SINDy with control & parametric models

Filton workshop 2024

Urban Fasel

Imperial College London

Literature

- **Control**
  - E Kaiser, JN Kutz, SL Brunton (2018) Sparse identification of nonlinear dynamics for model predictive control in the low-data limit.
  - U Fasel, E Kaiser, JN Kutz, BW Brunton, SL Brunton (2021) SINDy with Control: A Tutorial.
- **Parametric**
  - SL Brunton, JL Proctor, JN Kutz (2016) Discovering governing equations from data by sparse identification of nonlinear dynamical systems.
  - ZG Nicolaou, G Huo, Y Chen, SL Brunton, JN Kutz (2023) Data-driven discovery and extrapolation of parameterized pattern-forming dynamics.

Duffing Oscillator

Duffing Oscillator

1a) **Duffing oscillator**

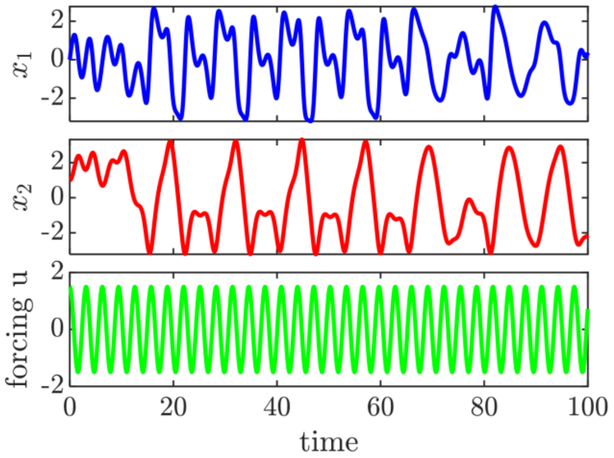
$$\ddot{x} = -\delta\dot{x} - \alpha x - \beta x^3 + u$$
$$u = \gamma\cos(\omega t)$$

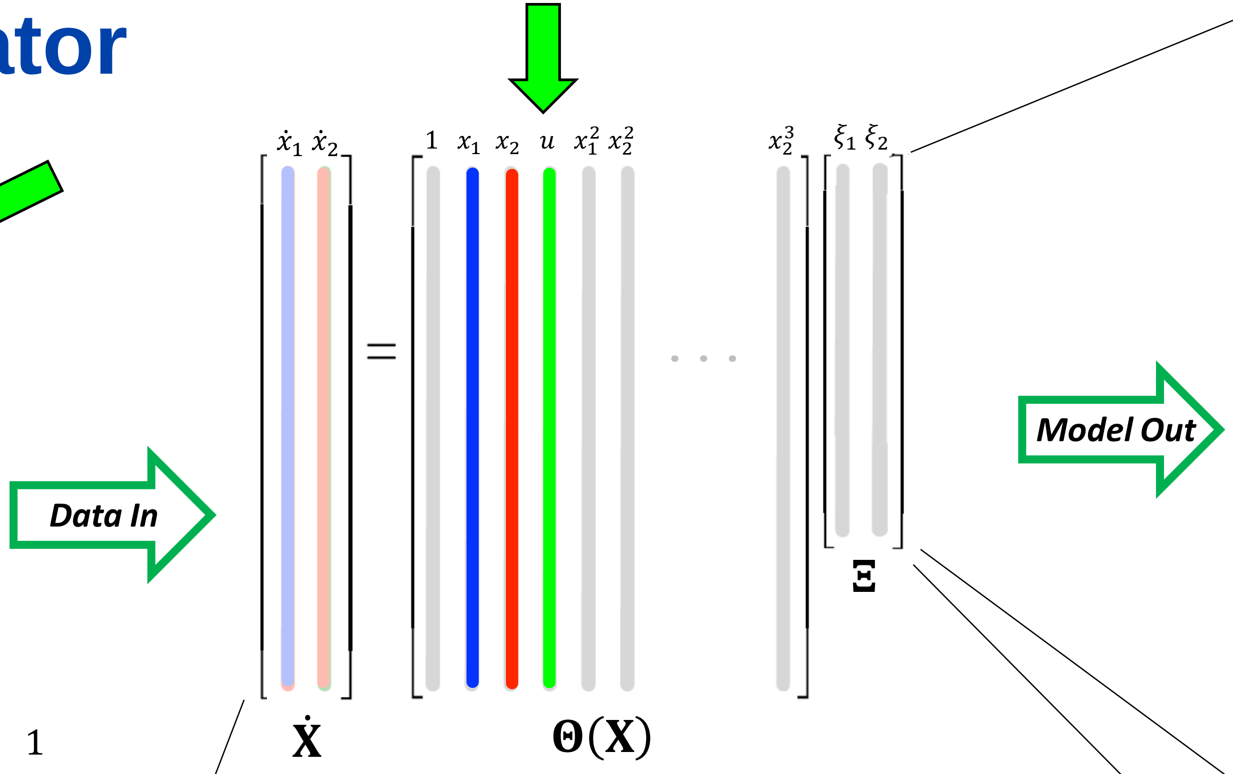
1b) **First order ODE**

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

1c) **Initial conditions**

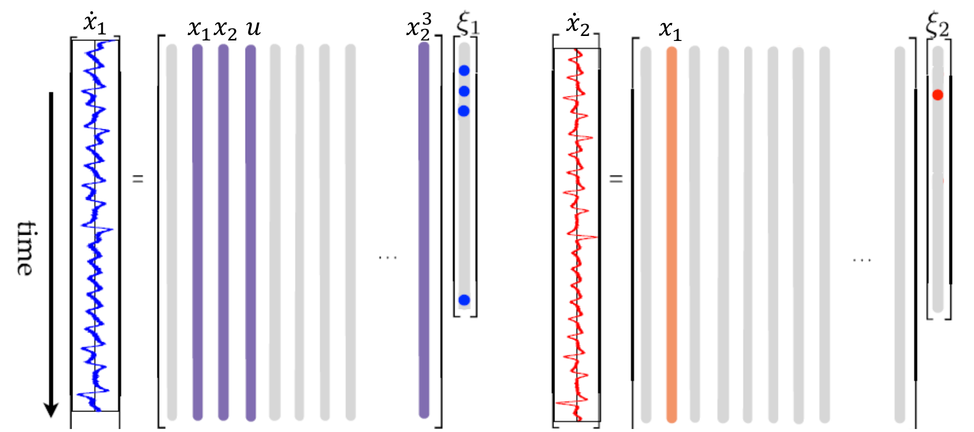
$$x_1(t=0) = 0, \quad x_2(t=0) = 1$$





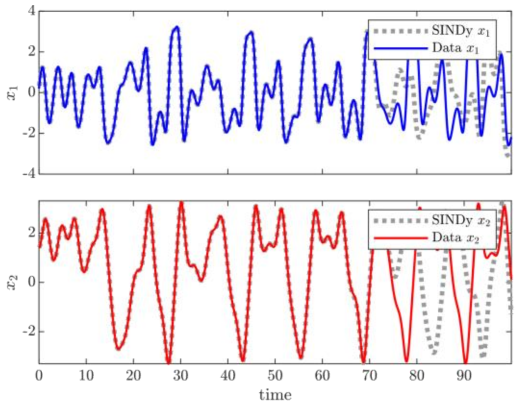
	$\dot{x}_1$	$\dot{x}_2$
1	[0]	[0]
x_1	[-delta]	[1]
x_2	[-alpha]	[0]
u	[gamma]	[0]
x_1^2	[0]	[0]
x_2^2	[0]	[0]
x_1x_2	[0]	[0]
...	...	...
x_1^3	[0]	[0]
x_2^3	[-beta]	[0]

3) **Sparse Coefficients of Dynamics**



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

4) **SINDy model prediction**



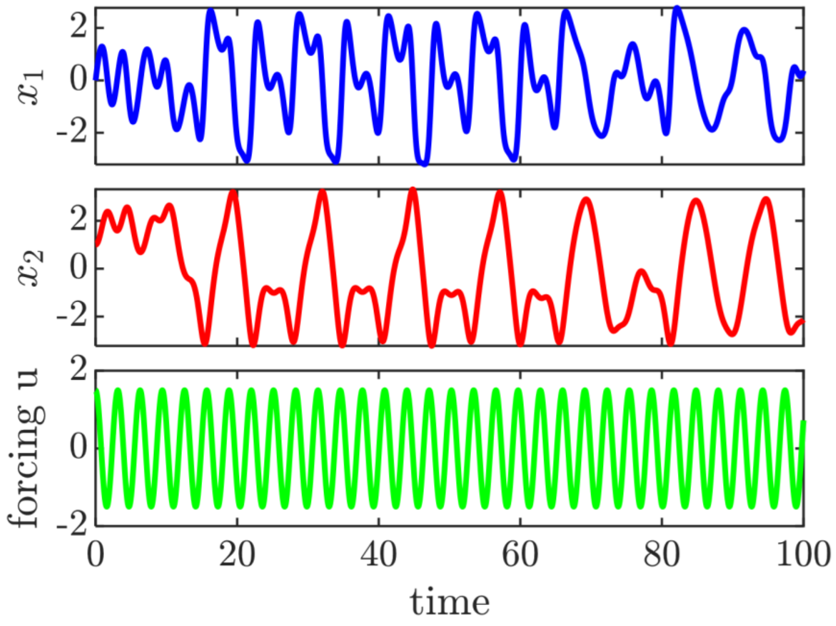
SL Brunton, JL Proctor, JN Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems, PNAS 2016.

MATLAB: Duffing Oscillator $\rightarrow$ https://github.com/urban-fasel

Duffing oscillator

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

Data: time series x_1, x_2, u

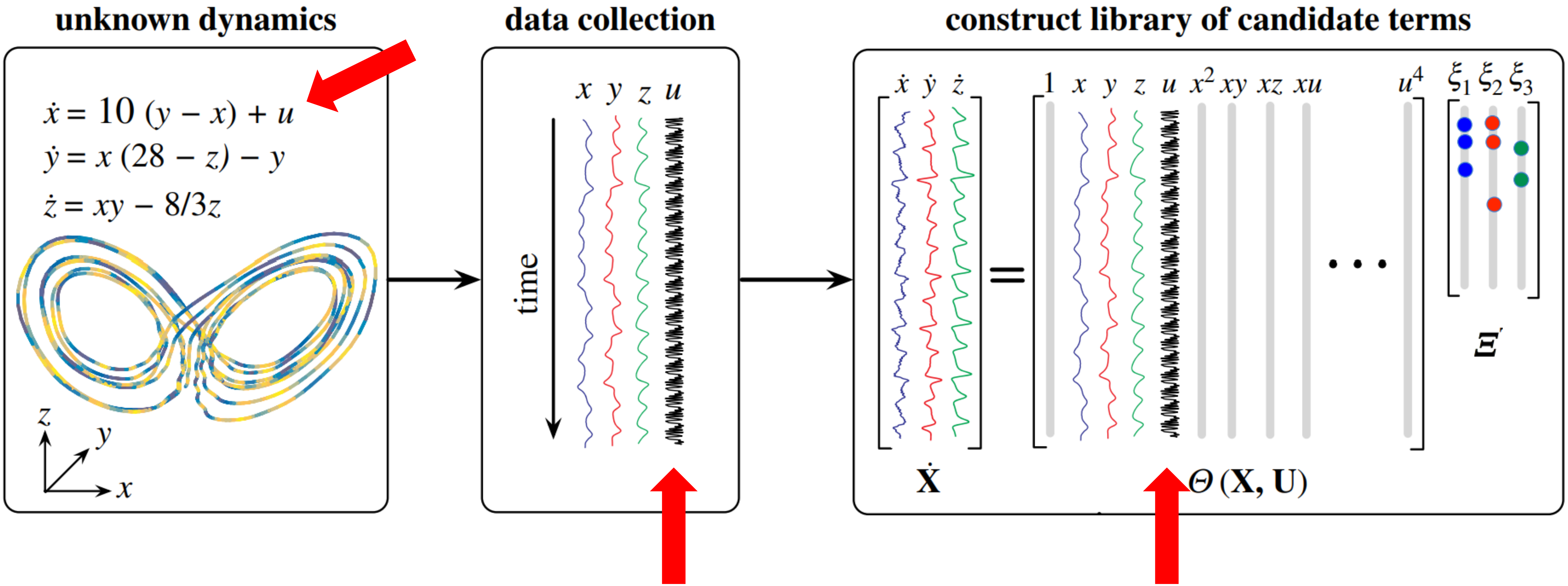


No MATLAB installed?

- → 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

Control

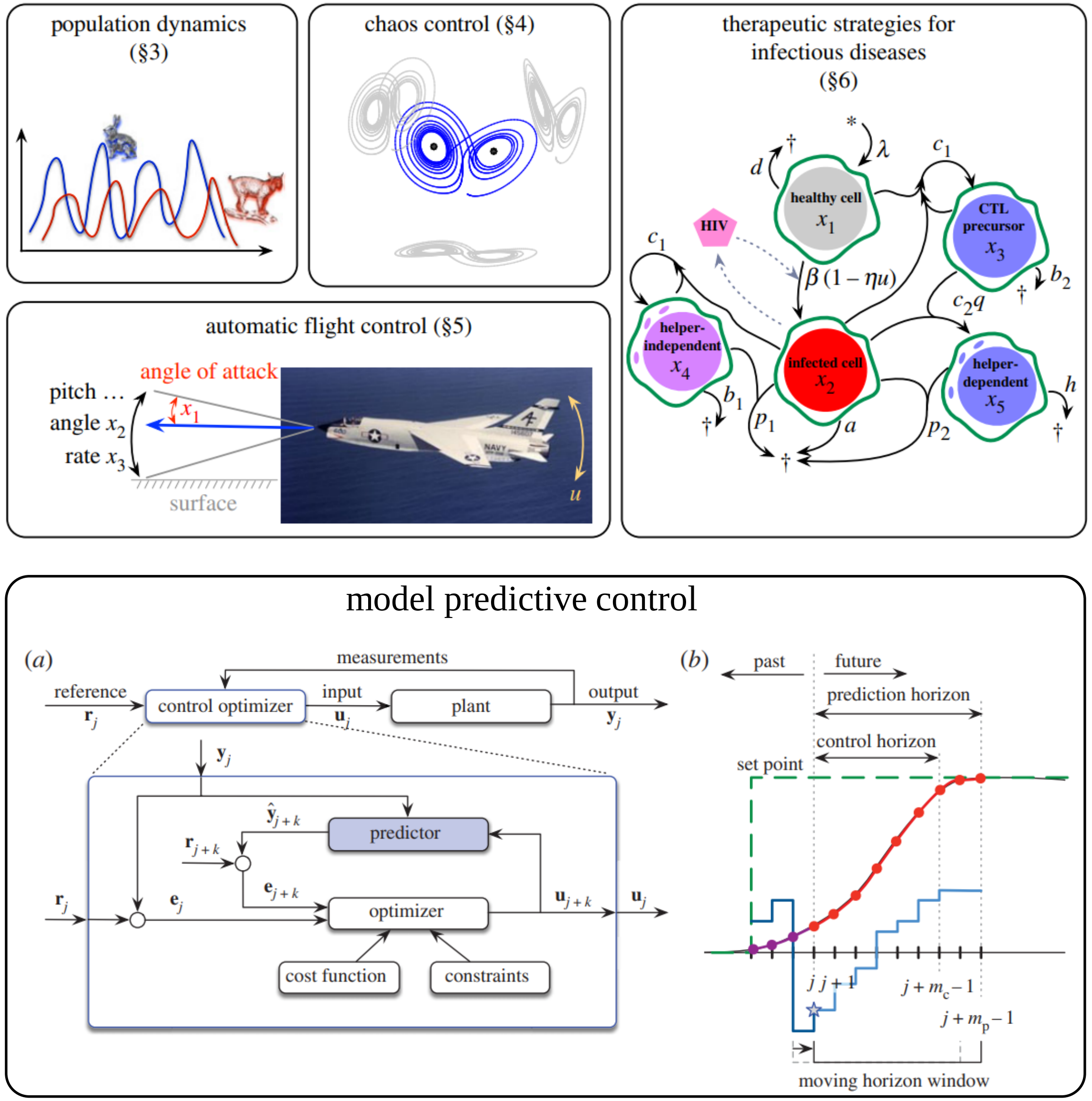
Control



Challenge

- How to (safely) excite the system while maximizing information gain?

Control



E Kaiser, JN Kutz, SL Brunton (2018) Sparse identification of nonlinear dynamics for model predictive control in the low-data limit

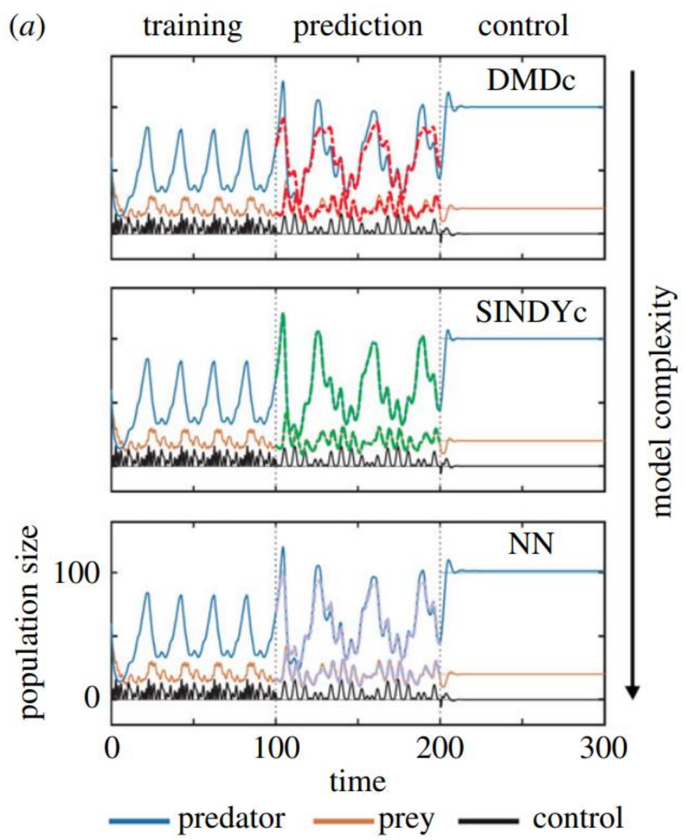
SINDy-MPC

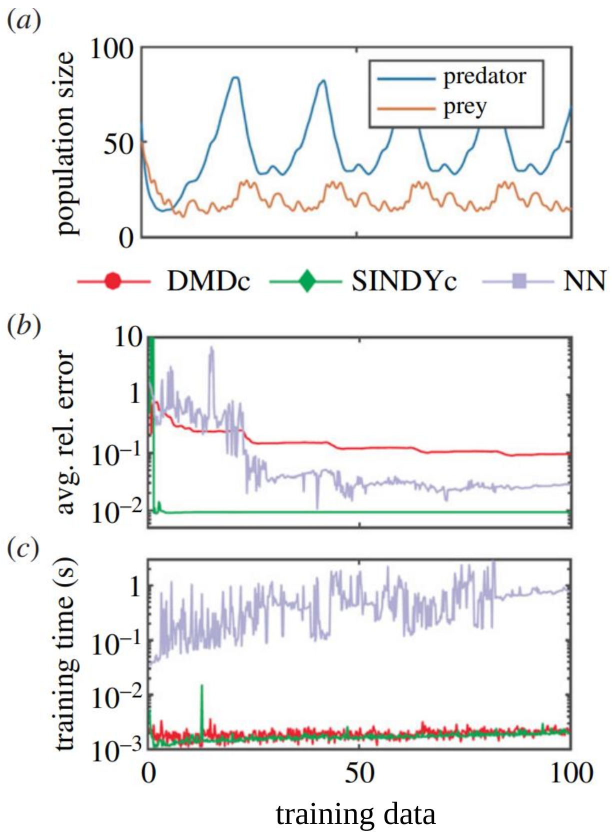
- Using SINDy models for nonlinear control
  - Control population dynamics
  - Stabilize fixed point of chaotic system
  - Optimize therapeutic strategies
  - Aircraft flight control

Model Predictive Control

- Use model to optimize control sequence
  - Reaching set point based on model predictions
  - Trade-off between control expenditure and reference tracking
  - Powerful because it can consider constraints

Control

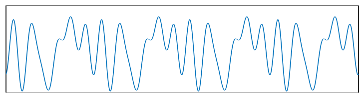


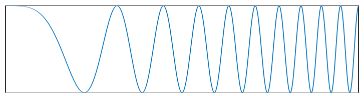


Predator prey population dynamics

- **Objective**: stabilize population (fixed point)
- **ODE**: $\dot{x}_1 = ax_1 - bx_1x_2$

  $\dot{x}_2 = -cx_2 + dx_1x_2 + u$
- **Training**: how to force system?
  - Schroeder sweep: phase-shifted sum of sines
  - Chirp: frequency increase with time





Comparisons DMD and NN

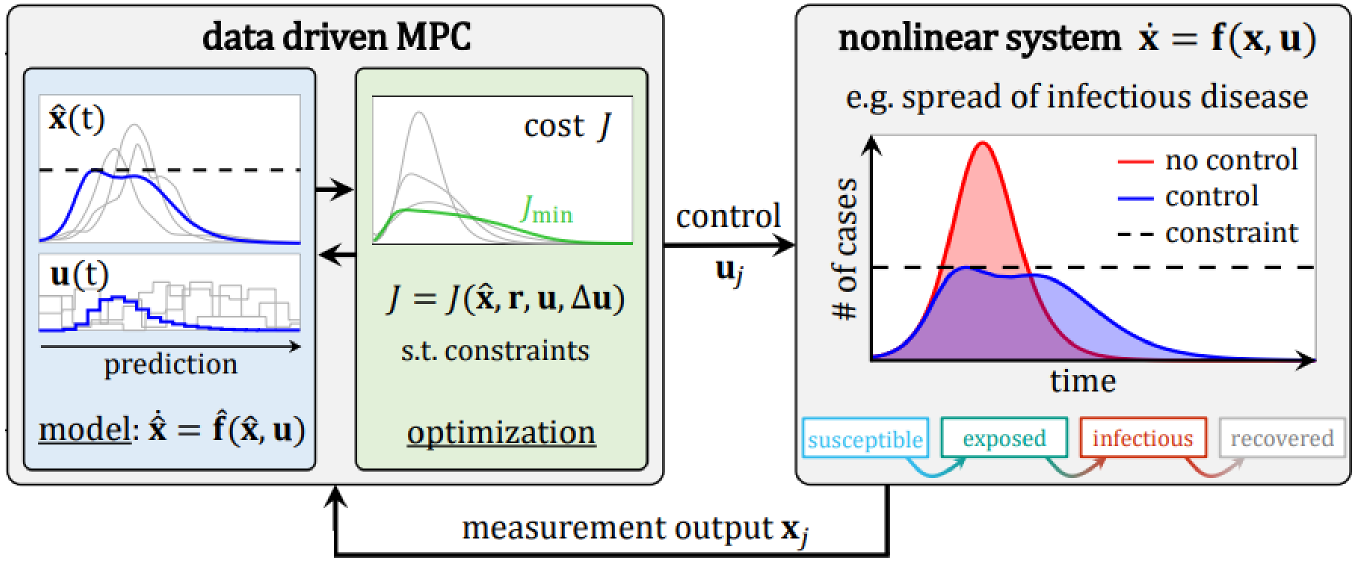
- DMD performs surprisingly well
- NN needs more data than SINDy to train an accurate model for prediction

E Kaiser, JN Kutz, SL Brunton (2018) Sparse identification of nonlinear dynamics for model predictive control in the low-data limit

Control MATLAB tutorial

IEEE CDC tutorial paper

- MATLAB tutorial
- U Fasel, E Kaiser, JN Kutz, BW Brunton, SL Brunton (2021) SINDy with Control: A Tutorial.
- https://github.com/urban-fasel/SEIR_SINDY_MPC
  - **Line 100**: add control input to time series data array **Line 107f**: Build library and identify SINDy model

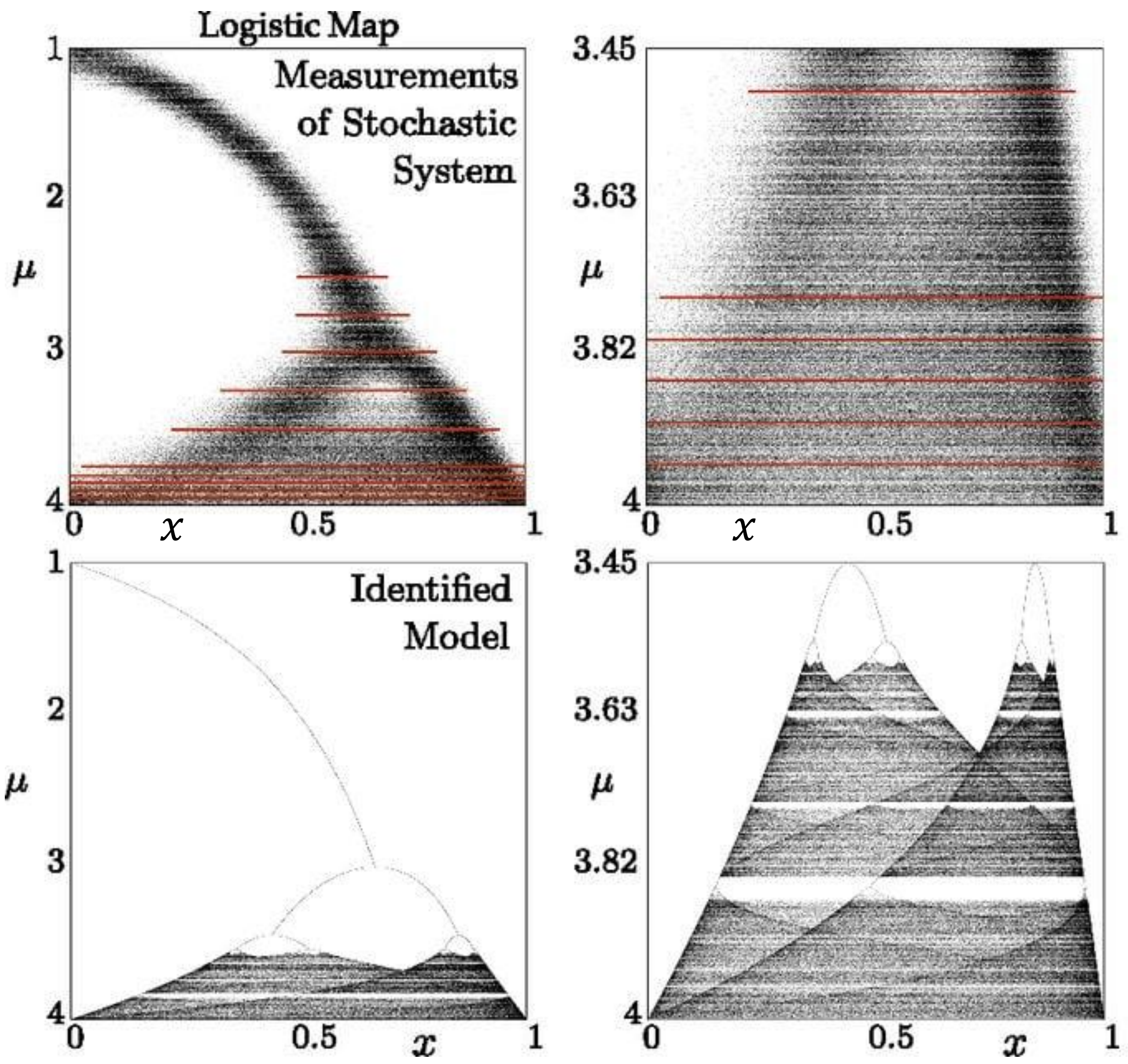



```
%% Initialize MPC
pMPC = MPCparams(); % define control parameters
x = x0; uopt = pMPC.uopt0;

%% Run nonlinear MPC with full-state feedback
for i = 1:(pMPC.Duration/pMPC.Ts)
    % Cost and constraint function
    COST = @(u) CostFCN(u,x,pMPC,uopt(1));
    CONS = @(u) ConstraintFCN(u,x,pMPC);
    % Optimization
    uopt = fmincon(COST,uopt,pMPC,CONS);
    % Apply control and step one timestep forward
    x = rk4u(@SEIR,x,uopt(1),pMPC.Ts,1,[],params);
    xHistory(i,:) = x;
    uHistory(i) = uopt(1);
end
```

Bifurcation parameters

Library – bifurcation parameters



Chaotic logistic map

- $x_{k+1} = \mu x_k(1 - x_k) + \eta_k$
  - discrete time dynamics
  - μ: bifurcation parameter (chaotic for $\mu > 3.6$)
  - η_k: stochastic forcing
- Considering bifurcation parameter μ
  - same as SINDy with control variables
  - collect noisy data for 10 different values μ
  - add μ to the library (same as u in SINDy-C)
- Identify true dynamics
  - … to generate full logistics map
  - **MATLAB tutorial (also in PySINDy)**

Additional examples / tutorials

- **MATLAB**
  - SINDy with control: run infectious disease dynamics MPC tutorial
  - Bifurcation parameters: apply SINDy to identify Hopf normal form → example from 2016 paper
    - $\dot{x} = \mu x + \omega y - Ax(x^2 + y^2)$
    - $\dot{y} = -\omega x + \mu y - Ay(x^2 + y^2)$

- **Python**
  - PySINDy **control**
    - 1_feature_overview: SINDy with control (SINDYc)
  - PySINDy **MPC**
    - https://github.com/CyrusLiu20/PySINDy-with-model-predictive-control/tree/main
  - PySINDy **parametric**
    - 1_feature_overview: SINDy with control parameters (SINDyCP)
    - SINDyCP for discovery of parametrized pattern formation