop outline

Part 1: SINDy – Sparse Identification of Nonlinear Dynamics

- ODEs (and PDEs): unforced Duffing oscillator
- MATLAB examples
- SINDy limitations

"Vanilla" SINDy

Part 2: SINDy with control & parametric models

SINDy extensions

Part 3: Model selection

Part 4: Noise robustness: weak form & ensemble SINDy

Part 5: Open discussion / Airbus data sets

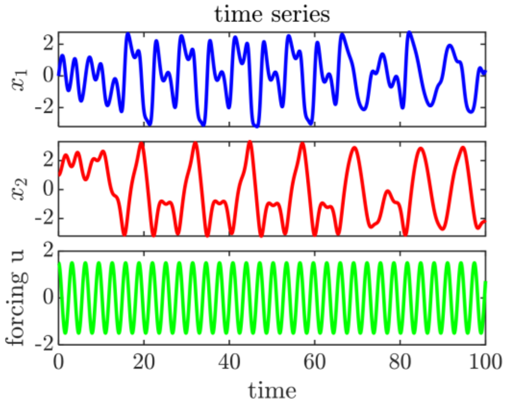
Applications

Part 1

- SINDy: Learning ODEs (and PDEs) from time series data
- **ODEs**
  1. Collect time series data & compute time derivatives
  2. Build library of nonlinear terms
  3. Sparse regression
- **MATLAB examples**
- Challenges & limitations of “Vanilla” SINDy

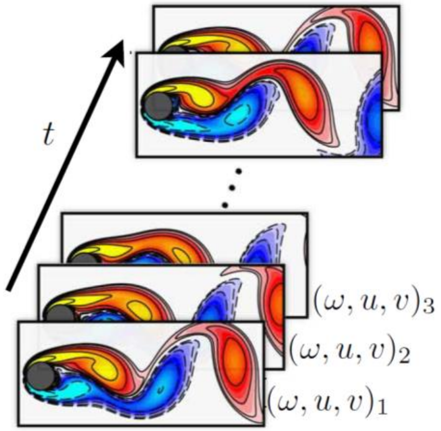
Learning ODEs (and PDEs) from data – SINDy

Data → **Dynamics (assumptions)** → **Model structure & coefficients**



$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x})$$

$$\begin{aligned} \dot{x}_1 &= -\delta x_1 - \alpha x_2 - \beta x_2^3 + u \\ \dot{x}_2 &= x_1 \\ u &= \gamma \cos(\omega t) \end{aligned}$$



$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{N}(\mathbf{u})$$

$$\omega_t + (\mathbf{u} \cdot \nabla)\omega = \frac{1}{Re}\nabla^2\omega$$

SL Brunton, JL Proctor, JN Kutz (2016) Discovering governing equations from data by sparse identification of nonlinear dynamical systems.

SINDy

***1a)* Duffing oscillator**

$$\ddot{x} \quad = \quad -\delta\dot{x} - \alpha x - \beta x^3 + \gamma\cos(\omega t)$$

***1b)* First order ODE**

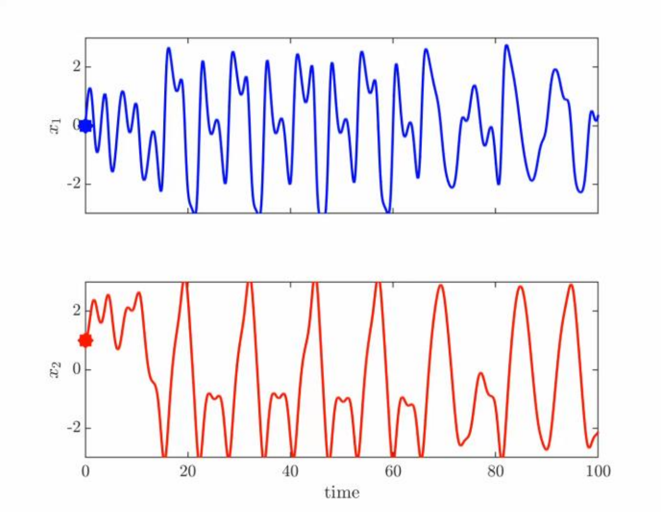
$$\dot{x}_1 \quad = \quad -\delta x_1 - \alpha x_2 - \beta x_2^3 + \gamma\cos(\omega t)$$

$$\dot{x}_2 \quad = \quad x_1$$

***1c)* Initial conditions**

$$x_1(t=0) \quad = \quad 0$$

$$x_2(t=0) \quad = \quad 1$$



SL Brunton, JL Proctor, JN Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems, PNAS 2016.

SINDy

1a) **Duffing oscillator**

$$\ddot{x} = -\delta\dot{x} - \alpha x - \beta x^3 + \cancel{\gamma\cos(\omega t)} \xrightarrow{} \gamma = 0$$

1b) **First order ODE (unforced: $\gamma = 0$)**

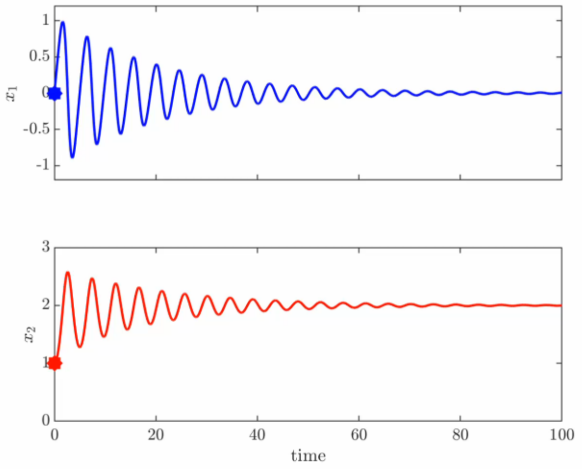
$$\dot{x}_1 = -\delta x_1 - \alpha x_2 - \beta x_2^3 + \cancel{\gamma\cos(\omega t)} \xrightarrow{} \gamma = 0$$

$$\dot{x}_2 = x_1$$

1c) **Initial conditions**

$$x_1(t=0) = 0$$

$$x_2(t=0) = 1$$



SL Brunton, JL Proctor, JN Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems, PNAS 2016.

SINDy

***1a)* Duffing oscillator**

$$\ddot{x} \quad = \quad -\delta\dot{x} - \alpha x - \beta x^3 + \gamma\cos(\omega t)$$

***1b)* First order ODE** (unforced: $\gamma = 0$)

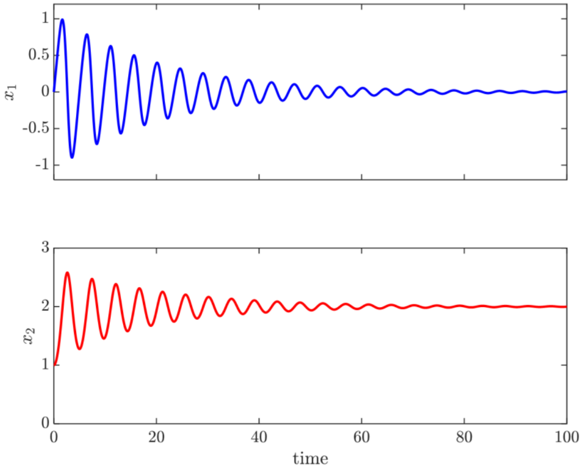
$$\dot{x}_1 \quad = \quad -\delta x_1 - \alpha x_2 - \beta x_2^3$$

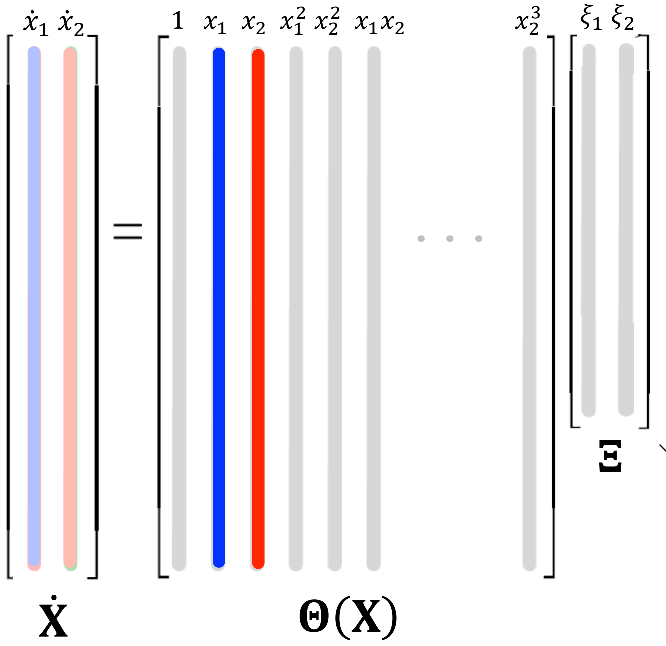
$$\dot{x}_2 \quad = \quad x_1$$

***1c)* Initial conditions**

$$x_1(t=0) \quad = \quad 0$$

$$x_2(t=0) \quad = \quad 1$$

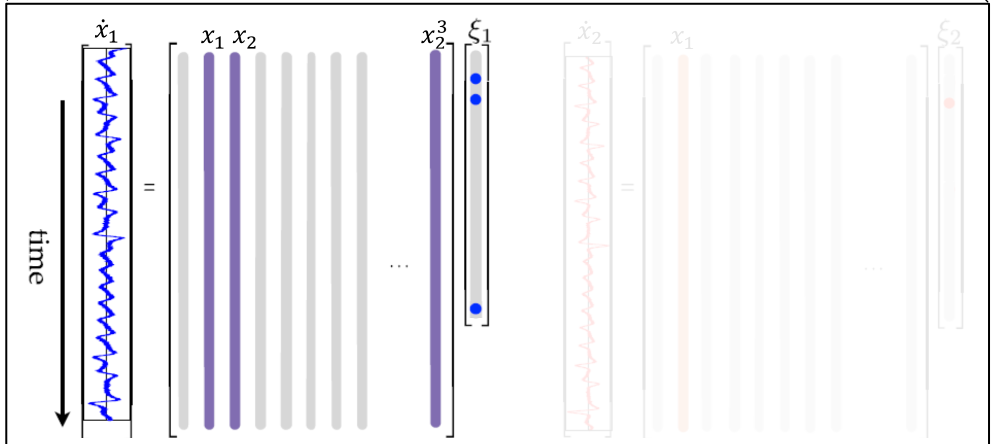




Sparse regression: penalised least squares

$\rightarrow$ $\hat{\xi}_k = \mathrm{argmin}_{\xi_k} \ \|\dot{\mathbf{X}}_k - \boldsymbol{\Theta}(\mathbf{X})\boldsymbol{\xi}_k\|_2^2 \ + \ \lambda\|\boldsymbol{\xi}_k\|_0$

$\rightarrow$ sequential thresholded least squares algorithm



***2)* Sparse Regression to Solve for Active Terms in the Dynamics**

SL Brunton, JL Proctor, JN Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems, PNAS 2016.

SINDy

Sparse regression: penalised least squares

$$\rightarrow\ \hat{\boldsymbol{\xi}}_k = \operatorname{argmin}_{\boldsymbol{\xi}_k} \left\|\dot{\mathbf{X}}_k - \boldsymbol{\Theta}(\mathbf{X})\boldsymbol{\xi}_k\right\|_2^2 \ +\ \lambda\|\boldsymbol{\xi}_k\|_0$$

```
function Xi = sparsifyDynamics(Theta,dXdt,lambda,n)
% Compute Sparse regression: sequential least squares
Xi = Theta\dXdt;  % Initial guess: Least-squares

% Lambda is our sparsification knob.
for k=1:10
    smallinds = (abs(Xi)<lambda);  % Find small coefficients
    Xi(smallinds)=0;               % and threshold
    for ind = 1:n                  % n is state dimension
        biginds = ~smallinds(:,ind);
% Regress dynamics onto remaining terms to find sparse Xi
        Xi(biginds,ind) = Theta(:,biginds)\dXdt(:,ind);
    end
end
```

SL Brunton, JL Proctor, JN Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems, PNAS 2016.

SINDy

1a) **Duffing oscillator**

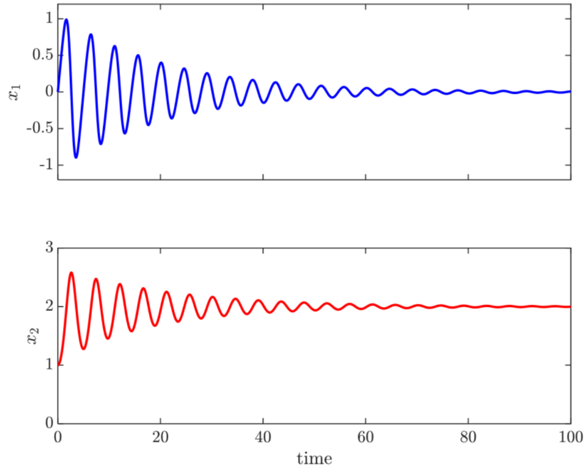
$$\ddot{x} \;=\; -\delta\dot{x} - \alpha x - \beta x^3 + \gamma\cos(\omega t)$$

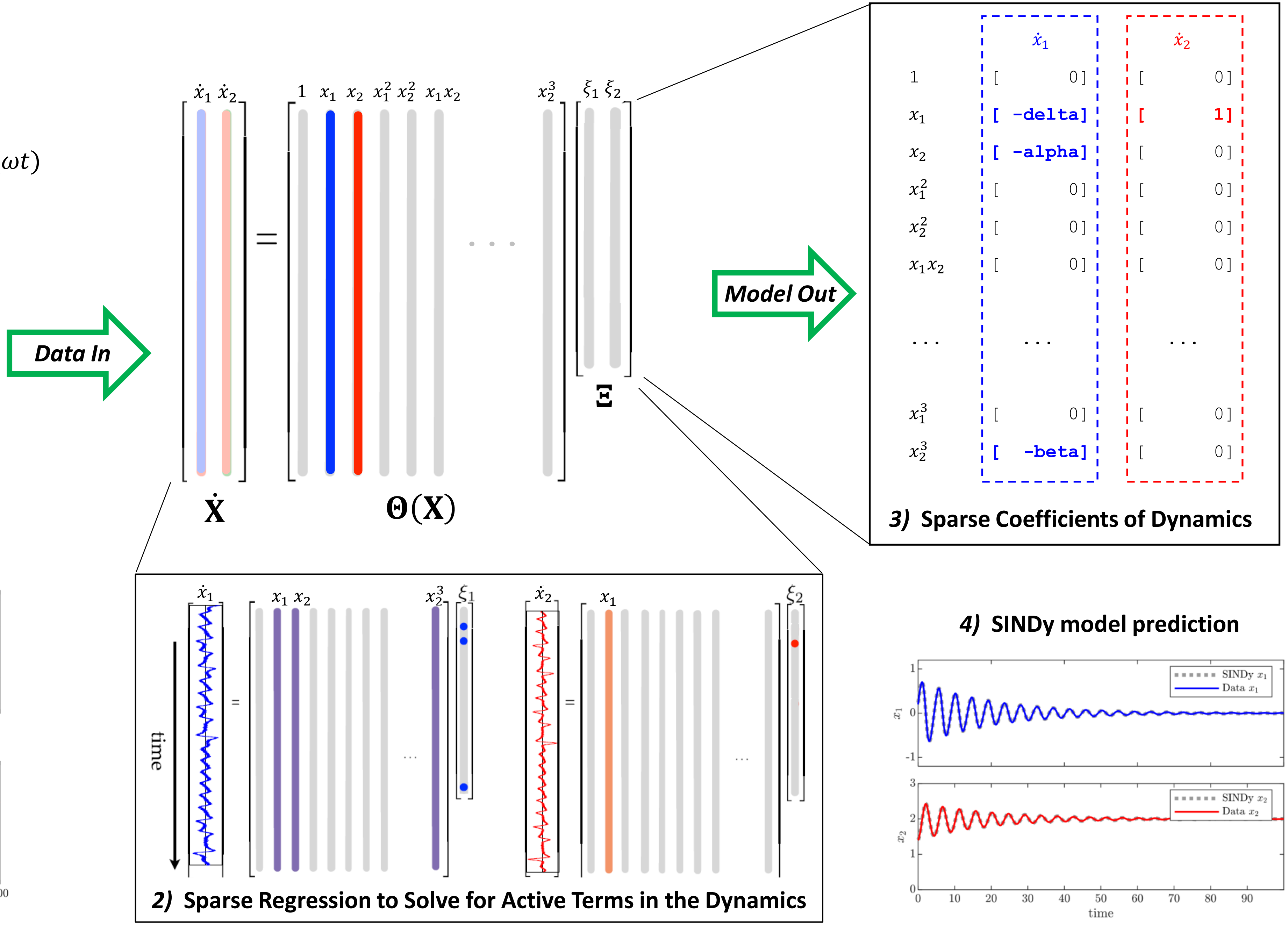
1b) **First order ODE** (unforced: $\gamma = 0$)

$$\dot{x}_1 \;=\; -\delta x_1 - \alpha x_2 - \beta x_2^3$$
$$\dot{x}_2 \;=\; x_1$$

1c) **Initial conditions**

$$x_1(t=0) \;=\; 0$$
$$x_2(t=0) \;=\; 1$$





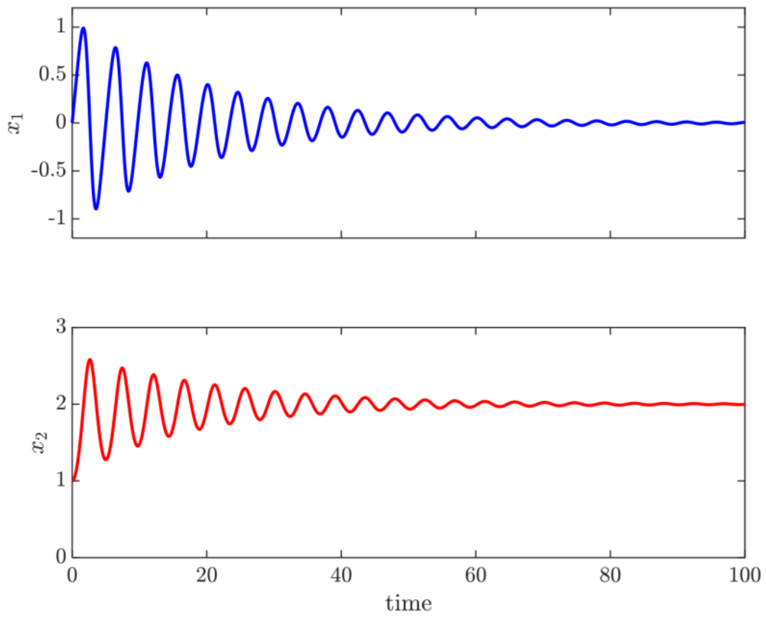
SL Brunton, JL Proctor, JN Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems, PNAS 2016.

MATLAB tutorial: identify ODE → https://github.com/urban-fasel

Duffing oscillator (unforced)

$$
\begin{aligned}
\dot{x}_1 &= -\delta x_1 - \alpha x_2 - \beta x_2^3 \\
\dot{x}_2 &= x_1
\end{aligned}
$$

Data: time series x_1, x_2



No MATLAB installed?

→ Run the tutorials on **MATLAB online**: https://matlab.mathworks.com/

→ Or use **PySINDy** (next slide): https://github.com/dynamicslab/pysindy

→ Or **Julia** SciML: https://docs.sciml.ai/DataDrivenDiffEq/stable/#Package-Overview

Challenges / limitations “vanilla” SINDy

Challenges / limitations "vanilla" SINDy

Data → how much and what quality is needed?

Library → how to choose an effective library of candidate terms?

Optimization → what algorithm/regularization to use?

Model selection → how to select models / tune hyperparameters?

(**Coordinates** → do we measure the right variables?)

MATLAB tutorial – challenges / limitations of SINDy

Additional MATLAB tutorials

- Different optimizers: STLS vs LASSO
- Different ODEs and PDEs (Roessler, Burger's → github autumn school Amsterdam)
- …

Workshop outline

Part 1: SINDy – Sparse Identification of Nonlinear Dynamics

- ODEs (and PDEs): unforced Duffing oscillator
- MATLAB examples
- SINDy limitations

"Vanilla" SINDy

Part 2: SINDy with control & parametric models

SINDy extensions

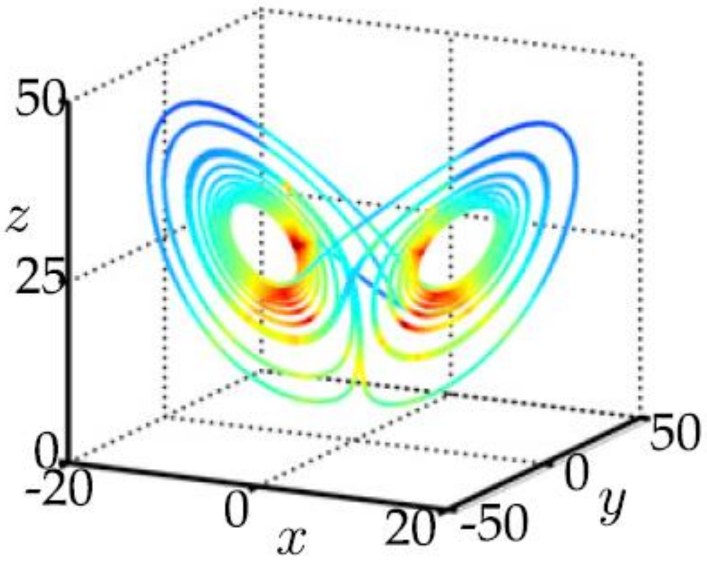
Part 3: Model selection

Part 4: Noise robustness: weak form & ensemble SINDy

Part 5: Open discussion / Airbus data sets

Applications

Python tutorial – identify ODE



1. PySINDy

- SINDy python package: [JOSS article]
- GitHub: https://github.com/dynamicslab/pysindy
  - Check out the **binder notebook** examples!
- PySINDy lectures: notebooks and YouTube videos
  - GitHub interactive notebook
  - Tutorial videos Alan Kaptanoglu

2. Lorenz system ODE tutorial

1. Start with the feature overview tutorial in PySINDy to **identify the Lorenz system**
2. Try to identify the **Rossler attractor**: `from pysindy.utils import rossler`
   - Generate data: `x_train = solve_ivp(rossler, ...`
3. Test other data sets generated from different ODEs → ODEs in PySINDy
   - Large library of chaotic systems: https://github.com/williamgilpin/dysts