A data-driven flow model for wind-farm control based on Koopman mode decomposition of large-eddy simulations

Wim Munters, Johan Meyers

Department of Mechanical Engineering

KU Leuven, Leuven, Belgium

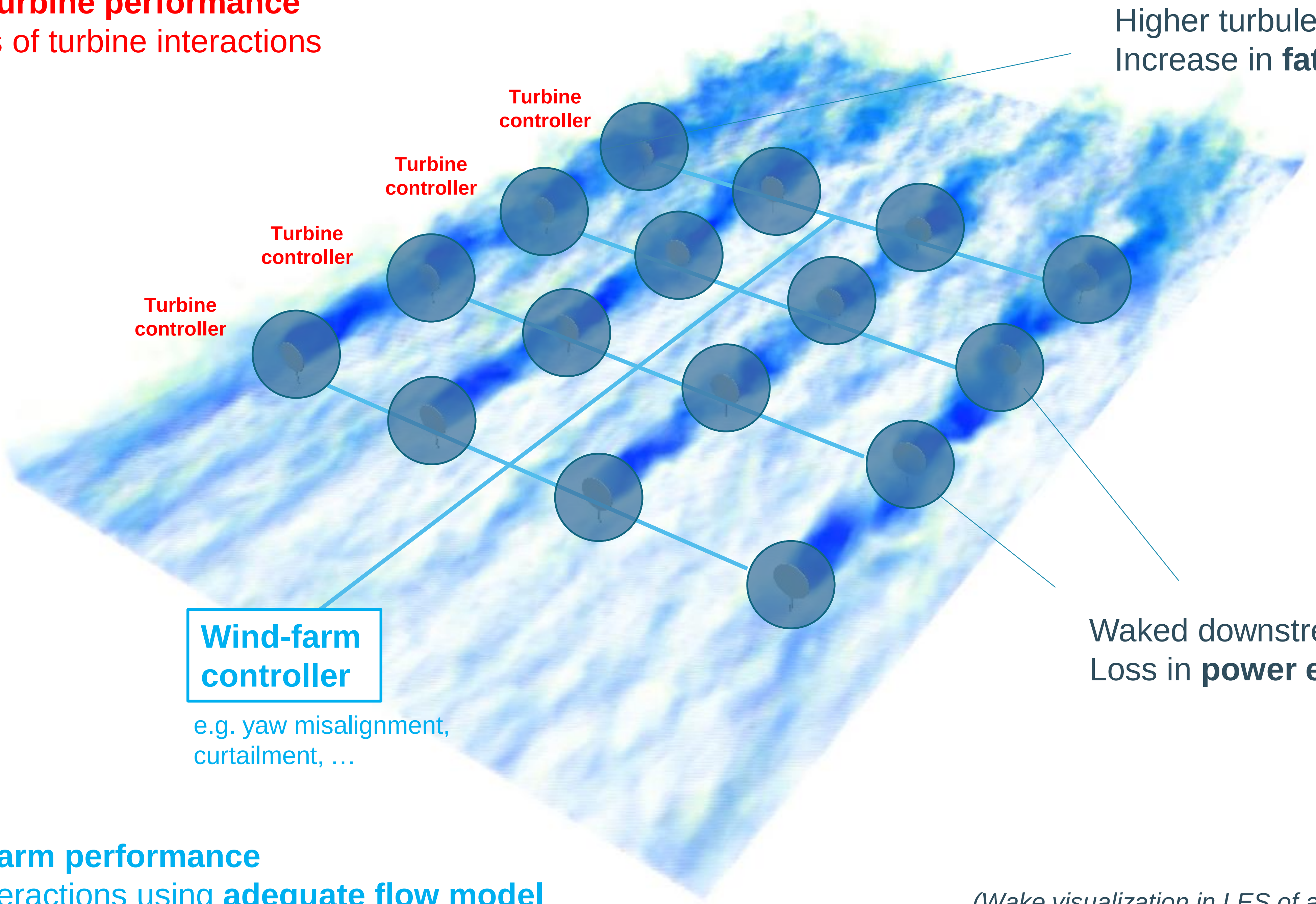
71st APS DFD Meeting, Atlanta, GA, USA

19/11/2018

Funded by the Horizon 2020 programme of the European Union

Optimize turbine performance
Regardless of turbine interactions

Higher turbulence levels
Increase in **fatigue loading**



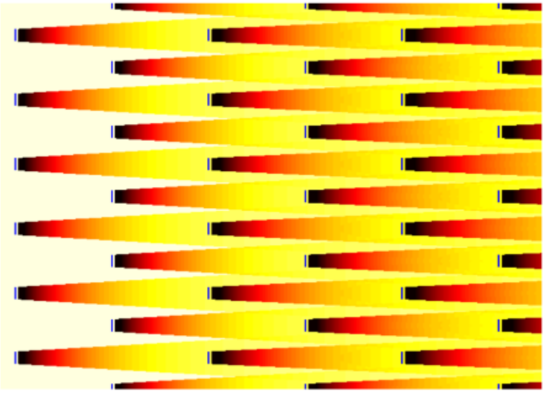
Waked downstream turbines
Loss in **power extraction**

Optimize farm performance
Mitigate interactions using **adequate flow model**

(Wake visualization in LES of a 4 x 4 wind farm)

Engineering models **Large Eddy Simulation models**

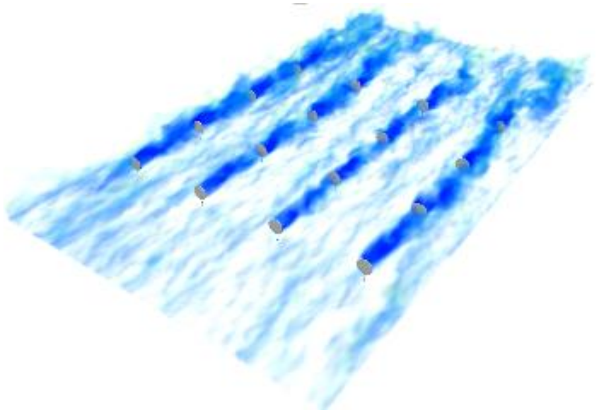
Cheap but do not capture non-linearities *Resolve non-linearities but very expensive*



Aim of current research

Data-driven linear flow model,

able to incorporate non-linear flow physics



Coupled Wake Boundary Layer Model
(Stevens, Gayme, Meneveau 2015)

Linear Wake Expansion (Jensen) Model
(Jensen 1984)

FLORIS Model
(Gebraad et al 2014)

…

SP-Wind (KU Leuven)

LESGO (JHU)

SOWFA (NREL)

…

Optimize farm performance

Mitigate interactions using **adequate flow model**

Linear approximation

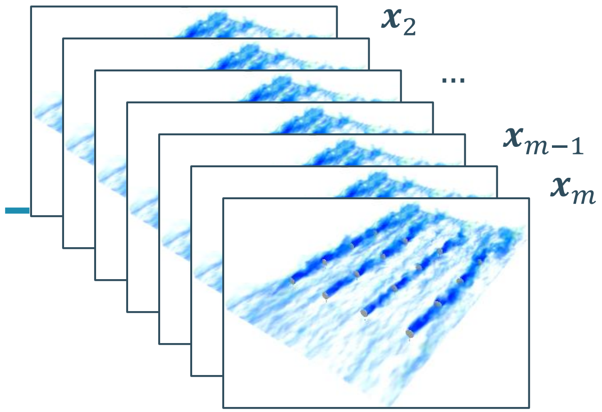
$x_{k+1} \approx Ax_k$

DMD (ADDREF

$g(x_{k+1}) = \mathcal{K} g(x_k)$

KMD (ADDREF

g



Non-linear dynamical system

$x_{k+1} = f(x_k)$

Wind-farm context:
$x = [v, p]$, f: time-discrete NS operator

x = system state

Koopman theory (1931)

Define a set of scalar observable functions $g: \mathbb{R}^n \mapsto \mathbb{R}$, spanning an infinite-dimensional Hilbert space $\mathcal{H}$.

→ There exists a linear infinite-dimensional Koopman operator $\mathcal{K}: \mathcal{H} \mapsto \mathcal{H}$, acting on the observables as

$$\mathcal{K} g(x_k) = g(x_{k+1})$$

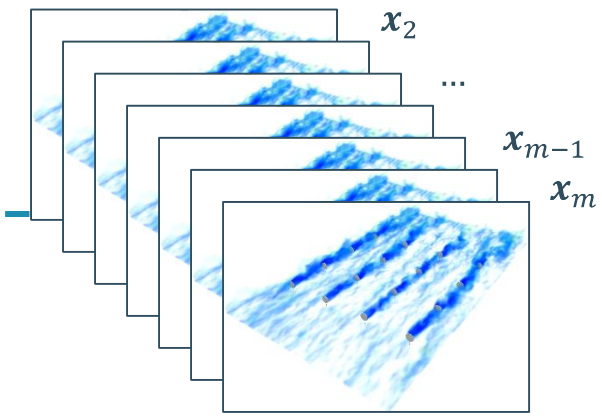
Linear approximation

$$x_{k+1} \approx Ax_k + Bu_k$$

Proctor, Brunton & Kutz,
SIAM 2018

$$g(x_{k+1}, u_{k+1}) = \mathcal{K} g(x_k, u_{k+1})$$

Proctor, Brunton & Kutz,
SIAM 2018



g

Non-linear dynamical system

$$x_{k+1} = f(x_k, u_k)$$

Wind-farm context:
$x = [v, p]$, f: time-discrete NS operator

x = system state

u = control input

Koopman theory (1931)

Define a set of scalar observable functions $g: \mathbb{R}^n \mapsto \mathbb{R}$, spanning an infinite-dimensional Hilbert space $\mathcal{H}$.

→ There exists a linear infinite-dimensional Koopman operator $\mathcal{K}: \mathcal{H} \mapsto \mathcal{H}$, acting on the observables as

$$\mathcal{K} g(x_k, u_k) = g(x_{k+1}, u_{k+1})$$

Non-linear system in original state $\boldsymbol{x}$

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_1 - x_1^2 \end{bmatrix}$$

However, virtually all non-linear systems require an infinite-dimensional embedding

→ Data-driven finite-dimensional approximation of Koopman operator for wind-farm flows

Koopman embedding

$$\boldsymbol{z} = \boldsymbol{g}(\boldsymbol{x}) = (x_1, x_2, x_1^2)$$

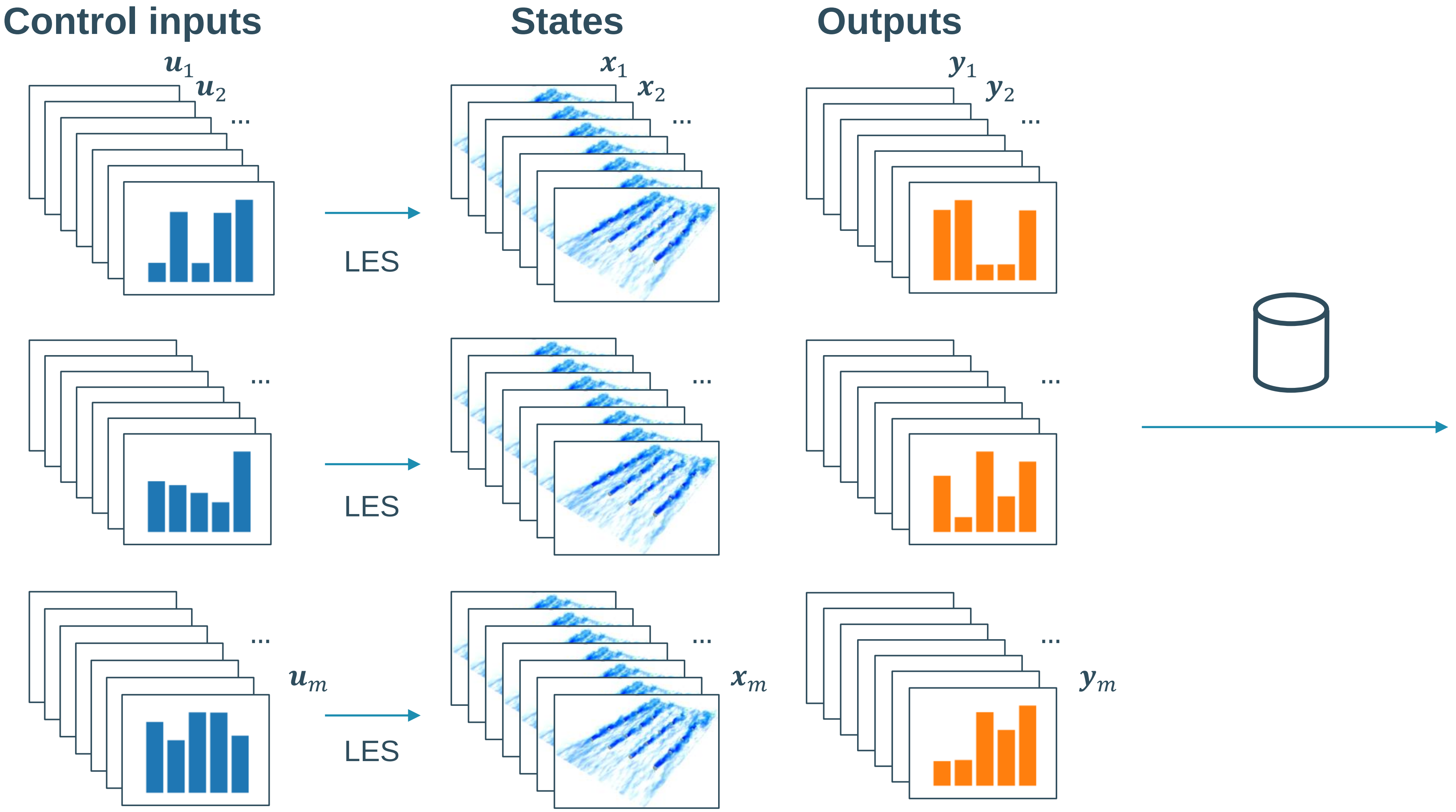
$$\dot{\boldsymbol{z}} = \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_1^2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_1 - x_1^2 \\ 2\dot{x}_1 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_1 - z_3 \\ 2z_1 \end{bmatrix} = K\boldsymbol{z}$$

…becomes a linear system in Koopman state $\boldsymbol{z}$

(example from Proctor, Brunton & Kutz, SIAM 2018)

KU LEUVEN

Approximating the Koopman operator



Snapshot matrices

Control $U = \begin{bmatrix} u_1 & u_2 & \cdots & u_m \end{bmatrix}$

State $X = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_{m-1} \\ | & | & & | \end{bmatrix}$

Image $X' = \begin{bmatrix} | & | & & | \\ x_2 & x_3 & \cdots & x_m \\ | & | & & | \end{bmatrix}$

Output $Y = \begin{bmatrix} y_1 & y_2 & \cdots & y_m \end{bmatrix}$

Approximating the Koopman operator

Dynamic mode decomposition

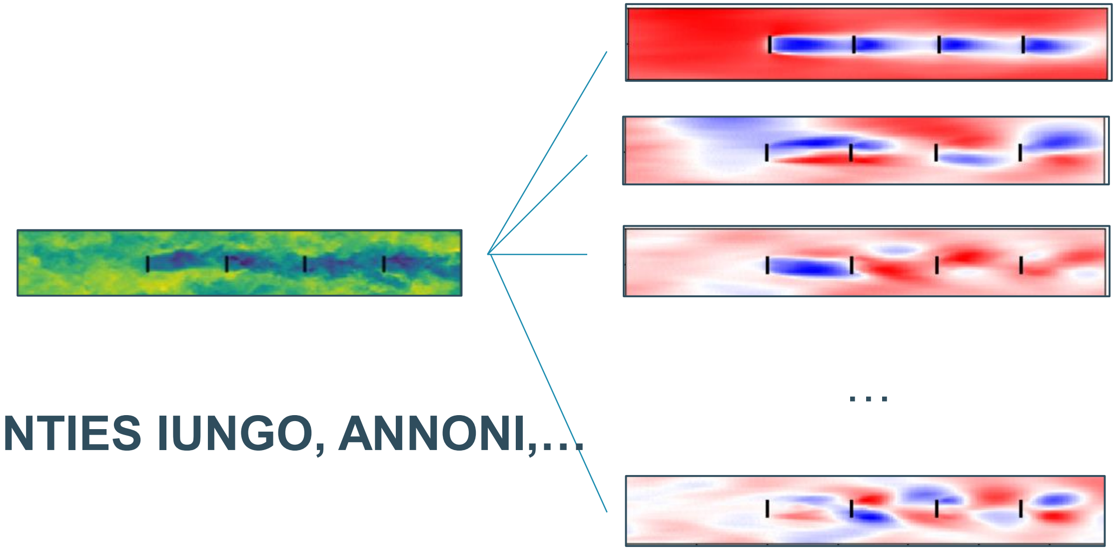
$$\begin{bmatrix} X' \\ Y \end{bmatrix} \approx \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X \\ U \end{bmatrix} \longrightarrow \min_{A,B,C,D} \left\| \begin{bmatrix} X' \\ Y \end{bmatrix} - \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X \\ U \end{bmatrix} \right\|_2 \longrightarrow \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} X' \\ Y \end{bmatrix} \begin{bmatrix} X \\ U \end{bmatrix}^+$$

$$\begin{aligned} \boldsymbol{x}_{k+1} &= A\boldsymbol{x}_k + B\boldsymbol{u}_k \\ \boldsymbol{y}_k &= C\boldsymbol{x}_k + D\boldsymbol{u}_k \end{aligned}$$

$$\boldsymbol{x} = P\tilde{\boldsymbol{x}}$$

$\boldsymbol{x} \in \mathbb{R}^n, \tilde{\boldsymbol{x}} \in \mathbb{R}^r$
$(r \ll n)$

$$\begin{aligned} \tilde{\boldsymbol{x}}_{k+1} &= \tilde{A}\,\tilde{\boldsymbol{x}}_k + \tilde{B}\,\boldsymbol{u}_k \\ \boldsymbol{y}_k &= \tilde{C}\,\tilde{\boldsymbol{x}}_k + \tilde{D}\,\boldsymbol{u}_k \end{aligned}$$



Typical basis:
r POD modes of X

REFERENTIES IUNGO, ANNONI,…

Snapshot matrices

Control $U = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_m \end{bmatrix}$

State $X = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_{m-1} \end{bmatrix}$

Image $X' = \begin{bmatrix} \boldsymbol{x}_2 & \boldsymbol{x}_3 & \cdots & \boldsymbol{x}_m \end{bmatrix}$

Output $Y = \begin{bmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_m \end{bmatrix}$

Approximating the Koopman operator

Dynamic mode decomposition

$$\begin{bmatrix} X' \\ Y \end{bmatrix} \approx \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X \\ U \end{bmatrix} \longrightarrow \min_{A,B,C,D} \left\| \begin{bmatrix} X' \\ Y \end{bmatrix} - \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X \\ U \end{bmatrix} \right\|_2 \longrightarrow \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} X' \\ Y \end{bmatrix} \begin{bmatrix} X \\ U \end{bmatrix}^+$$

$$\begin{aligned} \boldsymbol{x}_{k+1} &= A\boldsymbol{x}_k + B\boldsymbol{u}_k \\ \boldsymbol{y}_k &= C\boldsymbol{x}_k + D\boldsymbol{u}_k \end{aligned}$$

$$\boldsymbol{x} = P\widetilde{\boldsymbol{x}}$$

$\boldsymbol{x} \in \mathbb{R}^n, \widetilde{\boldsymbol{x}} \in \mathbb{R}^r$
$(r \ll n)$

$$\begin{aligned} \widetilde{\boldsymbol{x}}_{k+1} &= \tilde{A}\,\widetilde{\boldsymbol{x}}_k + \tilde{B}\,\boldsymbol{u}_k \\ \boldsymbol{y}_k &= \tilde{C}\,\widetilde{\boldsymbol{x}}_k + \tilde{D}\,\boldsymbol{u}_k \end{aligned}$$

Koopman: extended dynamic mode decomposition

1. Enrich snaps with historical data (delay embedding): $\Omega, X, X', Y \rightarrow \Omega_d, X_d, X'_d, Y_d$

$$X_d = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_{m-nd} \\ \boldsymbol{x}_2 & \boldsymbol{x}_3 & \cdots & \boldsymbol{x}_{m-nd+1} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{x}_{1+nd} & \boldsymbol{x}_{2+nd} & \cdots & \boldsymbol{x}_{m-1} \end{bmatrix}$$

Predict future state/output based on:
- Current state/control
- Previous states/controls

Snapshot matrices

Control $U = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_m \end{bmatrix}$

State $X = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_{m-1} \end{bmatrix}$

Image $X' = \begin{bmatrix} \boldsymbol{x}_2 & \boldsymbol{x}_3 & \cdots & \boldsymbol{x}_m \end{bmatrix}$

Output $Y = \begin{bmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_m \end{bmatrix}$

Approximating the Koopman operator

Dynamic mode decomposition

$$\begin{bmatrix} X' \\ Y \end{bmatrix} \approx \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X \\ U \end{bmatrix} \rightarrow \min_{A,B,C,D} \left\| \begin{bmatrix} X' \\ Y \end{bmatrix} - \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X \\ U \end{bmatrix} \right\|_2 \rightarrow \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} X' \\ Y \end{bmatrix} \begin{bmatrix} X \\ U \end{bmatrix}^+$$

$$\begin{aligned} \boldsymbol{x}_{k+1} &= A\boldsymbol{x}_k + B\boldsymbol{u}_k \\ \boldsymbol{y}_k &= C\boldsymbol{x}_k + D\boldsymbol{u}_k \end{aligned}$$

$$\boldsymbol{x} = P\tilde{\boldsymbol{x}}$$

$$\boldsymbol{x} \in \mathbb{R}^n, \tilde{\boldsymbol{x}} \in \mathbb{R}^r$$
$$(r \ll n)$$

$$\begin{aligned} \tilde{\boldsymbol{x}}_{k+1} &= \tilde{A}\tilde{\boldsymbol{x}}_k + \tilde{B}\boldsymbol{u}_k \\ \boldsymbol{y}_k &= \tilde{C}\tilde{\boldsymbol{x}}_k + \tilde{D}\boldsymbol{u}_k \end{aligned}$$

Koopman: extended dynamic mode decomposition

1. Enrich snaps with historical data (delay embedding): $\Omega, X, X', Y \rightarrow \Omega_d, X_d, X'_d, Y_d$
2. Lift delay-embedded snapshot matrices into Koopman snapshot matrices Z, Z'

$$Z = \begin{bmatrix} \boldsymbol{g}(\boldsymbol{x}_{d1}, \boldsymbol{u}_{d1}) & \boldsymbol{g}(\boldsymbol{x}_{d2}, \boldsymbol{u}_{d2}) & \dots & \boldsymbol{g}(\boldsymbol{x}_{dm-1}, \boldsymbol{u}_{dm-1}) \end{bmatrix}$$

$$Z' = \begin{bmatrix} \boldsymbol{g}(\boldsymbol{x}_{d2}, \boldsymbol{u}_{d2}) & \boldsymbol{g}(\boldsymbol{x}_{d3}, \boldsymbol{u}_{d3}) & \dots & \boldsymbol{g}(\boldsymbol{x}_{dm}, \boldsymbol{u}_{dm}) \end{bmatrix}$$

Snapshot matrices

Control $U = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_m \end{bmatrix}$

State $X = \begin{bmatrix} | & | & & | \\ \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_{m-1} \\ | & | & & | \end{bmatrix}$

Image $X' = \begin{bmatrix} | & | & & | \\ \boldsymbol{x}_2 & \boldsymbol{x}_3 & \cdots & \boldsymbol{x}_m \\ | & | & & | \end{bmatrix}$

Output $Y = \begin{bmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_m \end{bmatrix}$

Approximating the Koopman operator

Dynamic mode decomposition

$$\begin{bmatrix} X' \\ Y \end{bmatrix} \approx \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X \\ U \end{bmatrix} \longrightarrow \min_{A,B,C,D} \left\| \begin{bmatrix} X' \\ Y \end{bmatrix} - \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X \\ U \end{bmatrix} \right\|_2 \longrightarrow \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} X' \\ Y \end{bmatrix} \begin{bmatrix} X \\ U \end{bmatrix}^+$$

$$\begin{aligned} \boldsymbol{x}_{k+1} &= A\boldsymbol{x}_k + B\boldsymbol{u}_k \\ \boldsymbol{y}_k &= C\boldsymbol{x}_k + D\boldsymbol{u}_k \end{aligned} \qquad \boldsymbol{x} = P\tilde{\boldsymbol{x}} \qquad \begin{aligned} \tilde{\boldsymbol{x}}_{k+1} &= \tilde{A}\tilde{\boldsymbol{x}}_k + \tilde{B}\boldsymbol{u}_k \\ \boldsymbol{y}_k &= \tilde{C}\tilde{\boldsymbol{x}}_k + \tilde{D}\boldsymbol{u}_k \end{aligned}$$

$\boldsymbol{x} \in \mathbb{R}^n, \tilde{\boldsymbol{x}} \in \mathbb{R}^r$
$(r \ll n)$

Koopman: extended dynamic mode decomposition

1. Enrich snaps with historical data (delay embedding): $\Omega, X, X', Y \rightarrow \Omega_d, X_d, X'_d, Y_d$
2. Lift delay-embedded snapshot matrices into Koopman snapshot matrices Z, Z'
3. Fit mapping $Z \mapsto Z'$: finite-dimensional Koopman approximation K

$$K = Z'Z^+$$

$$\boldsymbol{z}_{k+1} = K\boldsymbol{z}_k \qquad \boldsymbol{z} = P\tilde{\boldsymbol{z}} \qquad \tilde{\boldsymbol{z}}_{k+1} = K\tilde{\boldsymbol{z}}_k$$

Snapshot matrices

Control $U = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_m \end{bmatrix}$

State $X = \begin{bmatrix} | & | & & | \\ \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_{m-1} \\ | & | & & | \end{bmatrix}$

Image $X' = \begin{bmatrix} | & | & & | \\ \boldsymbol{x}_2 & \boldsymbol{x}_3 & \cdots & \boldsymbol{x}_m \\ | & | & & | \end{bmatrix}$

Output $Y = \begin{bmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_m \end{bmatrix}$

KU LEUVEN

Approximating the Koopman operator

- Standard Koopman: advance observables of delay-embedded state and control

$$\boldsymbol{z}_{k+1} = K\boldsymbol{z}_k$$

$$\leftrightarrow \; \boldsymbol{g}(\boldsymbol{x}_{d,k+1}, \boldsymbol{u}_{d,k+1}) = K\boldsymbol{g}(\boldsymbol{x}_{d,k}, \boldsymbol{u}_{d,k})$$

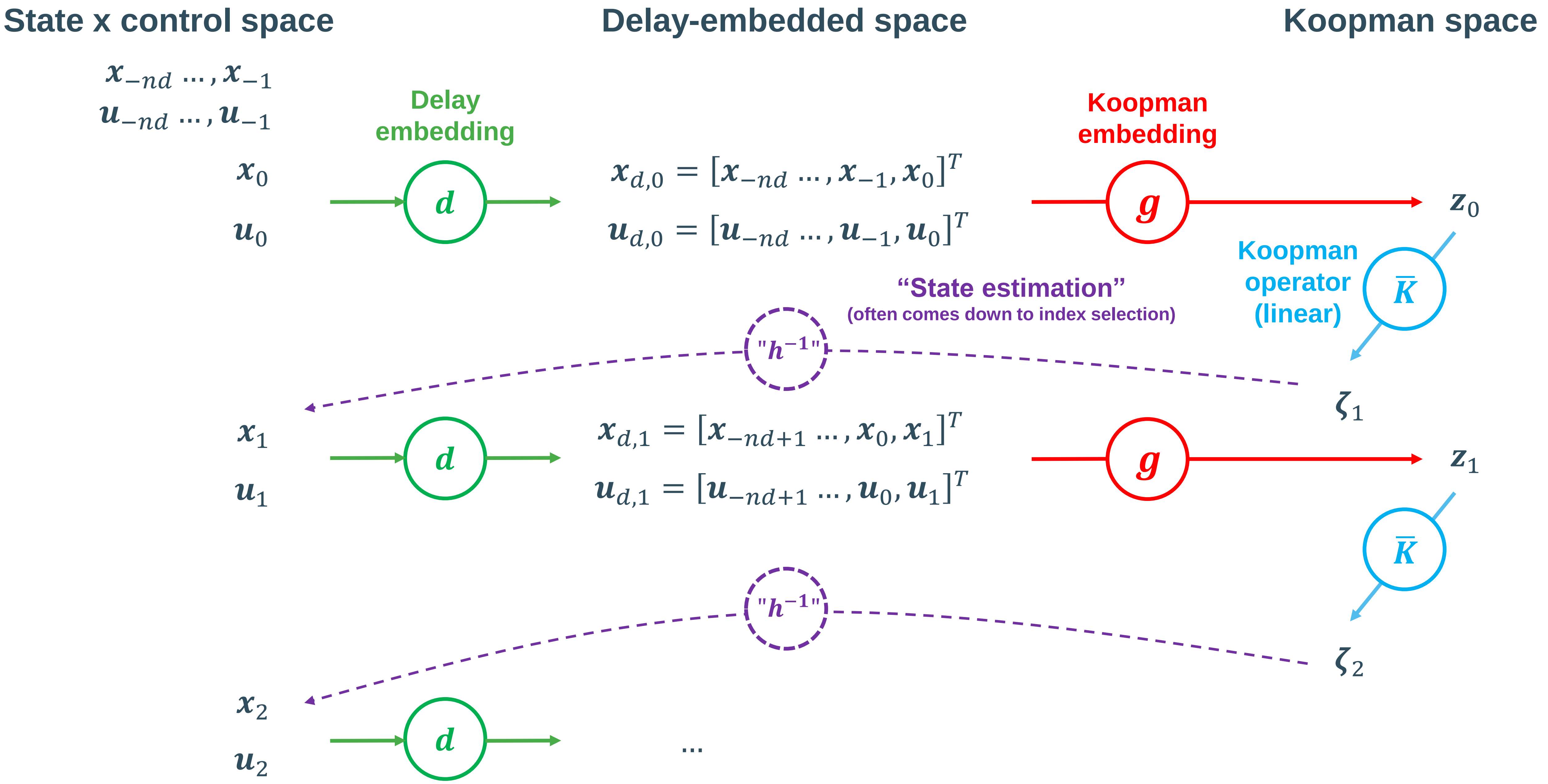
- In reality, we only care about latest state update $\boldsymbol{x}_{k+1}$

→ 'Modified' Koopman: subspace projection, only predict observables of $\boldsymbol{x}_{k+1}$

$$\boldsymbol{h}(\boldsymbol{x}_d) = \bar{K}\boldsymbol{g}(\boldsymbol{x}_{d,k}, \boldsymbol{u}_{d,k}) \qquad (\text{span } \boldsymbol{h} \subset \text{span } \boldsymbol{g})$$

$$\boldsymbol{\zeta}_{k+1} = \bar{K}\boldsymbol{z}_k$$

(in practice: eliminating rows of original least-squares fit $K = Z'Z^{+}$)

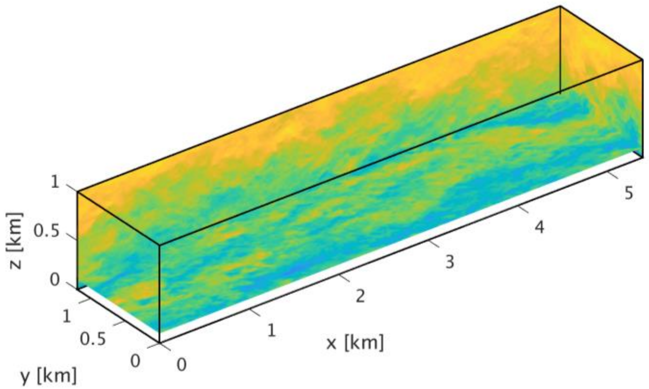

State x control space
Delay-embedded space
Koopman space
$x_{-nd} \dots, x_{-1}$
$u_{-nd} \dots, u_{-1}$
x_0
u_0
Delay embedding
d
$x_{d,0} = [x_{-nd} \dots, x_{-1}, x_0]^T$
$u_{d,0} = [u_{-nd} \dots, u_{-1}, u_0]^T$
Koopman embedding
g
z_0
Koopman operator (linear)
$\bar{K}$
"State estimation"
(often comes down to index selection)
"h^{-1}"
ζ_1
x_1
u_1
d
$x_{d,1} = [x_{-nd+1} \dots, x_0, x_1]^T$
$u_{d,1} = [u_{-nd+1} \dots, u_0, u_1]^T$
g
z_1
$\bar{K}$
"h^{-1}"
ζ_2
x_2
u_2
d
…

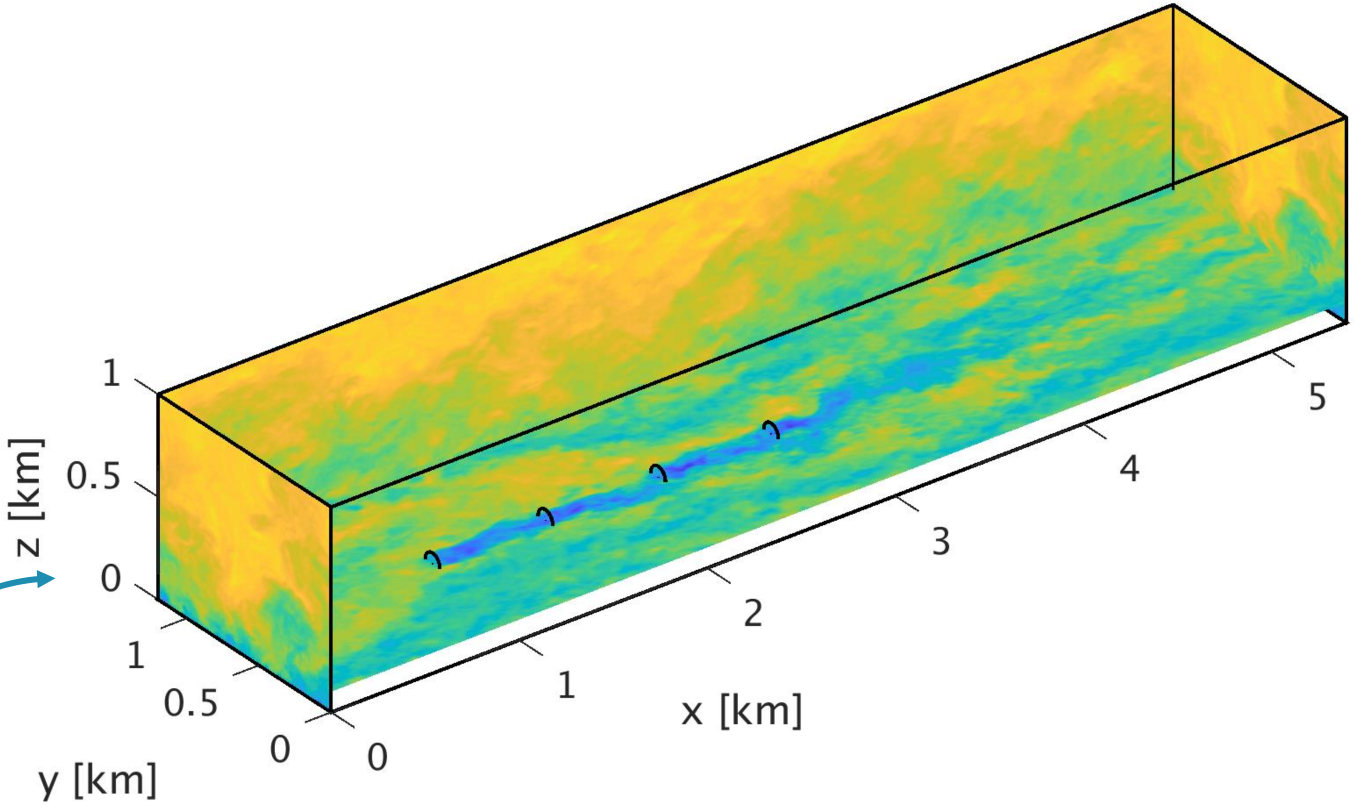
Demonstration case: setup

Wind Farm
4 aligned wind turbines
(actuator disk models)

Discretization
$L_x \times L_y \times L_z = 5.4 \times 1.2 \times 1 \text{ km}^3$
$\Delta_x \times \Delta_y \times \Delta_z = 18 \times 12 \times 6 \text{ m}^3$ (5M)

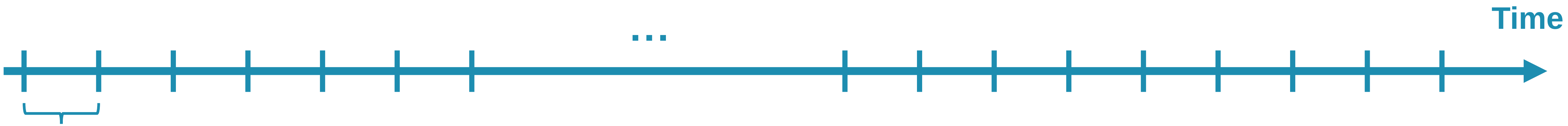
Turbulent inflow
from periodic precursor
domain (TI ≈ 10% at hub)





Computational cost
1 hr “wind-farm time” = 1 hr walltime on 224 Intel Broadwell cores

Demonstration case: snapshots

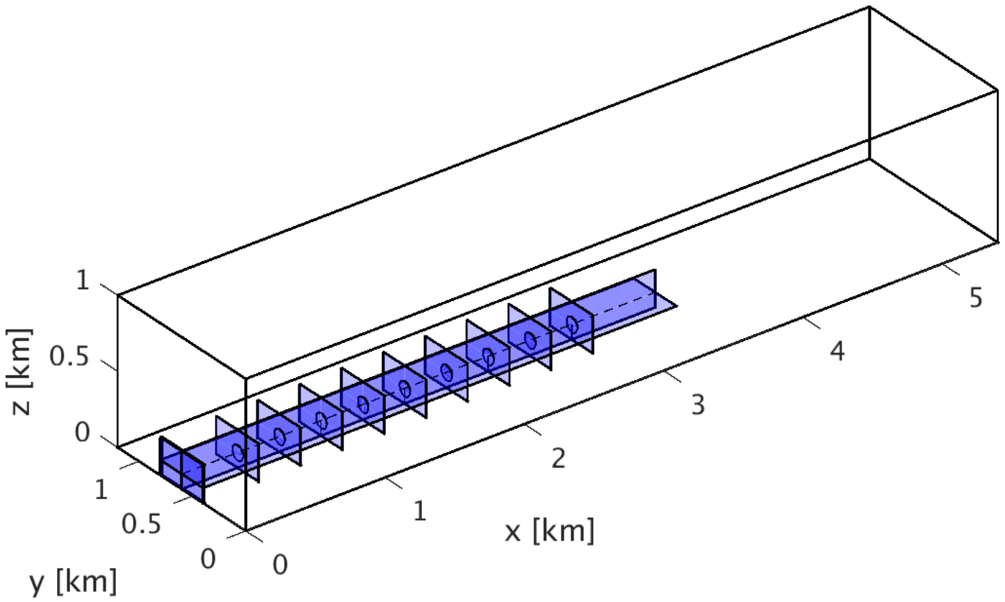


Time window

- $T = 900$ s
- $T_{samp} = 1$ s

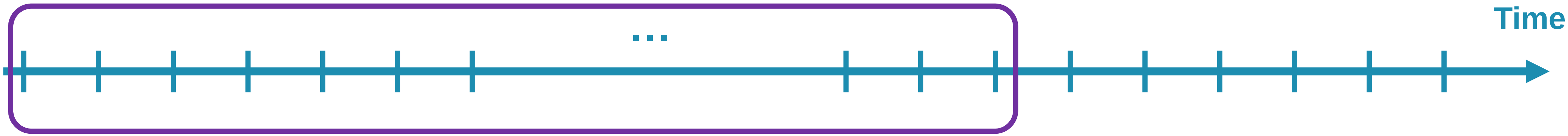
State snapshots:

- Disk-averaged velocities (scalar)
- Cross-sections (yz) x 10
- Planview (xy)
- Hub slice (xz)



Control snapshots: Turbine thrust coefficient
Output snapshots: Turbine power production

Demonstration case: snapshots

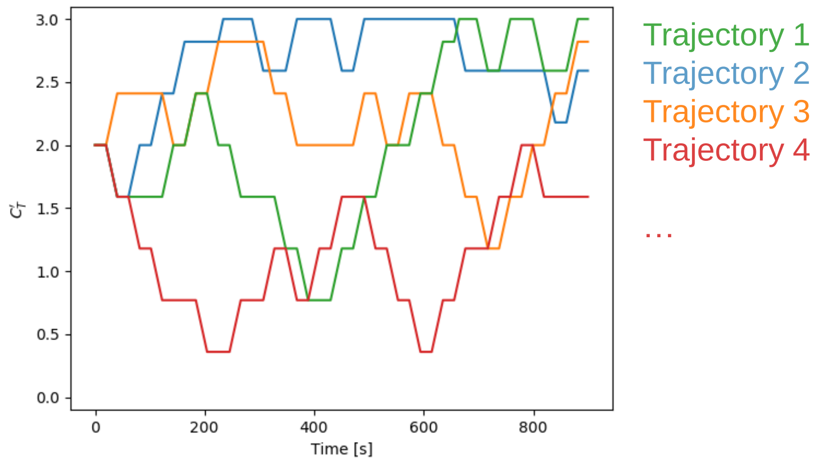


Time window

- T = 900 s
- T_{samp} = 1 s

Model training set

- 40 x steady turbines ($C_T' = 2$)
- 15 x turbines off ($C_T' = 0$)
- 30 x unsteady first turbine ($0 \leq C_T' \leq 3$)



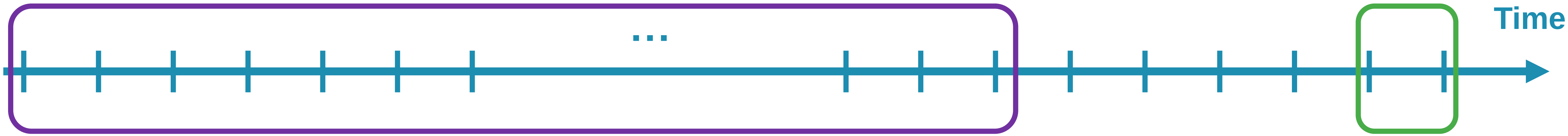
State snapshots:

- Disk-averaged velocities (scalar)
- Cross-sections (yz) x 10
- Planview (xy)
- Hub slice (xz)

Control snapshots: Turbine thrust coefficient
Output snapshots: Turbine power production

Demonstration case: snapshots

Unseen controls
Unseen flow fields
Unseen inflow conditions



Time window

- T = 900 s
- T_{samp} = 1 s

State snapshots:

- Disk-averaged velocities (scalar)
- Cross-sections (yz) x 10
- Planview (xy)
- Hub slice (xz)

Control snapshots: Turbine thrust coefficient
Output snapshots: Turbine power production

Model training set

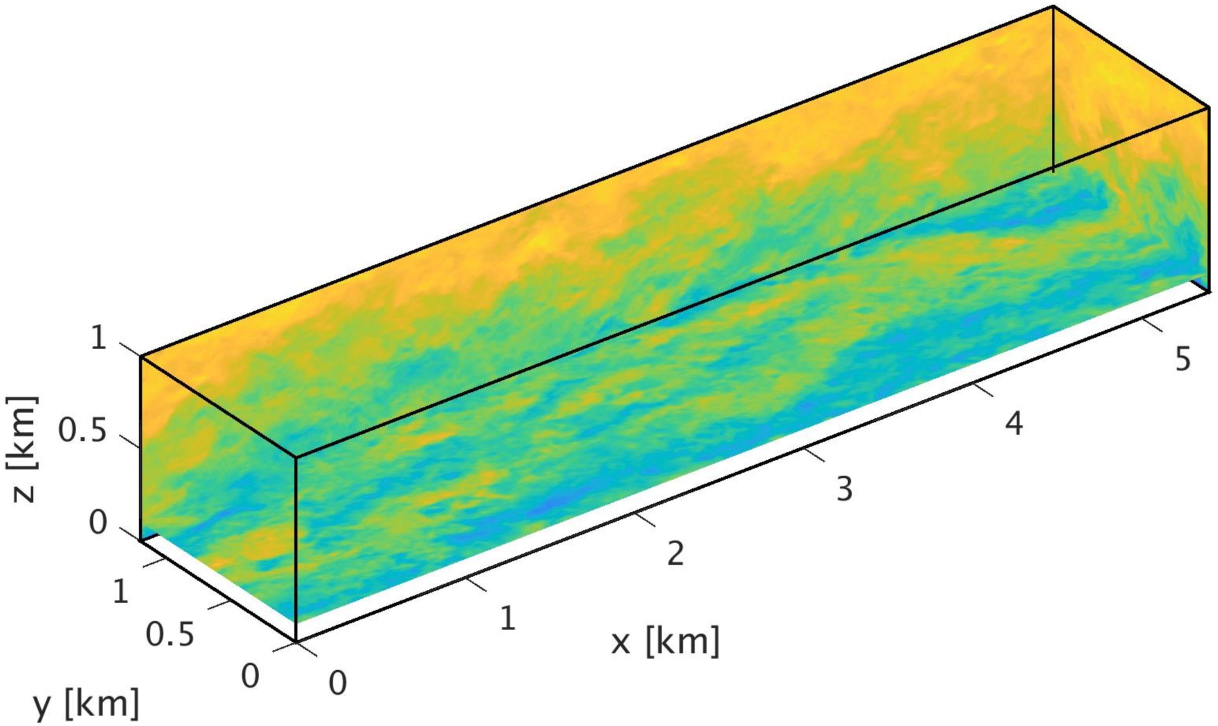
- 40 x steady turbines ($C_T' = 2$)
- 15 x turbines off ($C_T' = 0$)
- 30 x unsteady first turbine ($0 \leq C_T' \leq 3$)

Model testing set

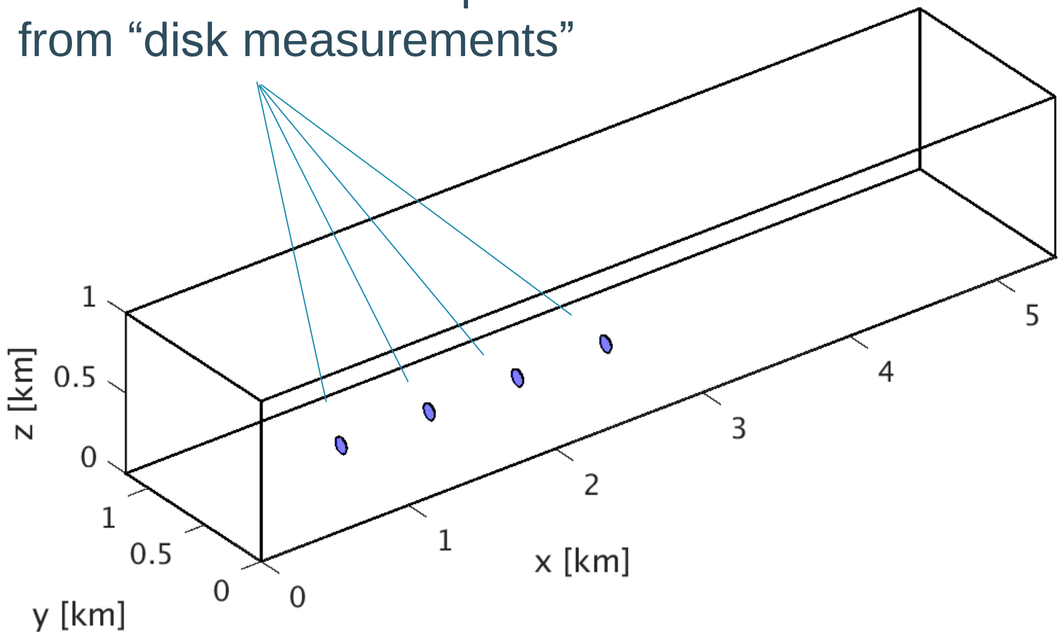
- 1 x steady turbine ($C_T' = 2$)
- 1 x turbines off ($C_T' = 0$)
- 1 x unsteady first turbine ($0 \leq C_T' \leq 3$)

Linear model: no turbines

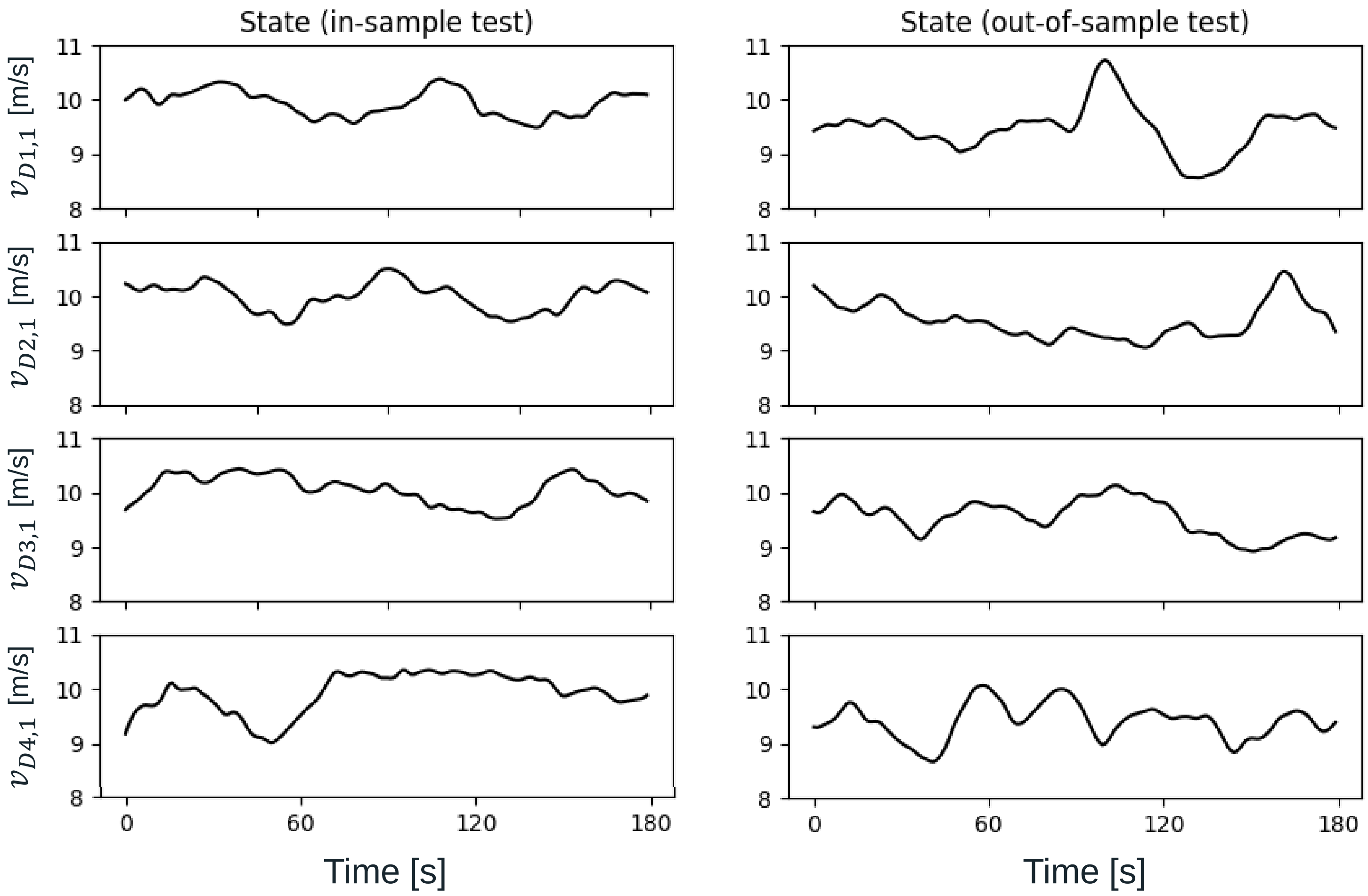
- No turbines: find linear model for flow propagation (~ Taylor’s hypothesis)

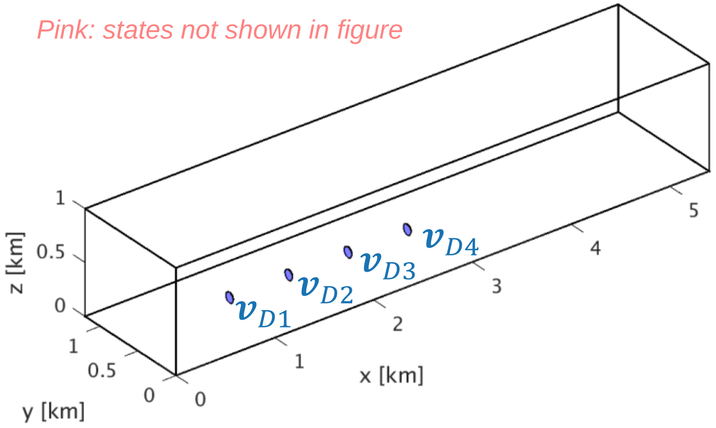


Assess linear model performance
from “disk measurements”



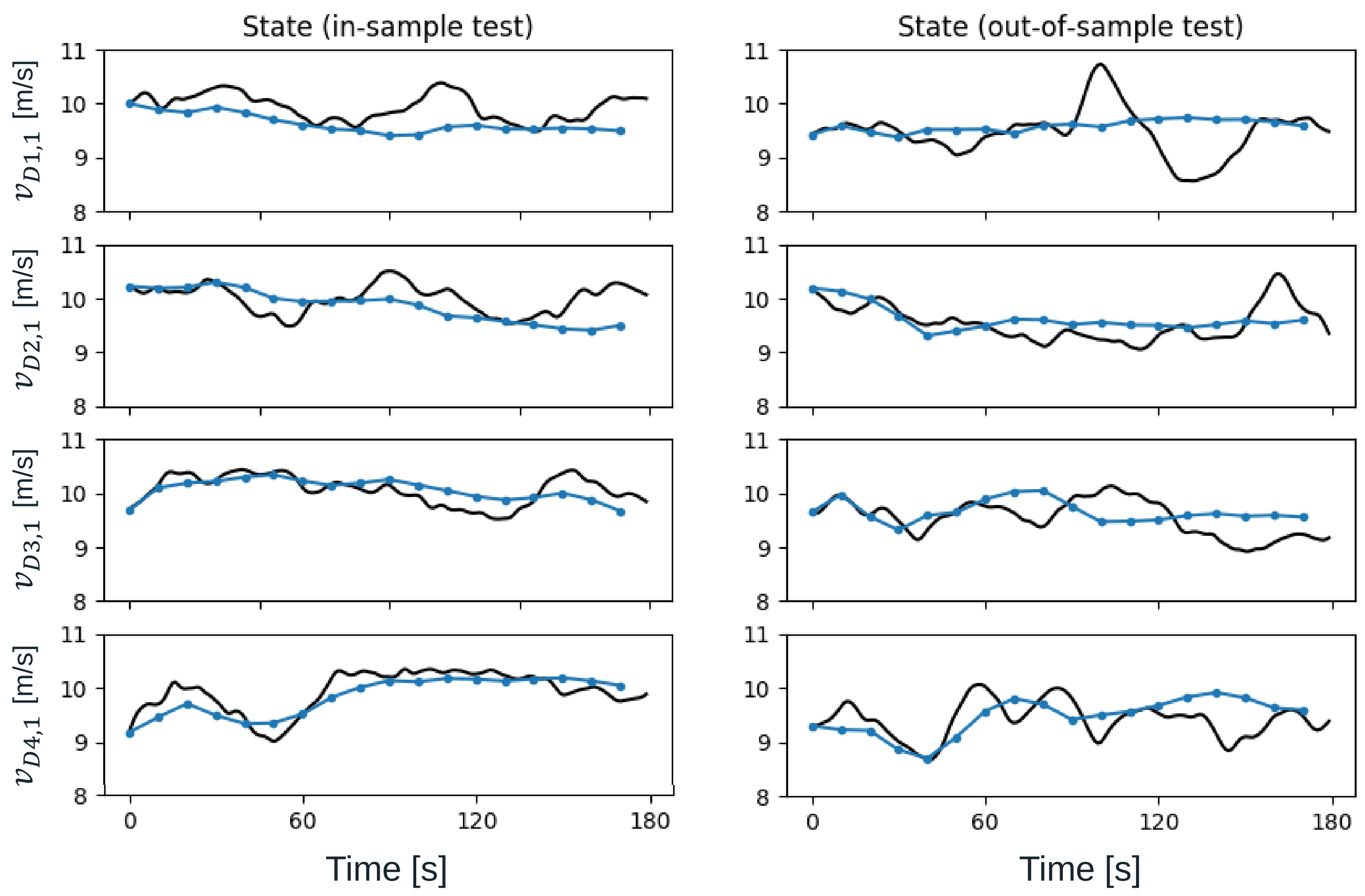
Linear model: no turbines

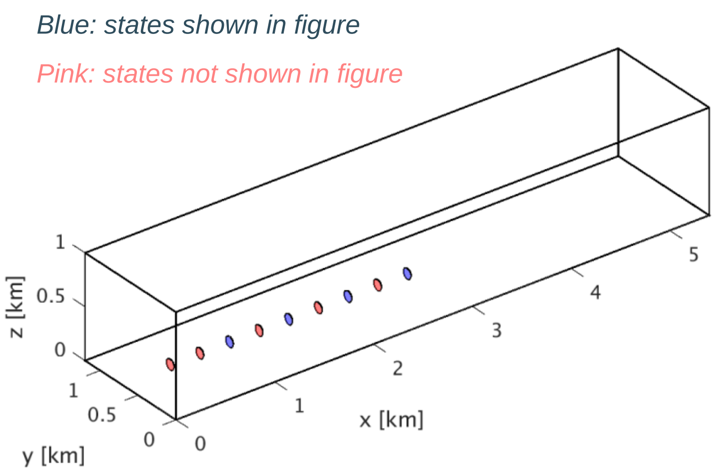




Nonlinear model (LES)

Linear model: no turbines



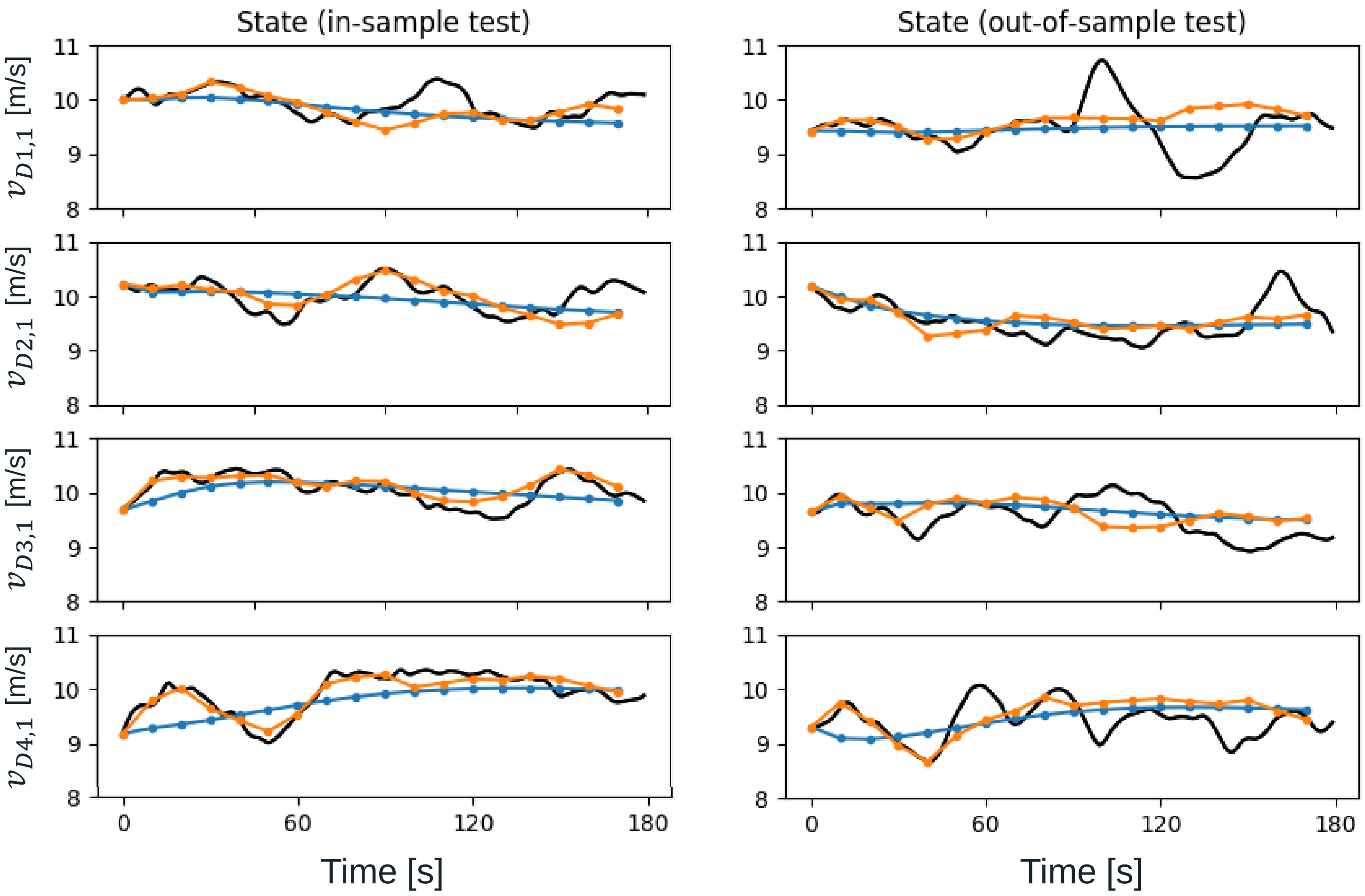


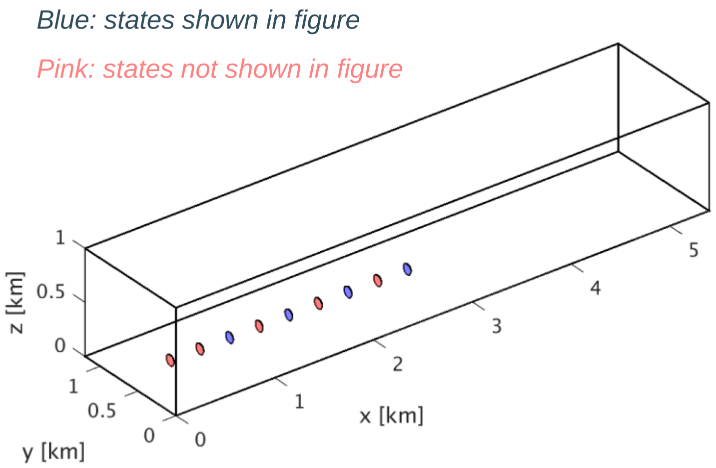
Nonlinear model (LES)

Model 1: disk velocities, no delay

$$x_k = [v_{D1..4}, v_{-3D1..4}]$$
$$x_{k,d} = x \quad (\tau = 0\ s)$$
$$z_k = g(x_{k,d}) = [x_{k,d}, 1]$$

Linear model: no turbines





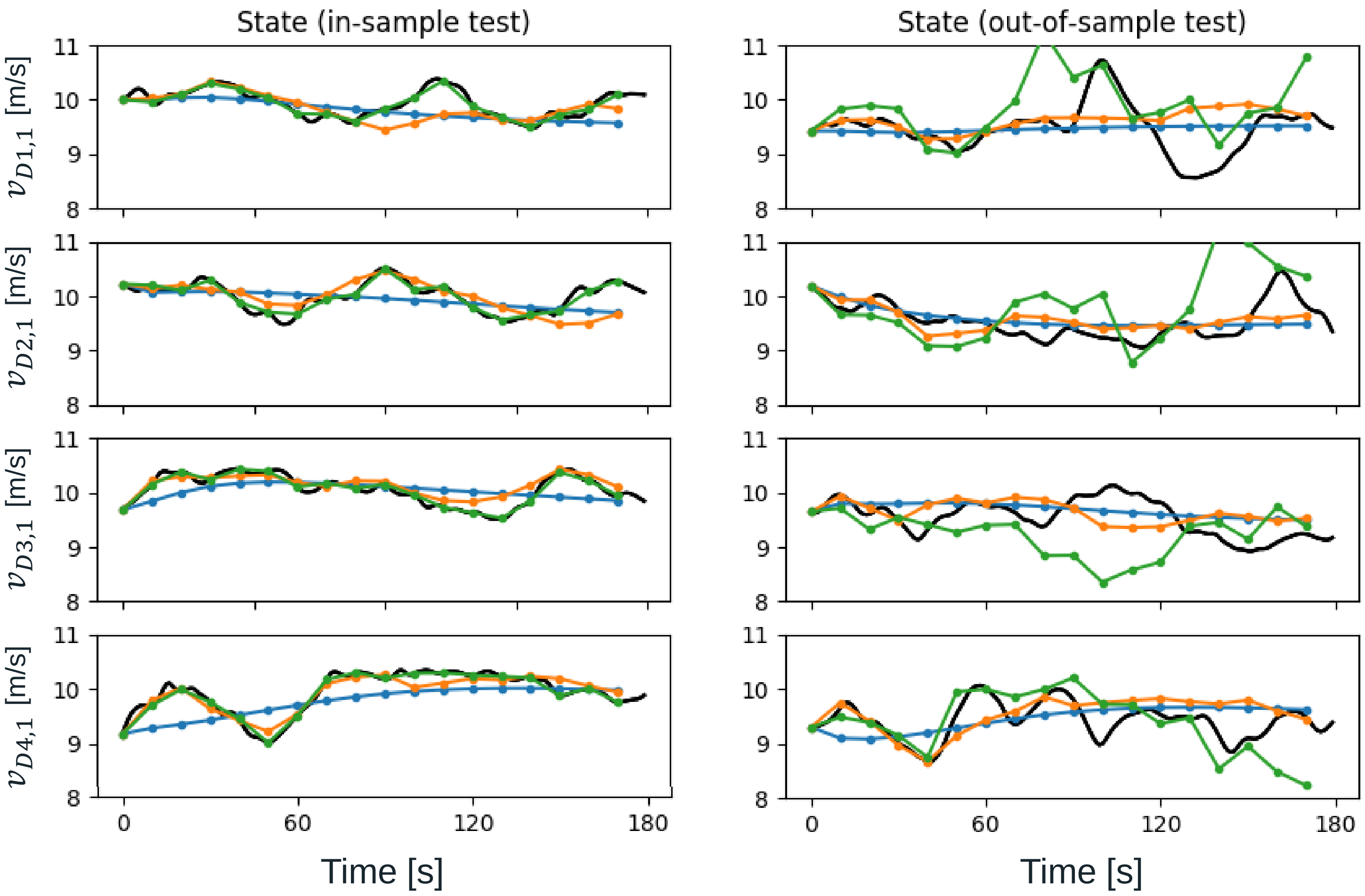
Nonlinear model (LES)

Model 1: disk velocities, no delay

Model 2: disk velocities, 120 s delay

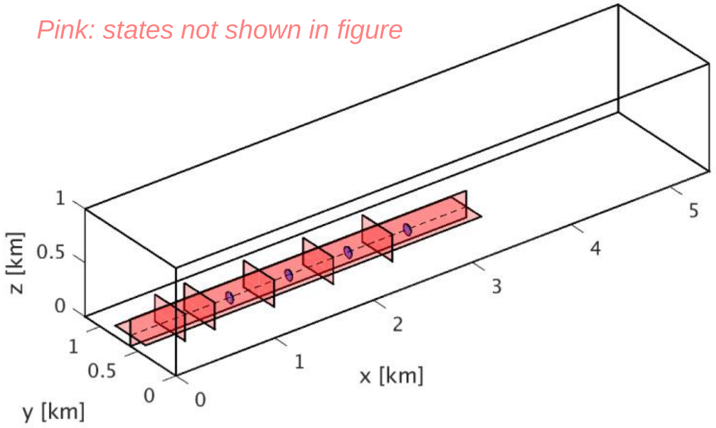
$$\boldsymbol{x}_k = [\boldsymbol{v}_{D1..4}, \boldsymbol{v}_{-3D1..4}]$$
$$\boldsymbol{x}_{k,d} = \boldsymbol{x} \quad (\tau = 120\ s)$$
$$\mathbf{z}_k = \boldsymbol{g}(\boldsymbol{x}_{k,d}) = [\boldsymbol{x}_{k,d}, 1]$$

Linear model: no turbines



Blue: states shown in figure

Pink: states not shown in figure



Nonlinear model (LES)

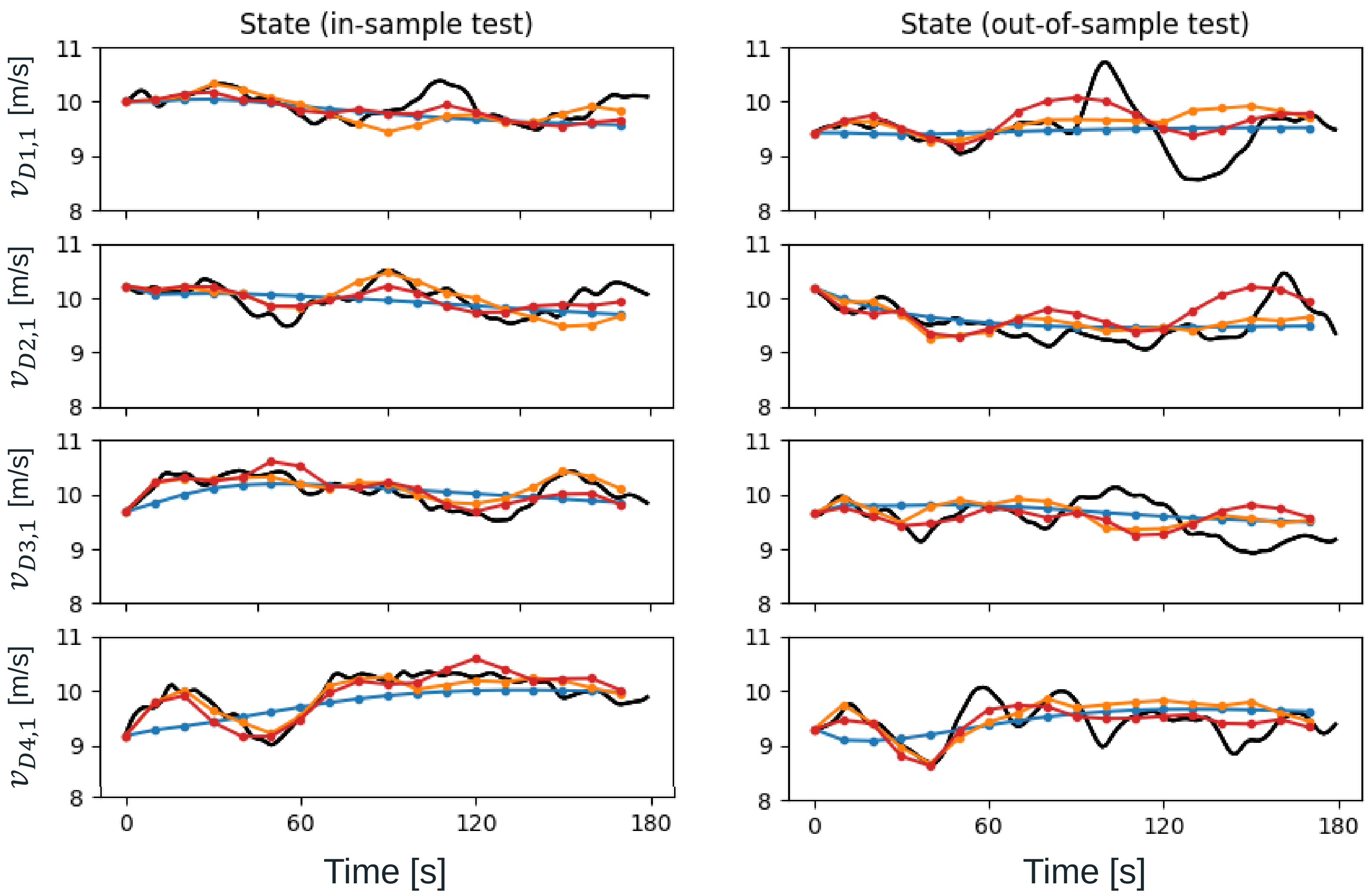
Model 1: disk velocities, no delay

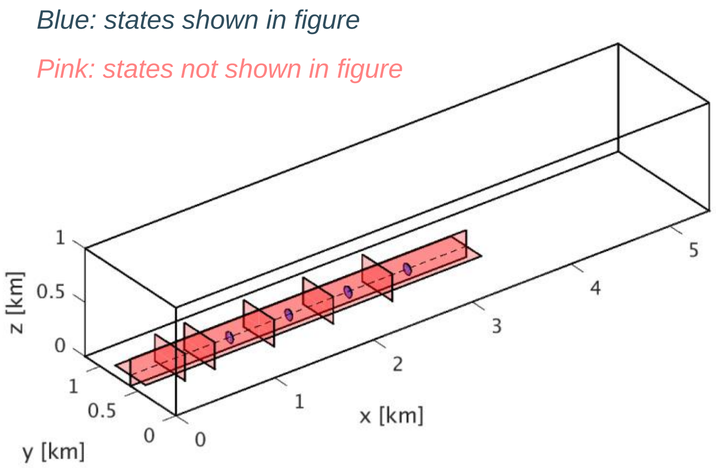
Model 2: disk velocities, 120 s delay

Model 3: disk velocities, sections, lift, no delay

$$\boldsymbol{x}_k = [\boldsymbol{v}_{D1..4}, \boldsymbol{v}_{-3D1..4}, \boldsymbol{v}_{plan}, \boldsymbol{v}_{side}, \boldsymbol{v}_{front}]$$
$$\boldsymbol{x}_{k,d} = \boldsymbol{x} \qquad (\tau = 0\ s)$$
$$\mathbf{z}_k = \boldsymbol{g}(\boldsymbol{x}_{k,d}) = [\boldsymbol{x}_{k,d}, 1, (\boldsymbol{x} \cdot \boldsymbol{\nabla})\boldsymbol{x}]$$

Linear model: no turbines





Nonlinear model (LES)

Model 1: disk velocities, no delay

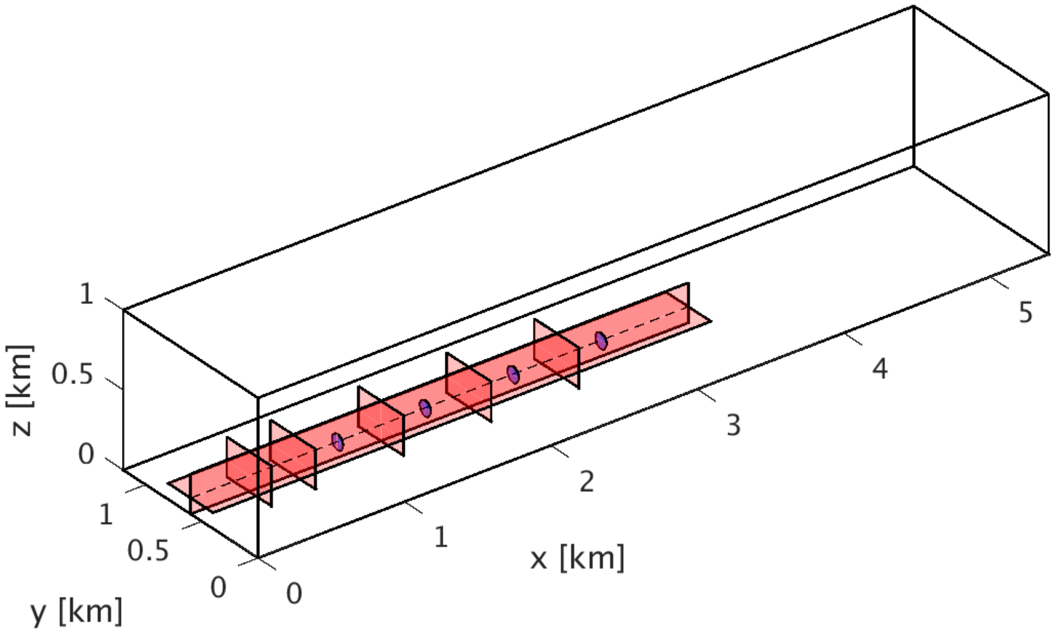
Model 2: disk velocities, 120 s delay

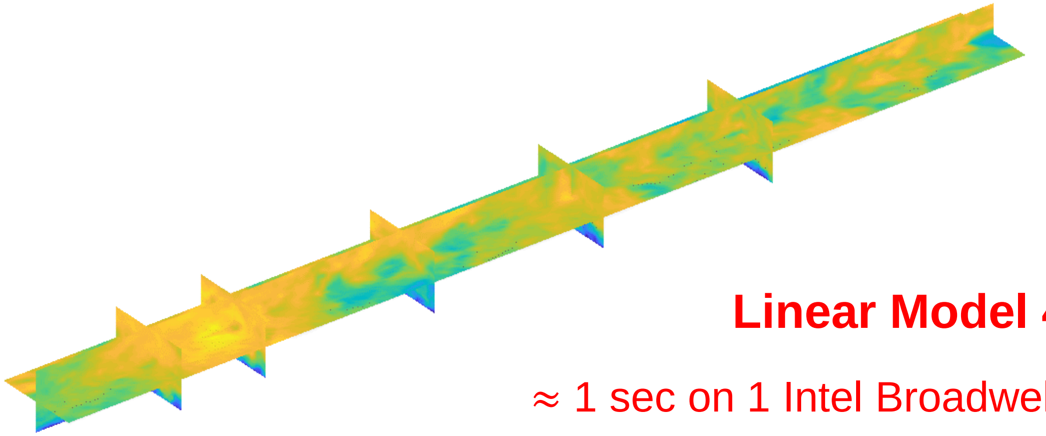
Model 3: disk velocities, sections, lift, no delay

Model 4: disk velocities, sections, lift, no delay + model reduction (250 modes)

$$\boldsymbol{\zeta}_{k+1} = \bar{K}\mathbf{z}_k \rightarrow \tilde{\boldsymbol{\zeta}}_{k+1} = P_{\zeta}^{*}\bar{K}P_{\mathbf{z}}\mathbf{z}_k$$

Linear model: no turbines



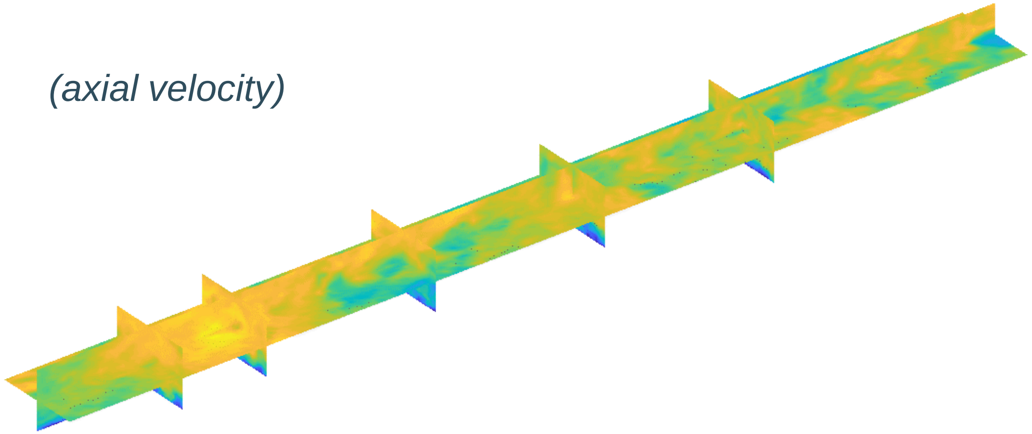


Linear Model 4

≈ 1 sec on 1 Intel Broadwell

t = 10 s

(axial velocity)



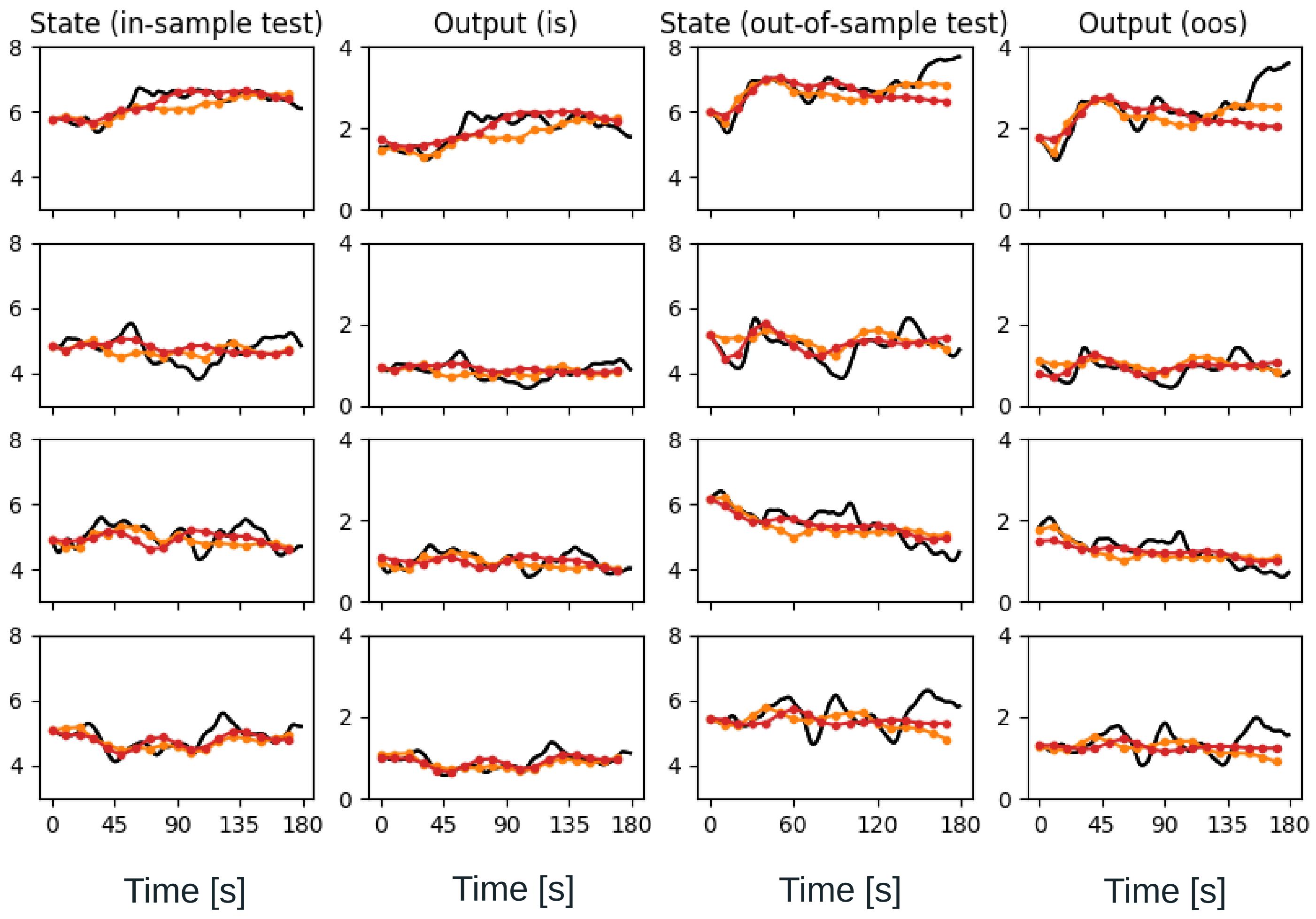
Large Eddy Simulation

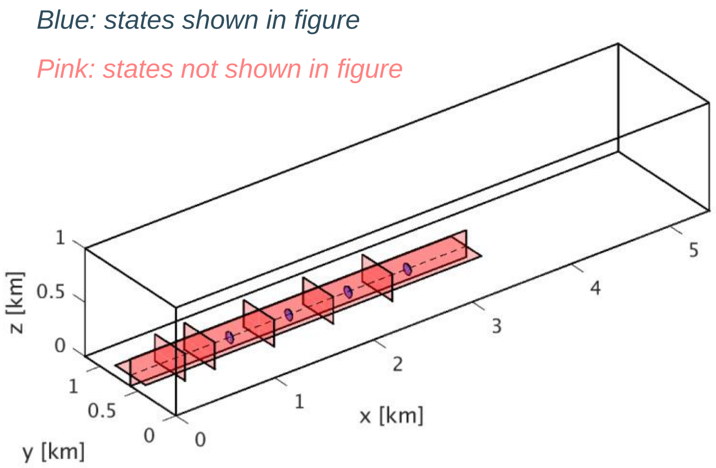
≈ 240 sec on 1 Intel Broadwell

Model captures flow advection
& supports turbulent fluctuations

(1 frame = 10 sec)

Linear model: steady turbines





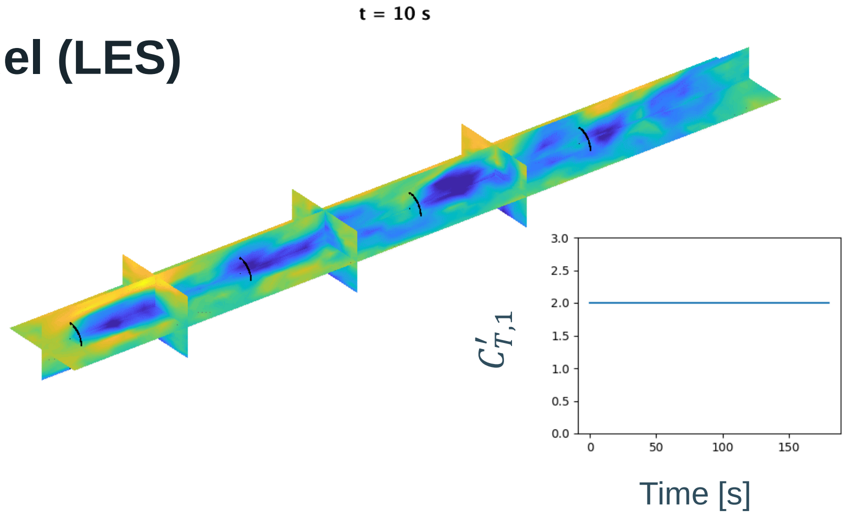
Nonlinear model (LES)

Model 2: disk velocities, 120 s delay

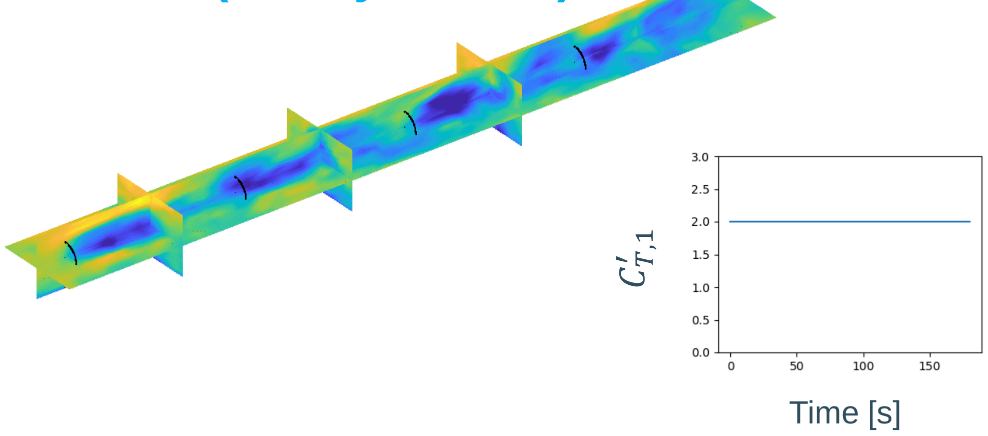
Model 4: disk velocities, sections, lift, no delay + model reduction (250 modes)

Linear model: controlled turbines

Nonlinear model (LES)



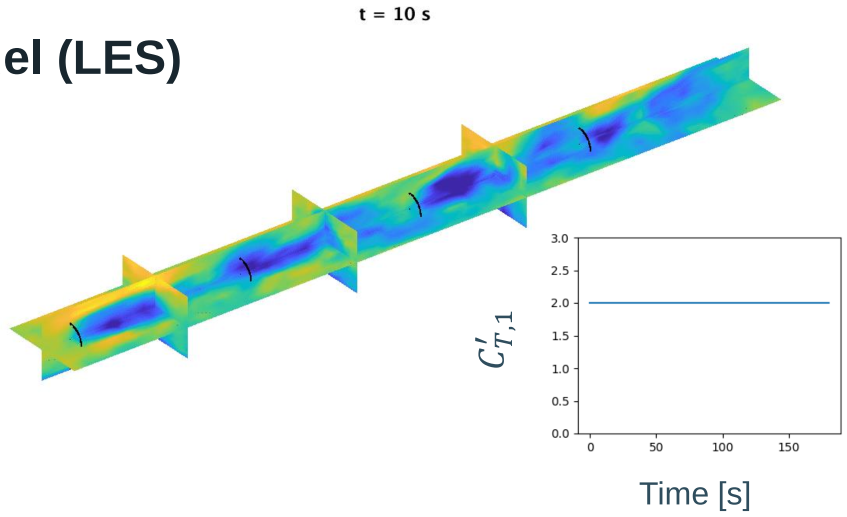
Linear model (steady turbine)



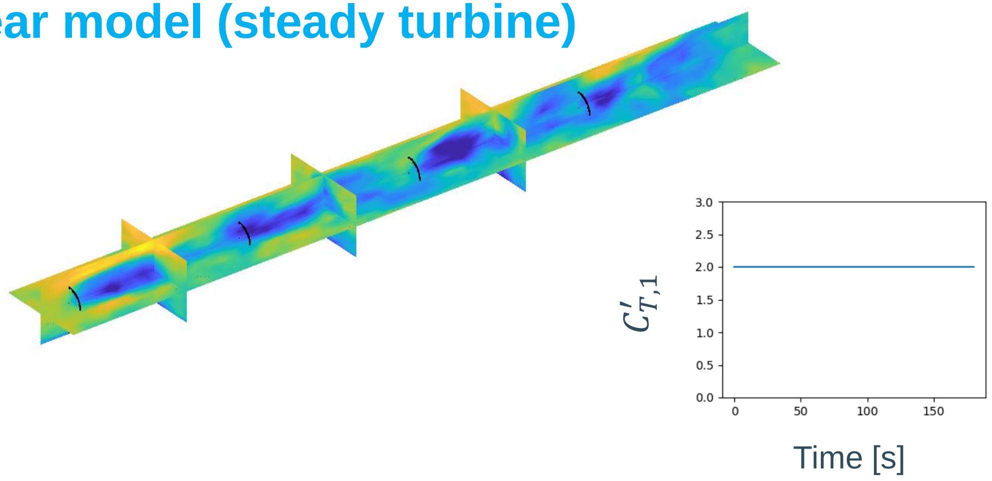
(1 frame = 10 sec)

Linear model: controlled turbines

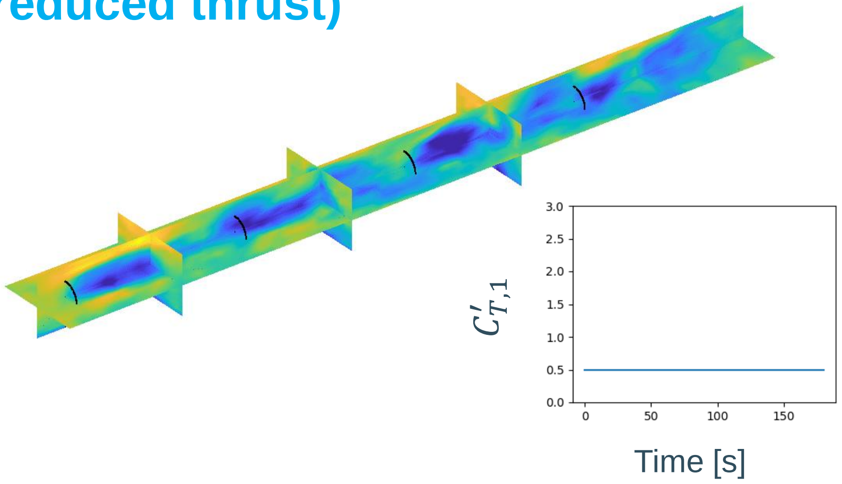
Nonlinear model (LES)



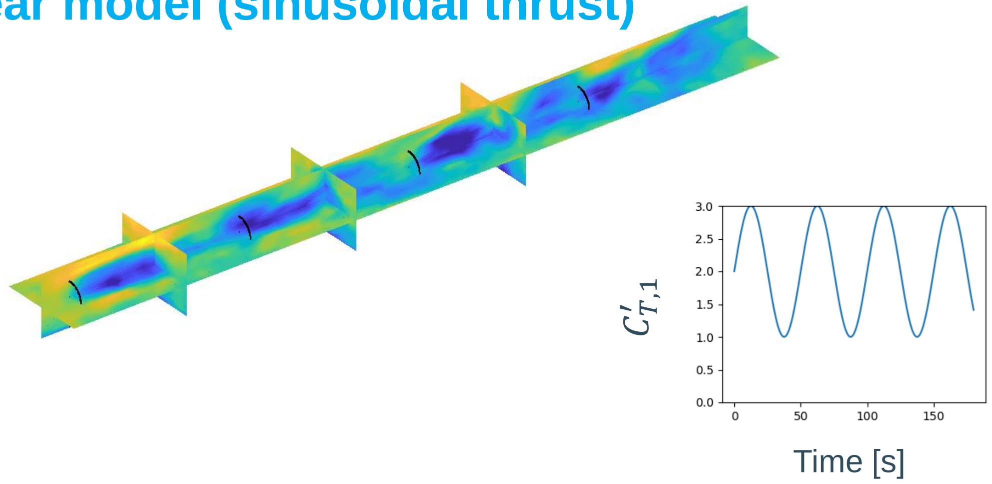
Linear model (steady turbine)



Linear model (reduced thrust)



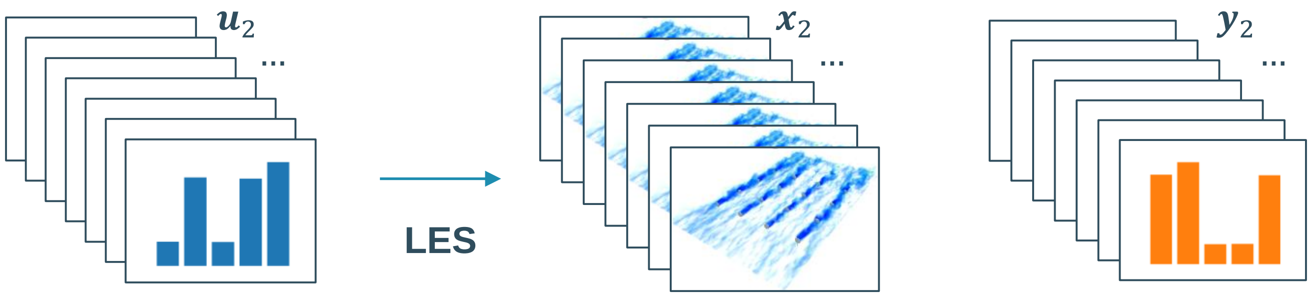
Linear model (sinusoidal thrust)



(1 frame = 10 sec)

Conclusion

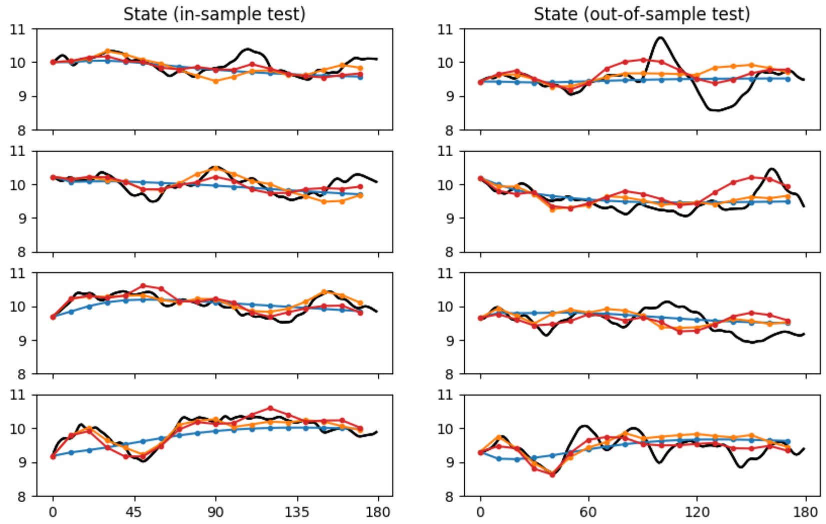
- Data-driven extendable approximation of Koopman operator



Conclusion

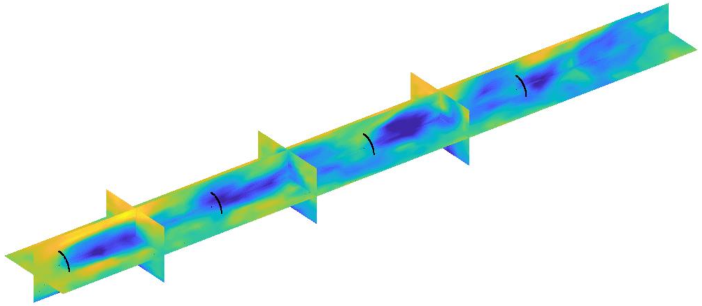
- Data-driven extendable approximation of Koopman operator

- Linear & fast models capture advection & some turbulent fluctuations



Conclusion

- Data-driven extendable approximation of Koopman operator
- Linear & fast models capture advection & some turbulent fluctuations
- Models can be used to explore control strategies at fraction of LES cost



Conclusion

- Data-driven extendable approximation of Koopman operator
- Linear & fast models capture advection & some turbulent fluctuations
- Models can be used to explore control strategies at fraction of LES cost
- Further work on fidelity before use as standalone predictive control model …

The authors received funding from the **European Union** Horizon 2020 research and innovation program under grant agreement no. 727680 (**TotalControl**)

The computational resources and services used in this work were provided by the **VSC (Flemish Supercomputer Center)**, funded by the Research Foundation - Flanders (FWO) and the Flemish Government – department EWI

A data-driven flow model for wind-farm control based on Koopman mode decomposition of large-eddy simulations

Wim Munters, Johan Meyers

Department of Mechanical Engineering

KU Leuven, Leuven, Belgium

71st APS DFD Meeting, Atlanta, GA, USA

19/11/2018

Funded by the Horizon 2020 programme of the European Union