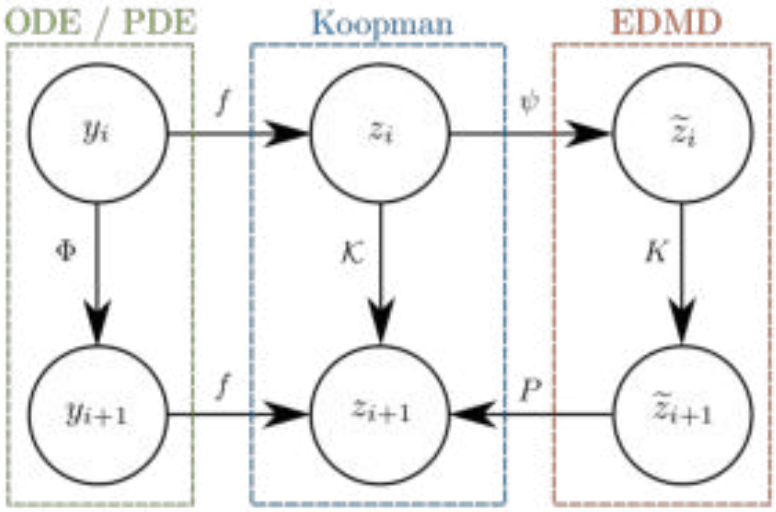
PADERBORN UNIVERSITY



Mini-Workshop on Model Reduction and Control

Efficient data-driven prediction and control of complex systems via the Koopman operator

SEBASTIAN PEITZ

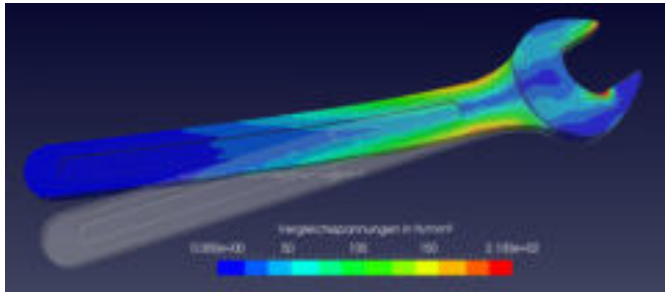
joint work with S. Klus, S. E. Otto & C. W. Rowley

FAU, 2022/05/24

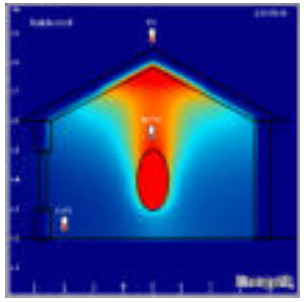
Supported by the Priority Programme SPP 1962 *Non-smooth and Complementarity-based Distributed Parameter Systems* of the DFG.

Optimal control of PDEs:

Aerodynamics



Mechanical strain



Heat flow

$$\dot{y}(x,t) = G(y(x,t),u(t)) \qquad (x,t) \in \Omega \times (t_0,t_e]$$

$$a(x,t)\frac{\partial y}{\partial n}(x,t) = c(x,t) - b(x,t)y(x,t) \qquad (x,t) \in \Gamma \times (t_0,t_e]$$

$$y(x,t_0) = y_0(x) \qquad x \in \Omega$$

- Standard discretization: FEM $\rightarrow$ Expensive
- Aim: Approximate dynamics **relevant for the control task** via reduced order model while preserving convergence properties

Outline

- Optimal control and surrogate modeling
- The Koopman operator for control
  - A few basics
  - EDMD and dictionary selection
  - Control using the Koopman operator & generator
  - Online learning
  - Some examples
- Conclusion & open questions

Optimal Control

We want to solve the general problem

$$\min_{u \in L^\infty([t_0,t_e],U)} J(y) = \int_{t_0}^{t_e} P(y(t))\,dt$$
$$\text{s.t.} \quad \dot{y}(t) = g(y(t), u(t)), \qquad y(t_0) = y_0.$$

Setting $u(t) = u_i \in U$ constant over the interval $[t_i, t_{i+1})$, we obtain the discrete-time version

$$\min_{u \in U^p} J(y) = \sum_{i=0}^{p-1} P(y_{i+1}) \tag{I}$$
$$\text{s.t.} \quad y_{i+1} = \Phi(y_i, u_i),$$

where evaluating $\Phi(y_i, u_i) = y_i + \int_{t_i}^{t_{i+1}} g(y(t), u_i)\,dt = y_{i+1}$ may be expensive.

Surrogate modeling

A highly popular approach to reduce the numerical cost is surrogate modeling[1]:

$$\text{Replace} \qquad y_{i+1} = \Phi(y_i, u_i) \qquad \text{by} \qquad z_{i+1} = \Phi^r(z_i, u_i)$$

for the dynamics of an observed quantity $z = f(y)$.

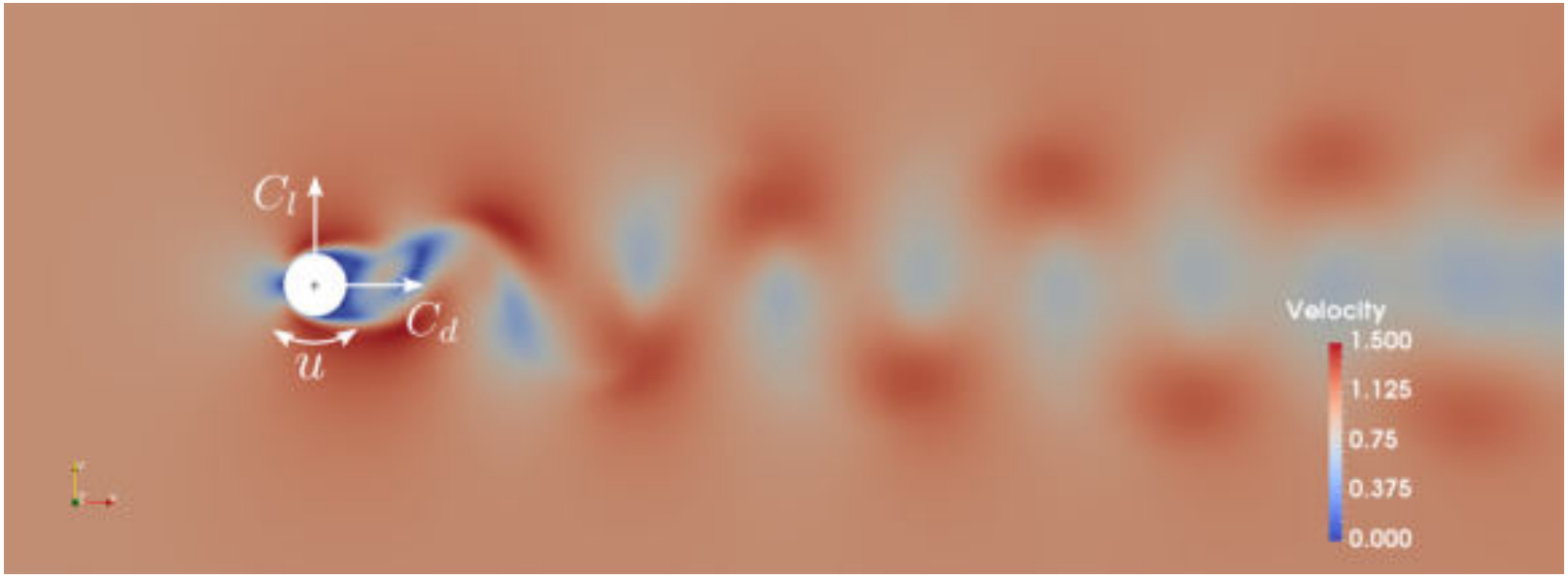
The function $f : Y \to \mathbb{R}^q$ is an observable that takes arbitrary measurements from the full state $y(t) \in Y$, e.g.,

- full-state observation (i.e., $z = y$),
- partial observations or point-wise measurements,
- arbitrary nonlinear functions of the state (e.g., forces acting on a surface).

[1] Benner, Mehrmann, Sorensen 2005; Benner, Gugercin, Willcox 2015

Example: Fluid dynamics (Navier-Stokes Equations)

Fluid enters from the left and faces a cylindrical obstacle. The cylinder can be rotated to influence the flow.



Control: We are not interested in the flow field $y(x, t)$ with $N \approx 66\,000$ DOF per time step, but want to control the forces acting on the cylinder (lift C_l and drag C_d).

Proper Orthogonal Decomposition (POD)[2]

Galerkin-Ansatz for partial differential equations (PDEs):

$$y(x,t) \approx \sum_{i=1}^{\ell} z_i(t)\psi_i(x)$$

with global basis functions $\psi_i(x)$ and time dependent coefficients $z_i(t)$.

Insert into (weak formulation) of the PDE:

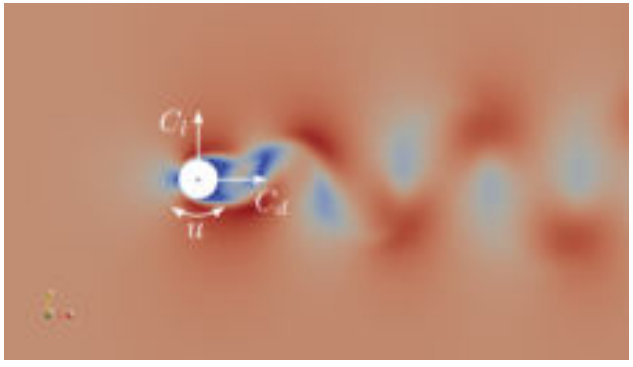
$$\dot{z} = G^r(z,u).$$

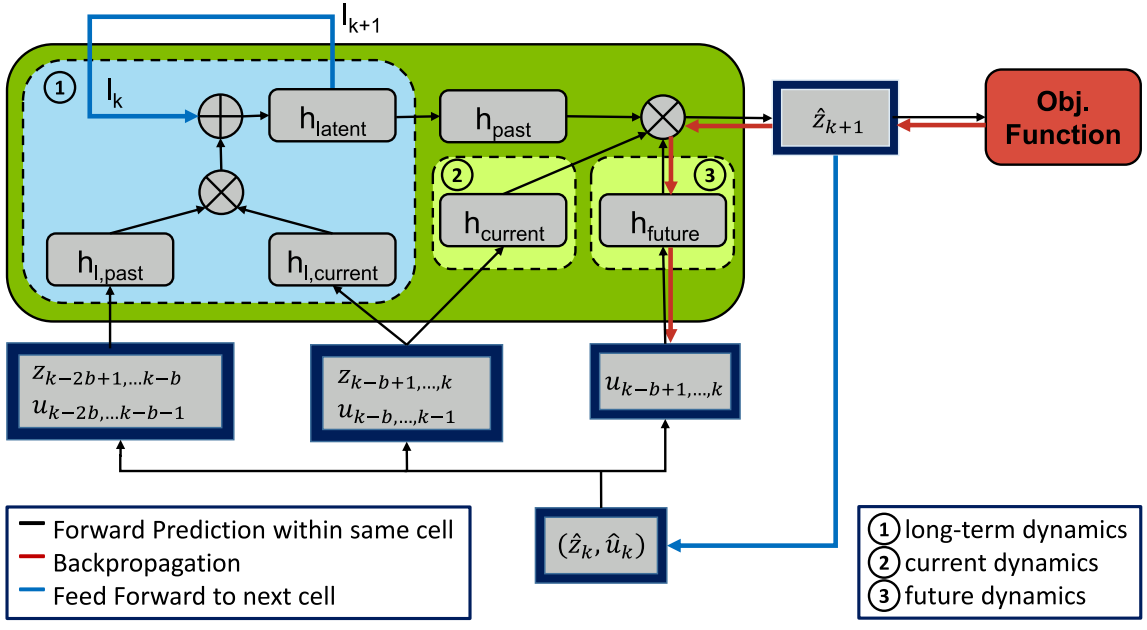
$\rightarrow$ Ordinary differential equation of dimension $\ell \ll N$ for the coefficients z.
$\rightarrow$ challenging for complex dynamics!

[2] Sirovich 1987, Holmes et al. 2012, Kunisch & Volkwein 1999, P. et al. 2019

Deep Learning

Direct modeling of $z = (C_l, C_d)$ via deep neural network architecture[3]





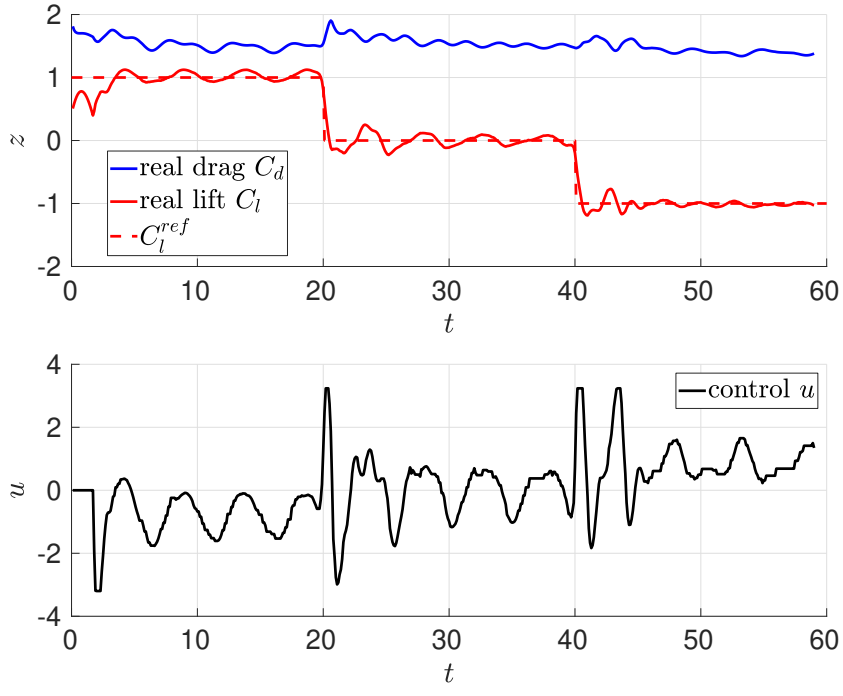
$$\min_u \sum_{i=1}^{p} \left(z_{1,i} - z_i^{\text{ref}} \right)^2$$

$$\text{s.t.} \quad z_{i+1} = \Phi^r(z_i, u_i)$$

[3] Bieker, P., Brunton, Kutz & Dellnitz, Theor. and Comp. Fluid Dyn. 2020

Deep Learning

Direct modeling of $z = (C_l, C_d)$ via deep neural network architecture[3]

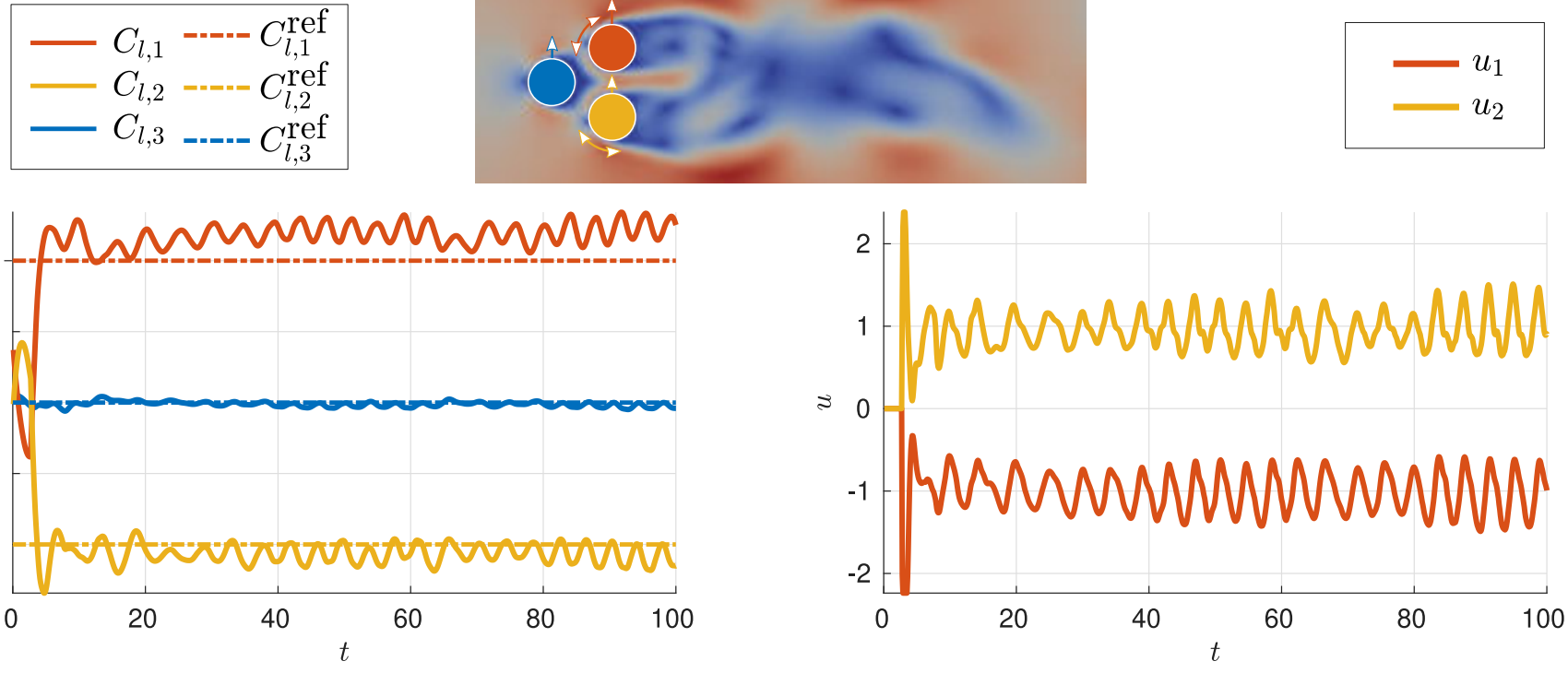


Model Predictive Control

[3] Bieker, P., Brunton, Kutz & Dellnitz, Theor. and Comp. Fluid Dyn. 2020

Deep Learning

Direct modeling of $z = (C_l, C_d)$ via deep neural network architecture[3]

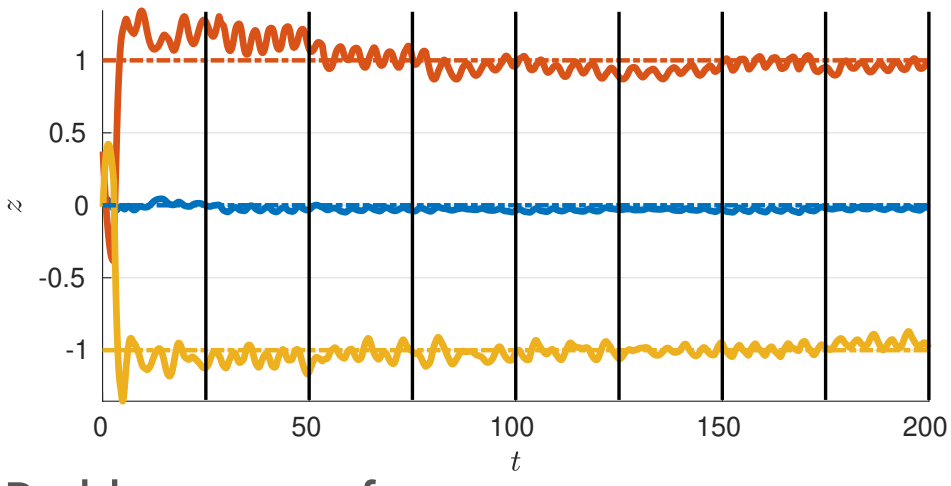


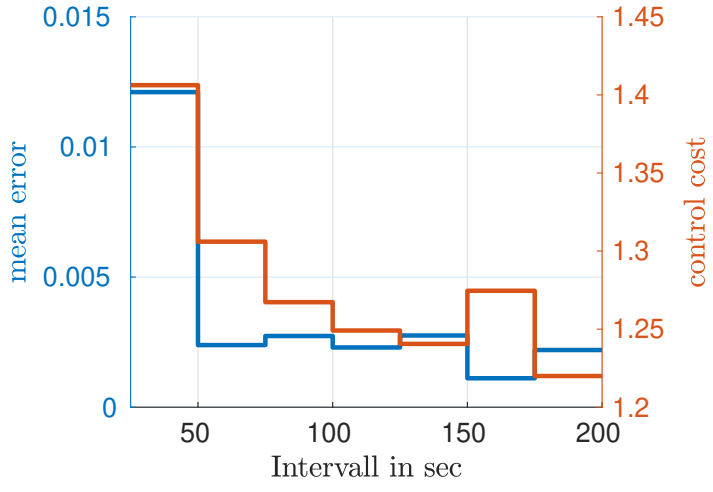
[3] Bieker, P., Brunton, Kutz & Dellnitz, Theor. and Comp. Fluid Dyn. 2020

Deep Learning

Direct modeling of $z = (C_l, C_d)$ via deep neural network architecture[3]

Batch-wise online learning using incoming sensor (i.e., streaming) data:





Problem: no performance guarantee

$\rightarrow$ Can we exploit the fact that we are dealing with dynamical systems?

[3] Bieker, P., Brunton, Kutz & Dellnitz, Theor. and Comp. Fluid Dyn. 2020

Outline

- Optimal control and surrogate modeling
- The Koopman operator for control
  - A few basics
  - EDMD and dictionary selection
  - Control using the Koopman operator & generator
  - Online learning
  - Some examples
- Conclusion & open questions

Koopman operator[4]

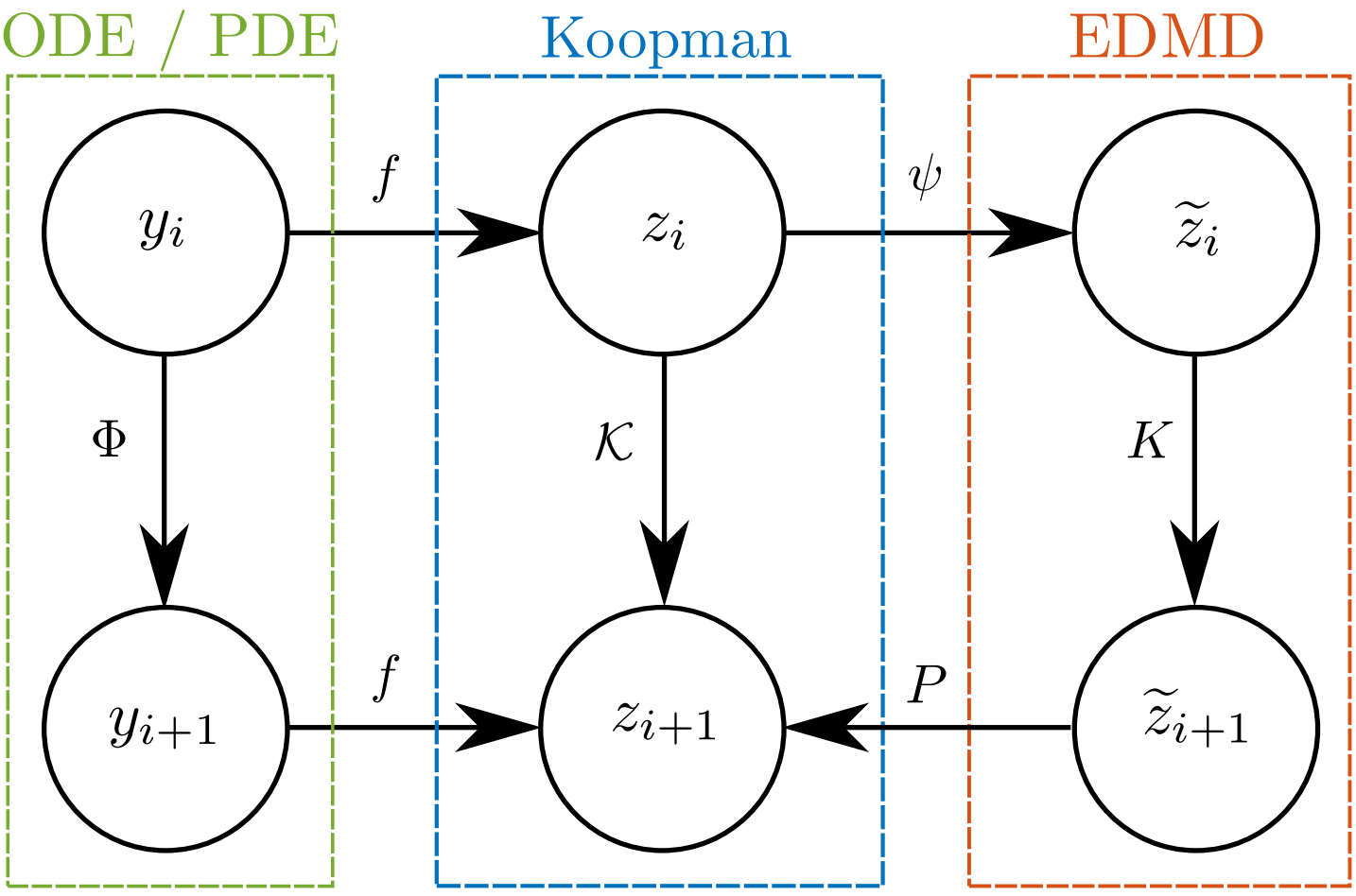
Definition: Let $\Phi^t : \mathbb{Y} \to \mathbb{Y}$ be a discrete-time dynamical system (or the time-T map of a continuous-time system with time step t) defined on a state space $\mathbb{Y}$, and $f \in \mathcal{F}, f : \mathbb{Y} \to \mathbb{R}$ an observable of this system.
The Koopman semigroup $\mathcal{K}^t : \mathcal{F} \to \mathcal{F}$ is defined by

$$(\mathcal{K}^t f)(y) = f(\Phi^t(y)).$$

- The Koopman operator is infinite-dimensional but linear.
- The Koopman operator is the adjoint of the Perron–Frobenius operator $\mathcal{P}^t$, i.e. $\langle \mathcal{P}^t f, g \rangle = \langle f, \mathcal{K}^t g \rangle$.
- Mental picture: $\mathcal{K}^t$ describes the evolution of observables, $\mathcal{P}^t$ the evolution of densities.

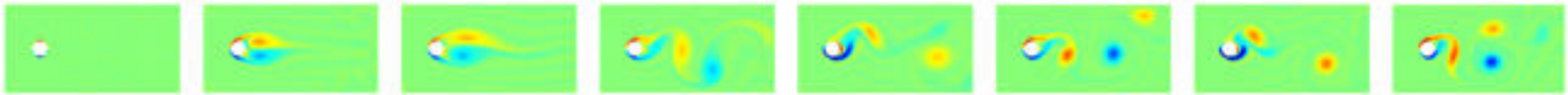
[4] Koopman 1931, von Neumann 1932, Mezić 2005

Koopman operator[3]



[3] Koopman 1931, von Neumann 1932, Mezić 2005

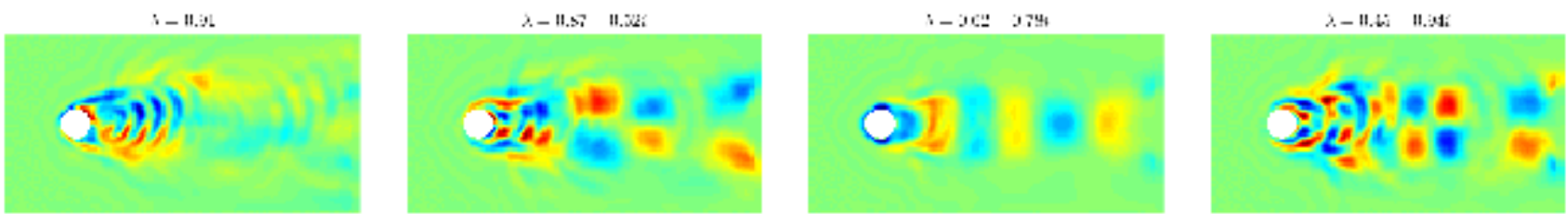
Dynamic Mode Decomposition (DMD)[4]



- DMD requires data, i.e. a set of values $z_i = f(y_i)$ and the corresponding $\hat{z}_i = f(\Phi(y_i))$ values, $i = 1, \ldots, m$,

$$Z = \begin{bmatrix} z_1 & z_2 & \cdots & z_m \end{bmatrix} \quad \text{and} \quad \hat{Z} = \begin{bmatrix} \hat{z}_1 & \hat{z}_2 & \cdots & \hat{z}_m \end{bmatrix}.$$

- Assume there exists a linear operator K such that $\hat{z}_i = Kz_i$, then $K = \hat{Z}Z^+$ minimizes $\left\| KZ - \hat{Z} \right\|_F$ $\Rightarrow$ simple linear regression!
- The eigenvectors of K are called DMD modes, the complex eigenvalues indicate growth/decay and phase



[4] Schmid 2010, Tu et al. 2014

Extended DMD (EDMD)[5]

- DMD breaks down rather quickly, as linear dynamics of z (or y) are assumed
- Define basis functions $\psi = [\psi_1, \psi_2, \cdots, \psi_M]^\top$ and restrict $\mathcal{K}$ to a finite-dimensional subspace of L^p ($\rightarrow K$) such that

$$\psi(\hat{z}_i) = K\psi(z_i)$$

- Minimize $\left\| K\psi(Z) - \psi(\hat{Z}) \right\|_F$:

$$K = \psi(\hat{Z})\psi(Z)^+ = (\underbrace{\psi(\hat{Z})\psi(Z)^\top}_{A})(\underbrace{\psi(Z)\psi(Z)^\top}_{G})^+ = AG^+$$

- If $\psi(z) = z$, we obtain DMD

[5] Williams, Kevrekidis & Rowley 2015.

Convergence of EDMD

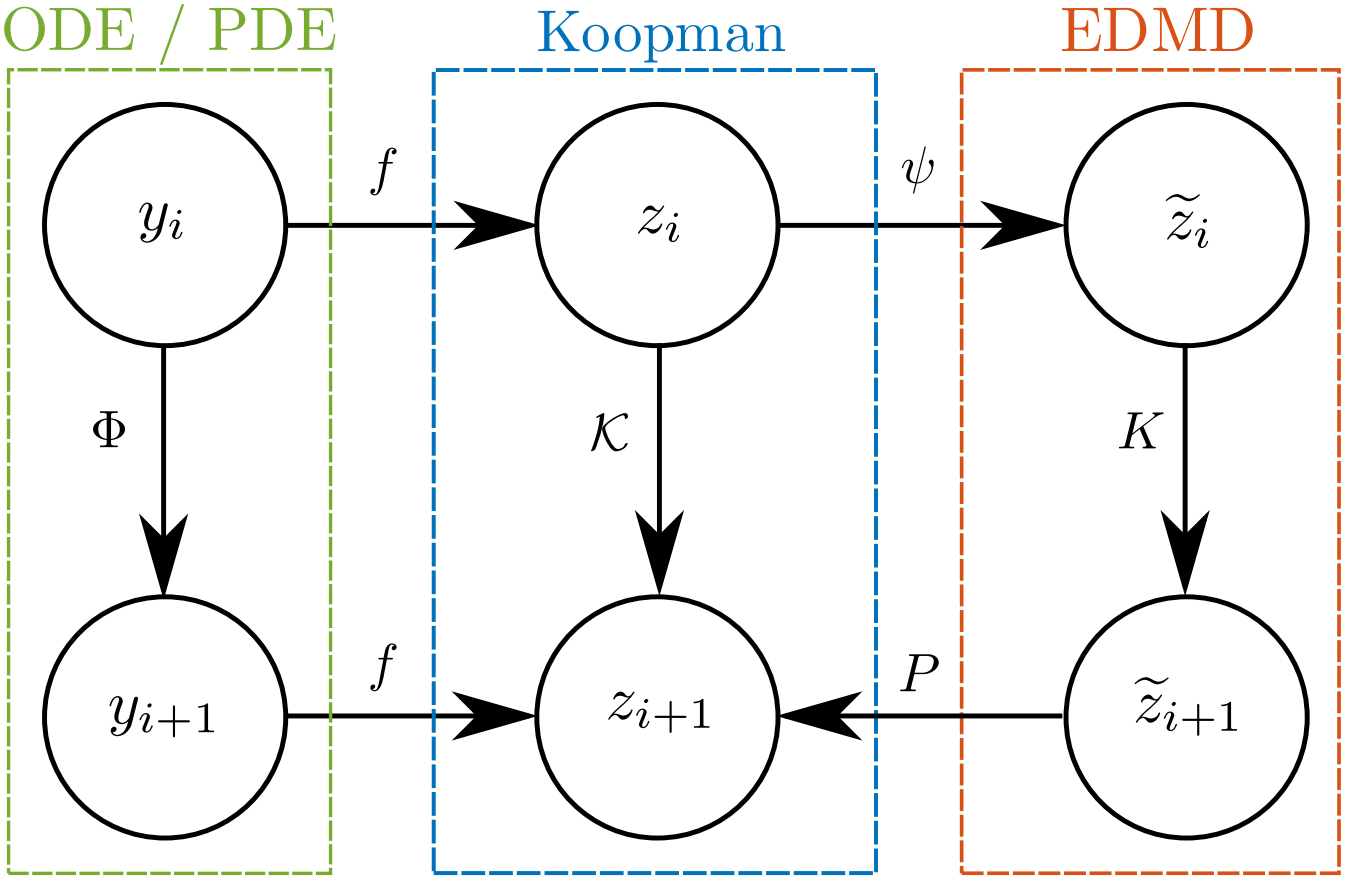
There exist several convergence results:

- In the infinite data limit, EDMD converges to the Galerkin projection of the Koopman operator.[6]
- The same is true for the Koopman generator[7]: $\mathcal{K}f = \lim_{t \to 0} \frac{f \circ \Phi^t - f}{t}$
- If the size of the (orthogonal) L^2 Galerkin basis tends to infinity as well, then EDMD yields convergence towards the Koopman operator in the strong operator topology.[8]
- First finite-data bounds.[9]

[6] Williams, Kevrekidis & Rowley, JNLS 2015; Klus, Koltai & Schütte, JCD 2016
[7] Klus, Nüske, P., Niemann, Clementi & Schütte, Physica D 2020
[8] Korda & Mezić, JNLS 2018
[9] Nüske, P., Philipp, Schaller & Worthmann, arXiv 2021; Zhang & Zuazua, HAL 2021

Reduced order model

Replace dynamics for y by dynamics for observations $z = f(y)$:



Control using the Koopman operator $\mathcal{K}^t$ & generator $\mathcal{K}$

- Augmented state $\bar{y} = (y, u)^\top$: [10]

$$[A, B] = \arg\min_{[\tilde{A},\tilde{B}]} \left\| \psi(\hat{Y}) - \tilde{A}\psi(Y) - \tilde{B}U \right\|_F$$

$$\Rightarrow \qquad \tilde{z}_{i+1} = A\tilde{z}_i + Bu_i$$

- Koopman invariant subspaces (f is an *eigenfunction* φ): [11]

$$\frac{d}{dt}\varphi(y,u) = \mathcal{K}\varphi(y,u) = \nabla\varphi(y,u)\cdot g(y,u)$$
$$= \lambda\varphi(y,u) + \nabla_u\varphi(y,u)\cdot\dot{u} = \underbrace{\lambda\varphi(y) + \nabla\varphi(y)\cdot Bu}_{\text{if } g(y,u)=h(y)+Bu}$$

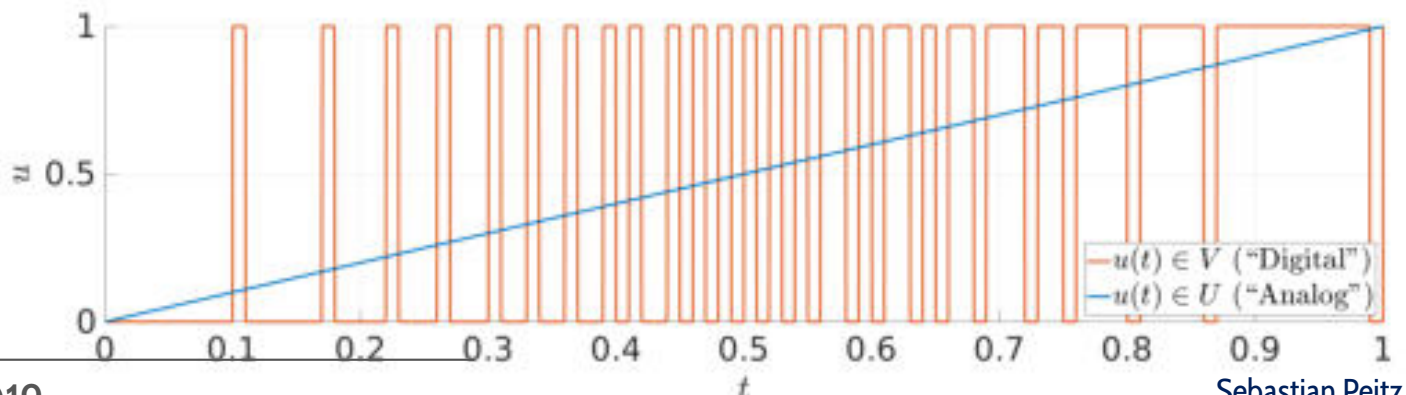
[10] Proctor et al. 2014, Proctor et al. 2016, Korda & Mezić 2018
[11] Brunton et al. 2016, Kaiser et al. 2021

Control using the Koopman operator $\mathcal{K}^t$ & generator $\mathcal{K}$

- Replace $g(y,u)$ by n_c autonomous systems ($V = \{u^1, \ldots, u^{n_c}\} \subset U$): [12]

$$\min_{u \in L^\infty([t_0,t_e],V)} J(y) = \int_{t_0}^{t_e} P(y(t))\,dt$$
$$\text{s.t.} \quad \dot{y}(t) = g_{u(t)}(y(t)) \quad \text{with} \quad g_{\bar{u}}(y(t)) = g(y(t),\bar{u})$$
$$y(t_0) = y_0.$$

"Analog-to-digital transformation" of the control problem:



[12] P. & Klus 2019.

Control using the Koopman operator $\mathcal{K}^t$ & generator $\mathcal{K}$

This yields n_c Koopman operators:

$$(\mathcal{K}^t_{u^j} f)(y) = f(\Phi_{u^j}(y)), \quad j = 1, \ldots, n_c$$

Approximation via EDMD: $\widetilde{z} = \psi(z) = \psi(f(y))$

$$\widetilde{z}_{k+1} = K_{u^j} \widetilde{z}_k, \quad j = 1, \ldots, n_c$$

- $\rightarrow$ linear increase in data requirements with $dim(u)$
- $\rightarrow$ Switching time optimization[13] for open loop optimal control
- $\rightarrow$ MPC for feedback control

Combination with convergence result for the Koopman operator

[13] Egerstedt et al. 2006, Flaßkamp et al. 2013, Stellato et al. 2016

Control using the Koopman operator $\mathcal{K}^t$ & generator $\mathcal{K}$

Full system:

$$\min_{u \in \{u^0, \dots, u^{n_c-1}\}^p} J = \sum_{i=1}^{p} L(y_i)$$
$$\text{s.t.} \quad y_{i+1} = \Phi_{u_i}(y_i)$$

Koopman:

$$\min_{u \in \{u^0, \dots, u^{n_c-1}\}^p} \widetilde{J} = \sum_{i=1}^{p} \widetilde{L}(\widetilde{z}_i)$$
$$\text{s.t.} \quad \widetilde{z}_{i+1} = K_{u_i} \widetilde{z}_i$$

Theorem (P. & Klus 2019)

If the assumptions for L^2 convergence of EDMD towards the Koopman operator are satisfied, then we have convergence in measure (w.r.t. z_0):

$$\lim_{M \to \infty} \lim_{N \to \infty} \mu\left(\left\{ z_0 \in \mathcal{Z} \middle| \sum_{i=0}^{p} |\widetilde{L}(\widetilde{z}_i) - L(y_i)| \geq \epsilon \right\} \right) = 0$$

Control using the Koopman operator $\mathcal{K}^t$ & generator $\mathcal{K}$

- Linear interpolation[14] to mitigate drawbacks of quantization, i.e.,
  - combinatorial optimization
  - less control flexibility

For control affine systems (i.e., $g(y,u) = g_1(y) + g_2(y)u$), we have

$$\mathcal{K}_{u^j} f = (g_1 + g_2 u^j) \cdot \nabla f = g_1 \cdot \nabla f + (g_2 u^j) \cdot \nabla f.$$

Introduce $\mathcal{B}_{u^j} = \mathcal{K}_{u^j} - \mathcal{K}_0 = \nabla f \cdot (g_2 u^j) \rightarrow$ control affinity is preserved:

$$\mathcal{K}_{\alpha_1 u^1 + \alpha_2 u^2} = \mathcal{K}_0 + \alpha_1 \mathcal{B}_{u^1} + \alpha_2 \mathcal{B}_{u^2}.$$

Control system is created using the standard basis $\{e_j\}_{j=1}^{n_c}$ for u:

$$\mathcal{K}_{u(t)} = \mathcal{K}_0 + \sum_{j=1}^{n_c} u_j(t) \mathcal{B}_{e_j}.$$

[14] P., Otto & Rowley 2019

Control using the Koopman operator $\mathcal{K}^t$ & generator $\mathcal{K}$

Constructing the matrices

$$\Psi = \begin{bmatrix} \psi(z^{(1)}) & \cdots & \psi(z^{(m)}) \\ u^{(1)} \otimes \psi(z^{(1)}) & \cdots & u^{(m)} \otimes \psi(z^{(m)}) \end{bmatrix}$$

$$\text{and} \quad \dot{\Psi} = \begin{bmatrix} \dot{\psi}(z^{(1)}) & \cdots & \dot{\psi}(z^{(m)}) \end{bmatrix},$$

from training data $\{(z^{(i)}, \dot{z}^{(i)}, u^{(i)})\}_{i=1}^{m}$, we find

$$\begin{bmatrix} K_O & B_{e_1} & \cdots & B_{e_{n_c}} \end{bmatrix} = \dot{\Psi}\Psi^{+}.$$

The data requirement grows again linearly with the dimension n_c of the input.

Outline

- Optimal control and surrogate modeling
- The Koopman operator for control
  - A few basics
  - EDMD and dictionary selection
  - Control using the Koopman operator & generator
  - Online learning
  - Some examples
- Conclusion & open questions

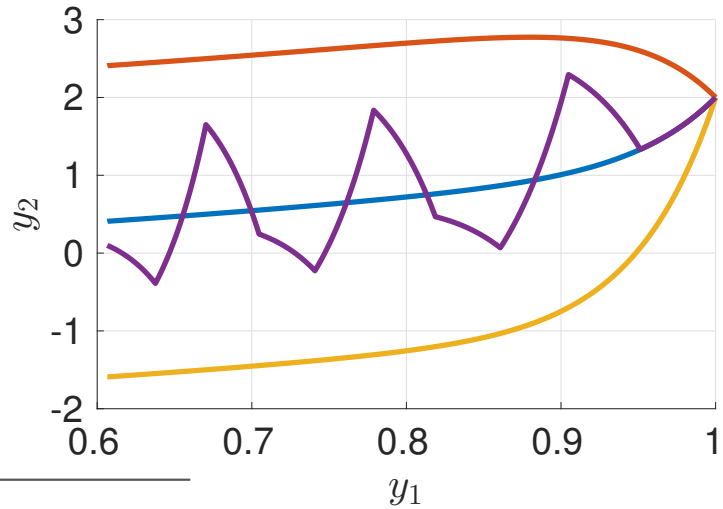
Example: 2D ODE[15]

$$\dot{y}(t) = g_{u^j}(y(t)) = \begin{pmatrix} \mu y_1(t) \\ \lambda(y_2(t) - (y_1(t))^2) \end{pmatrix} + \begin{pmatrix} 0 \\ u^j \end{pmatrix}, \quad j = 1, \ldots, n_c$$

$$y(0) = y_0$$

$n_c = 3$ systems: $u^0 = 0, u^1 = 2, u^2 = -2$

$z = f(y) = y; \psi(z) = \{y_1, y_2, y_1^2\}$



[15] Proctor et al. 2016

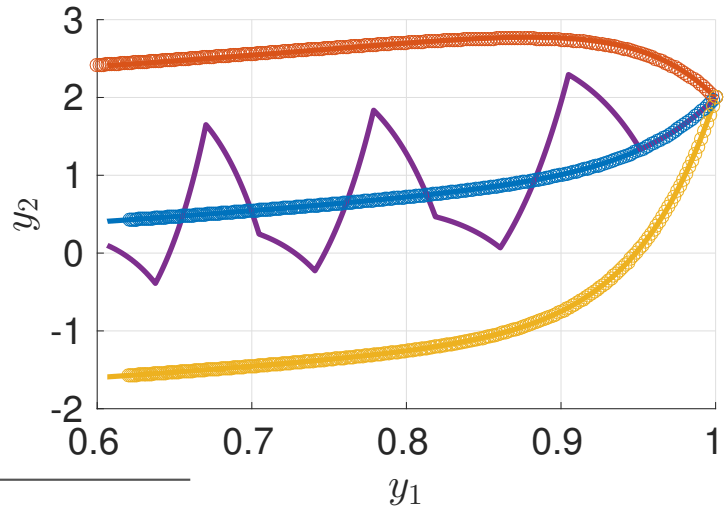
Example: 2D ODE[15]

$$\dot{y}(t) = g_{u^j}(y(t)) = \begin{pmatrix} \mu y_1(t) \\ \lambda(y_2(t) - (y_1(t))^2) \end{pmatrix} + \begin{pmatrix} 0 \\ u^j \end{pmatrix}, \quad j = 1, \ldots, n_c$$

$$y(0) = y_0$$

$n_c = 3$ systems: $u^0 = 0, u^1 = 2, u^2 = -2$

$z = f(y) = y; \psi(z) = \{y_1, y_2, y_1^2\}$



[15] Proctor et al. 2016

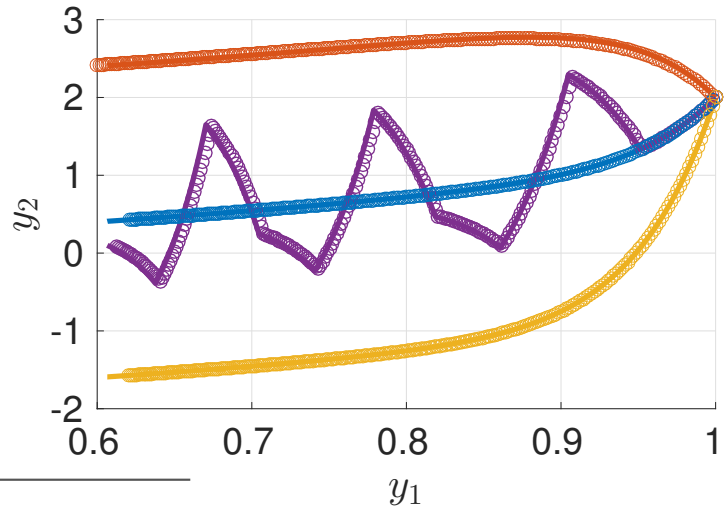
Example: 2D ODE[15]

$$\dot{y}(t) = g_{u^j}(y(t)) = \begin{pmatrix} \mu y_1(t) \\ \lambda(y_2(t) - (y_1(t))^2) \end{pmatrix} + \begin{pmatrix} 0 \\ u^j \end{pmatrix}, \quad j = 1, \ldots, n_c$$

$$y(0) = y_0$$

$n_c = 3$ systems: $u^0 = 0, u^1 = 2, u^2 = -2$

$z = f(y) = y; \psi(z) = \{y_1, y_2, y_1^2\}$



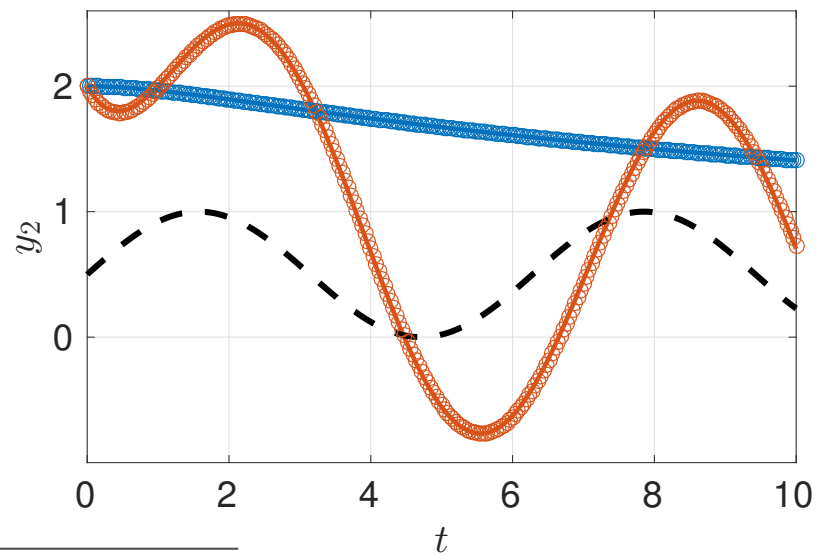
[15] Proctor et al. 2016

Example: 2D ODE[16]

$$\dot{y}(t) = g_{u^j}(y(t)) = \begin{pmatrix} \mu y_1(t) \\ \lambda(y_2(t) - (y_1(t))^2) \end{pmatrix} + \begin{pmatrix} 0 \\ u^j \end{pmatrix}, \quad j = 1, \ldots, 2$$

$$y(0) = y_0, \quad u^0 = 0, \quad u^1 = 1$$

$$u = 0.5(1 + \sin(t))$$

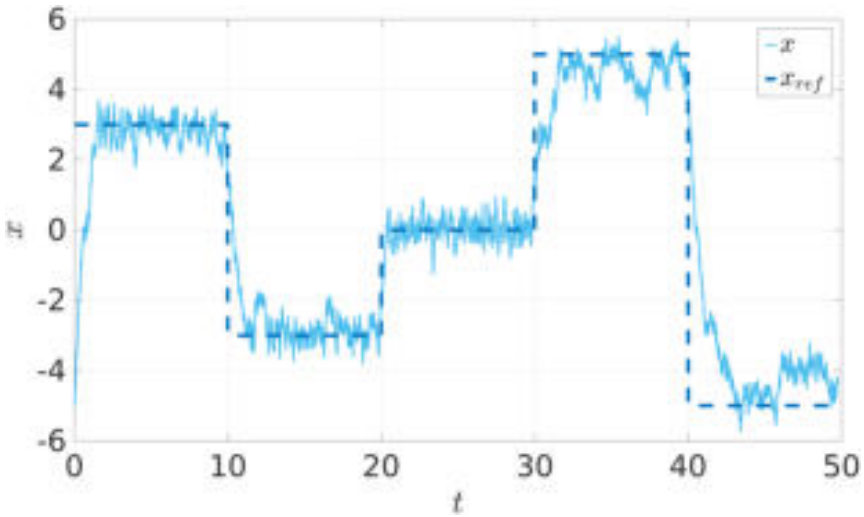


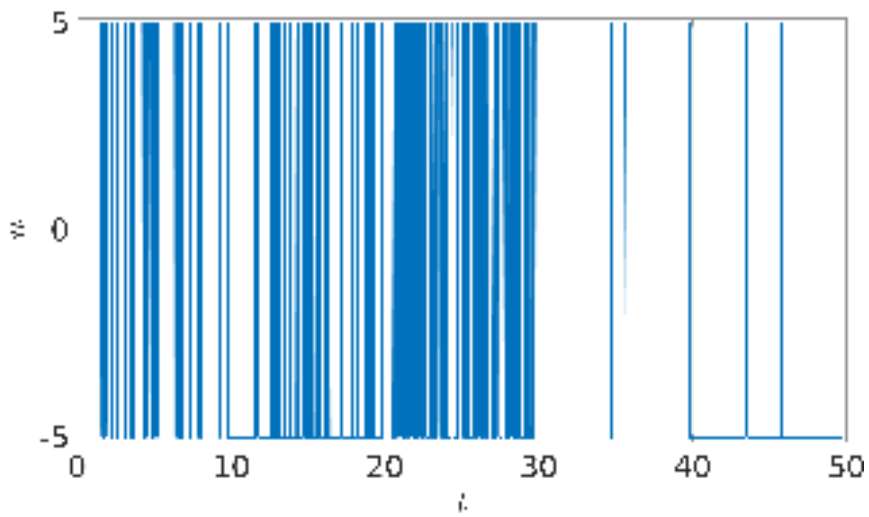
[16] Proctor et al. 2016

A simple stochastic system

Ornstein-Uhlenbeck process

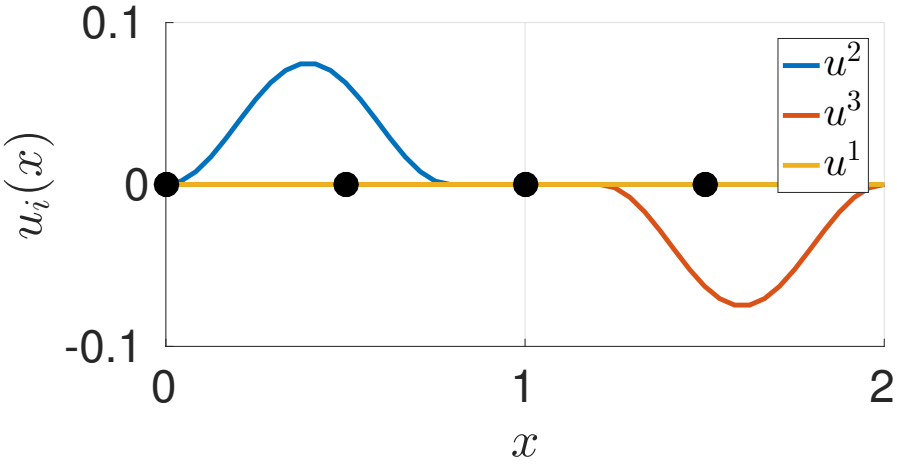
$$\mathrm{d}X_t = -\alpha(X_t - u)\mathrm{d}t + \sqrt{2\beta^{-1}}\mathrm{d}W_t,$$
$$(\mathcal{K}^t f)(x) = \mathbb{E}[f(\Phi^t(x))].$$





1D Burgers Equation

$$\dot{y}(x,t) - \nu \Delta y(x,t) + y \nabla y = u^j(x)$$
$$y(x,0) = y_0(x)$$

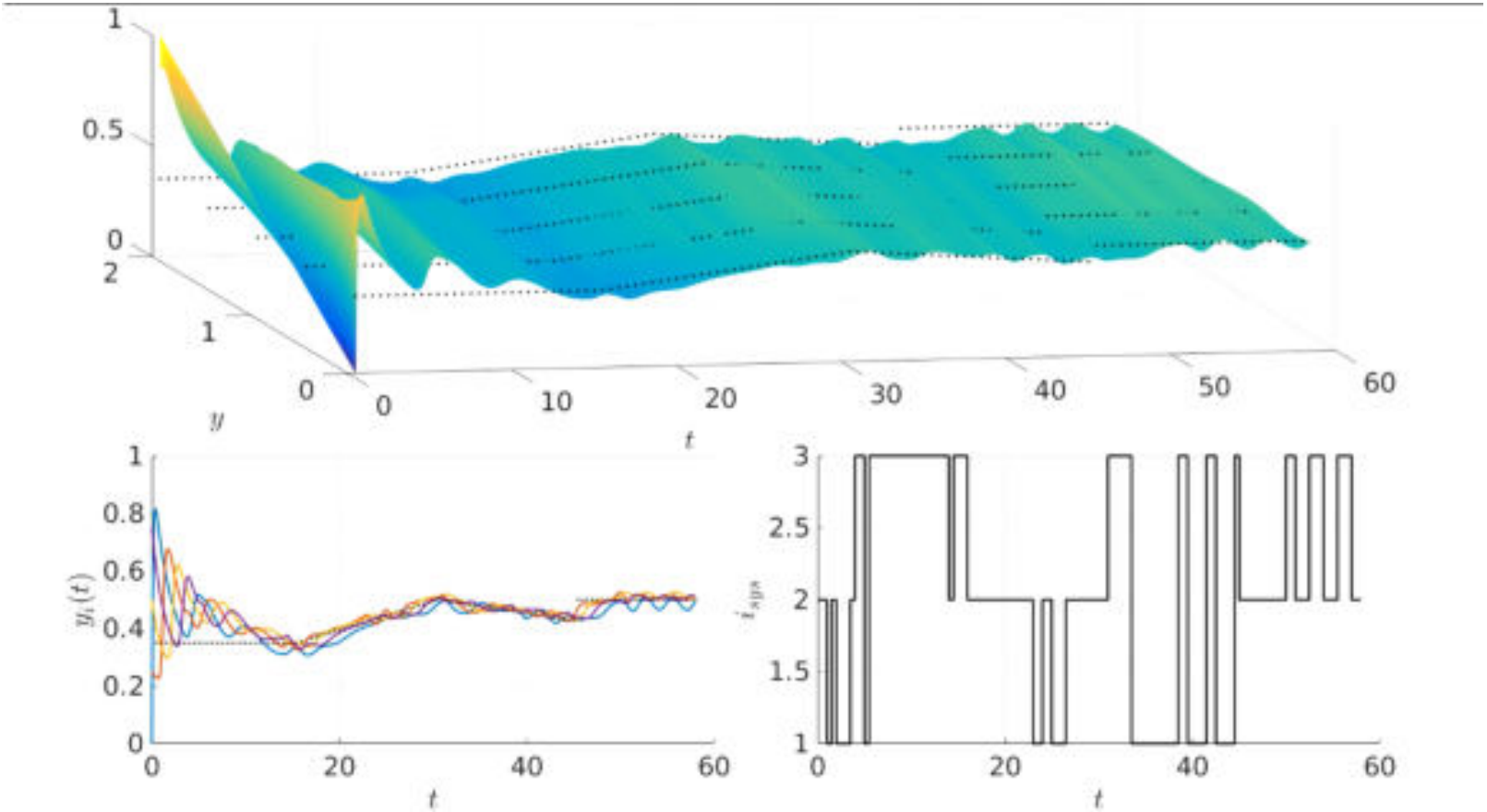


Observable: $f(y(x,t)) = (y(0,t), y(0.5,t), y(1,t), y(1.5,t))^\top$

Reduced model: $\psi(f(y_{i+1})) = K_{u^j} \psi(f(y_i))$

Objective: $\min_{u \in \{u^1,u^2,u^3\}^p} \sum_{i=1}^{p} \sum_{l=1}^{4} \| f_l(y_i) - f_{l,i}^{opt} \|_2^2$

1D Burgers Equation

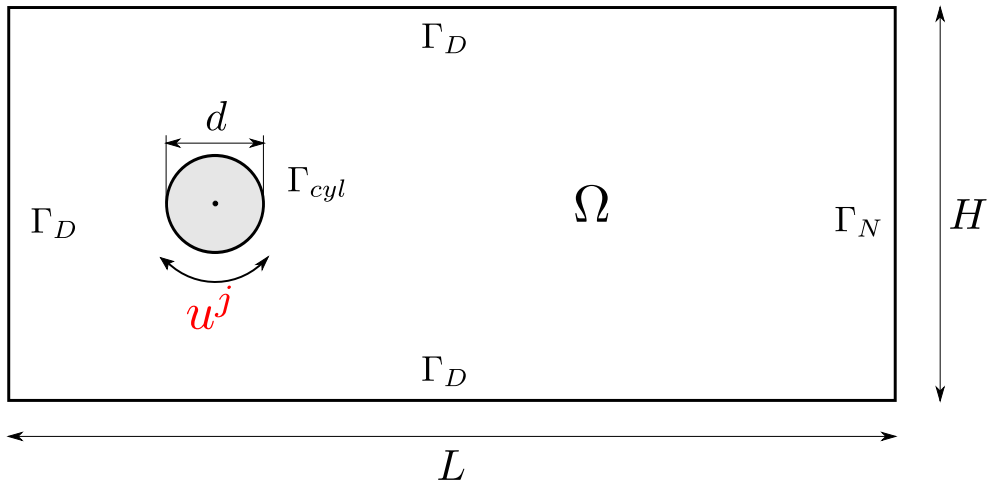


Navier-Stokes Equations

$$\dot{y}(x,t) + y(x,t) \cdot \nabla y(x,t) = \nabla p(x,t) + \frac{1}{Re} \Delta y(x,t)$$

$$\nabla \cdot y(x,t) = 0$$

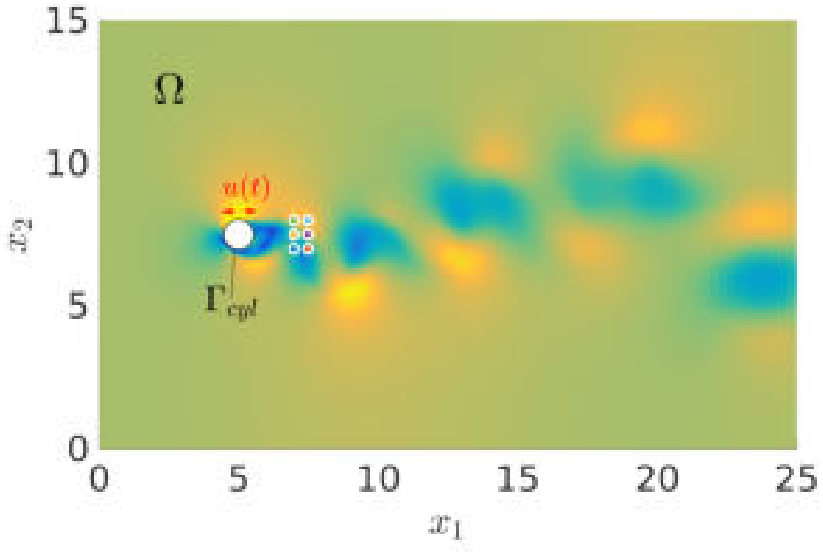
$$y(x,0) = y_0(x)$$



$u^0 = 0, u^1 = 2, u^2 = -2$

Navier-Stokes Equations

Observable f: lift and drag + vertical velocity at six positions in the wake

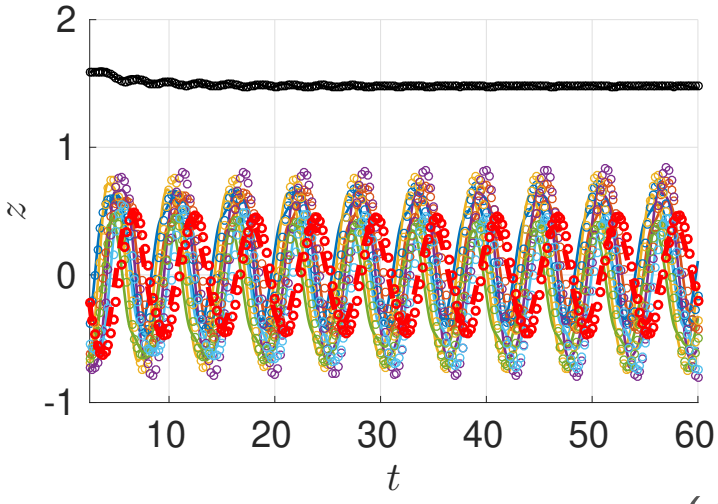


Reduced model: $\psi(f(y_{i+1})) = K_u \psi(f(y_i))$

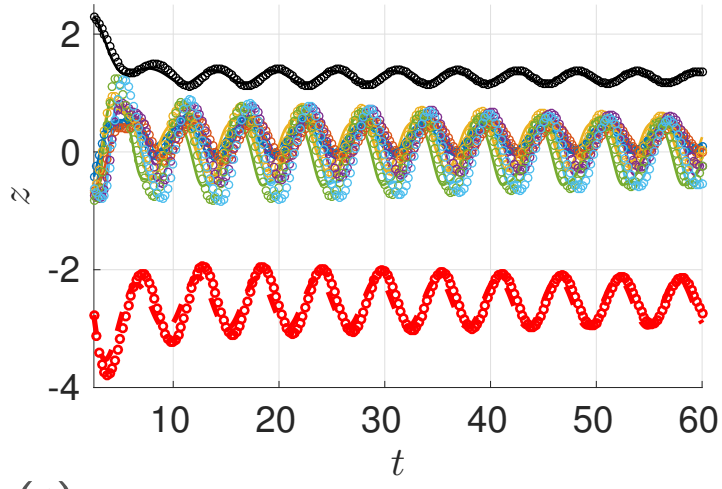
Objective: $\min_{u \in \{u^1, u^2, u^3\}^p} \sum_{i=1}^{p} \|f_1(y_i) - f_i^{opt}\|_2^2$

Navier-Stokes Equations

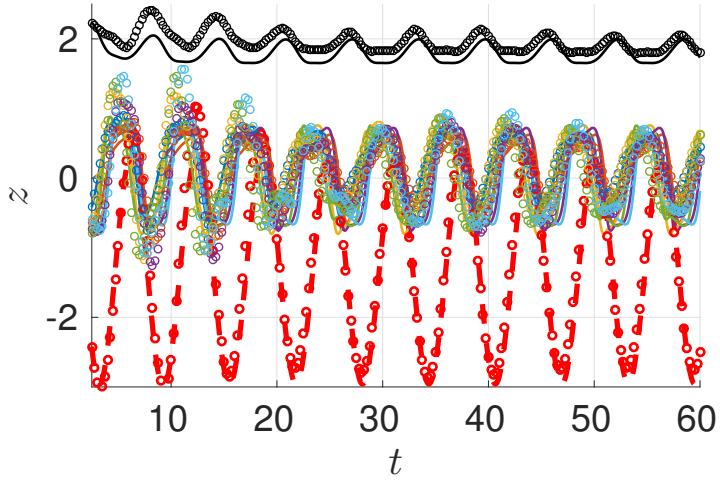
$u(t) = 0$



$u(t) = 2$

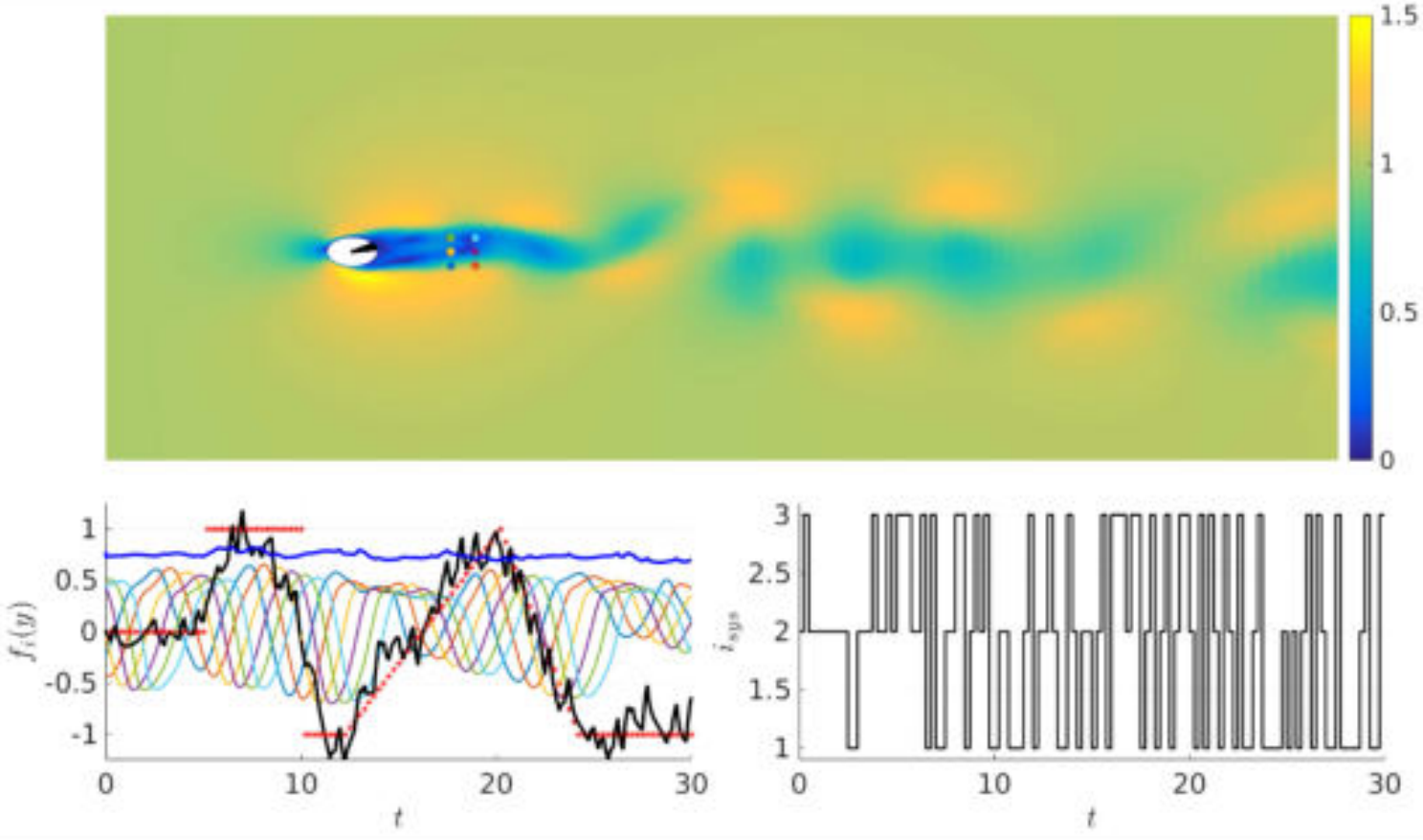


$u(t) = 1 + \sin(t)$



Acceleration:
$\approx 7.5 \cdot 10^4$

Navier-Stokes Equations



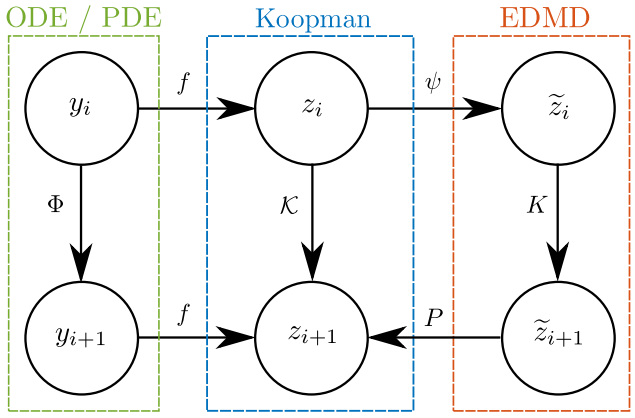
Outline

- Optimal control and surrogate modeling
- The Koopman operator for control
  - A few basics
  - EDMD and dictionary selection
  - Control using the Koopman operator & generator
  - Online learning
  - Some examples
- Conclusion & open questions

Conclusion

Koopman ROMs of control relevant dynamics

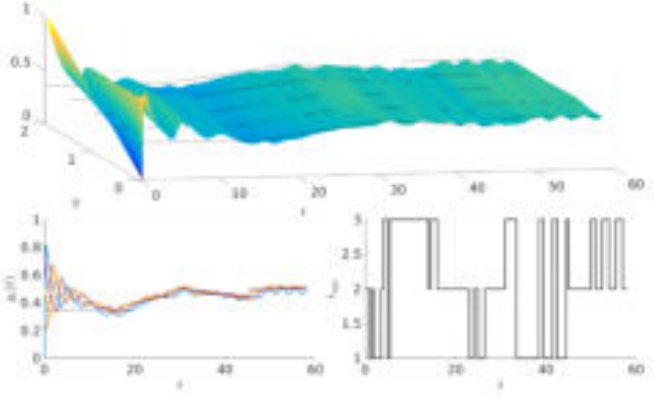
Optimality using convergence of EDMD

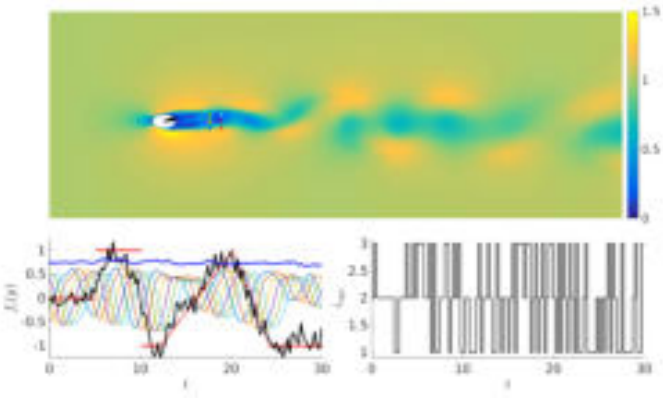


Data-efficient bilinear and switched systems

Feedback control via MPC

Speedup by several orders





Open questions

How to treat more complex systems – in particular turbulence:

- The choice of observables
- The choice of the dictionary
- Distributed control

Thank you for your attention