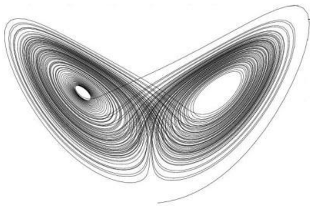
The Koopman operator framework for analysis and control of nonlinear dynamical systems

Milan Korda

(LAAS, CNRS)



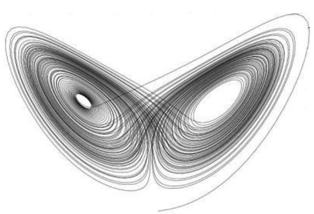
$$x^+ = f(x)$$

Nonlinear system

Linear operator

$$\mathcal{K}g = g \circ f$$

$\nearrow$



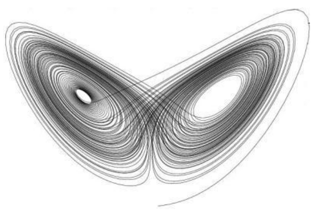
$$x^+ = f(x)$$

Nonlinear system

Linear operator

$$\mathcal{K} g = g \circ f$$

$\Nearrow$ $\Searrow$



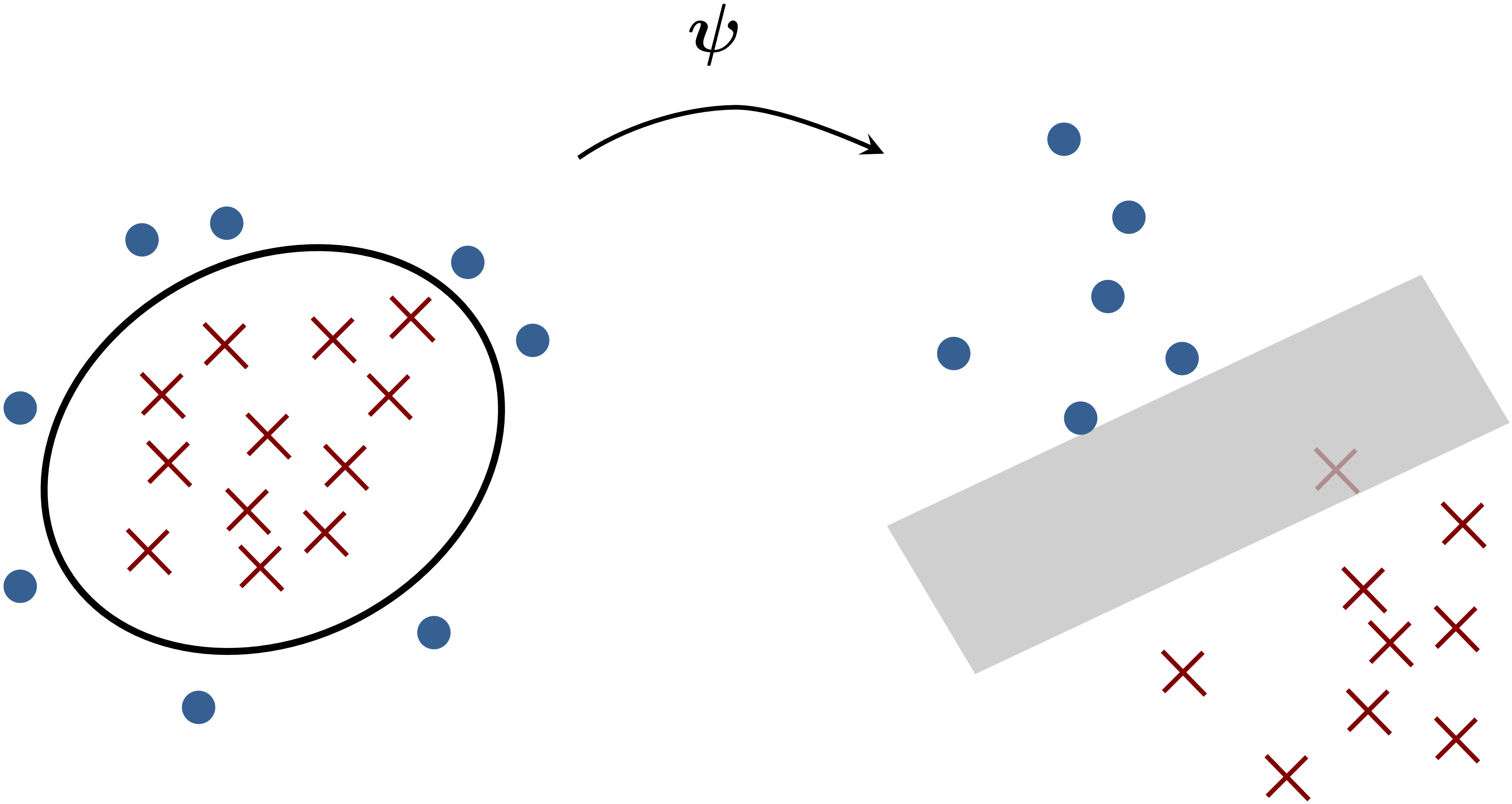
$$x^+ = f(x)$$

Nonlinear system

Analysis,

Prediction & Control

using **linear** techniques


ψ

Koopman operator

Koopman operator

$$\mathcal{K} : g \mapsto g \circ f$$

$$g : X \to \mathbb{C}$$

Koopman operator

$$\mathcal{K} : g \mapsto g \circ f \qquad g : X \to \mathbb{C}$$

Linearity

$$\begin{aligned}\mathcal{K}(\alpha g_1 + \beta g_2) &= (\alpha g_1 + \beta g_2) \circ f \\ &= \alpha g_1 \circ f + \beta g_2 \circ f \\ &= \alpha \mathcal{K} g_1 + \beta \mathcal{K} g_2\end{aligned}$$

Koopman operator

$$\mathcal{K} : g \mapsto g \circ f$$

$$g : X \to \mathbb{C}$$

$(1900 - 1981)$

[B. O. Koopman, 1931]

⋮

[Mezić, Banaszuk, 2004]

Koopman operator

$$\mathcal{K} : g \mapsto g \circ f \qquad g : X \to \mathbb{C}$$

Eigenfunctions

$$\mathcal{K}\phi = \lambda\phi \quad \Leftrightarrow \quad \phi \circ f = \lambda\phi$$

Koopman operator

$$\mathcal{K} : g \mapsto g \circ f$$

$$g : X \to \mathbb{C}$$

Eigenfunctions

$$\mathcal{K}\phi = \lambda\phi \quad \Leftrightarrow \quad \phi \circ f = \lambda\phi$$

$$\phi \circ f^{(k)} = \lambda^k \phi$$

Linear coordinate

Koopman operator

$$\mathcal{K} : g \mapsto g \circ f \qquad g : X \to \mathbb{C}$$

Eigenfunctions $$\mathcal{K}\phi = \lambda\phi \quad \Leftrightarrow \quad \phi \circ f = \lambda\phi$$

(1) $\phi_1, \ldots, \phi_N$ eigenfunctions $\Rightarrow \phi_1^{k_1}, \ldots, \phi_N^{k_N}$

(2) ϕ eigenfunction $\Rightarrow |\phi|$ eigenfunction

(3) ϕ eigenfunction $\Rightarrow \phi^*$ eigenfunction

Examples: linear system

$$x^+ = Ax$$

$$w^\top A = \lambda w^\top \quad \Rightarrow \quad w^\top x \text{ eigenfunction with eigenvalue } \lambda$$

Examples: linear system

$$x^+ = Ax$$

$$w^\top A = \lambda w^\top \quad \Rightarrow \quad w^\top x \text{ eigenfunction with eigenvalue } \lambda$$

$w_1, \ldots, w_n$ left eigenvectors of A

$$\Rightarrow (w_1^\top x)^{k_1} \cdot \ldots \cdot (w_n^\top x)^{k_n} \text{ eigenfunctions}$$

with eigenvalues $\lambda_1^{k_1} \cdot \ldots \cdot \lambda_n^{k_n}$

Examples: linear system

$$x^+ = Ax$$

$$w^\top A = \lambda w^\top \quad \Rightarrow \quad w^\top x \text{ eigenfunction with eigenvalue } \lambda$$

$w_1, \ldots, w_n$ left eigenvectors of A

$$\Rightarrow (w_1^\top x)^{k_1} \cdot \ldots \cdot (w_n^\top x)^{k_n} \text{ eigenfunctions}$$

with eigenvalues $\lambda_1^{k_1} \cdot \ldots \cdot \lambda_n^{k_n}$

k_i's integers $\Rightarrow$ eigenfunctions are polynomials

& spectrum has **lattice** structure

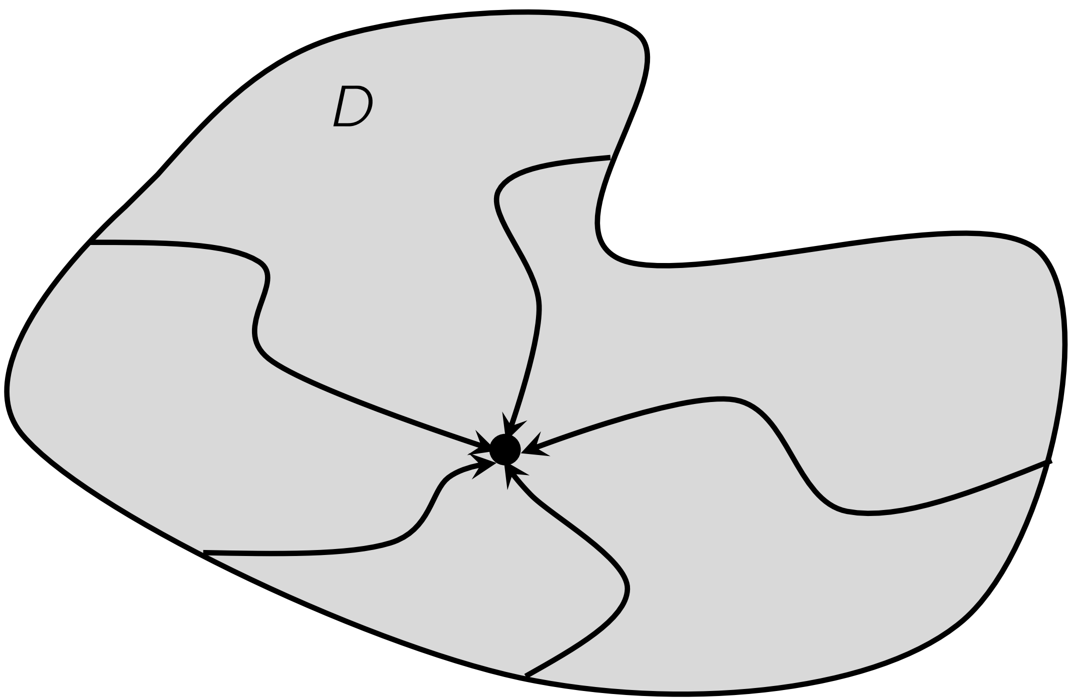
Examples: stable nonlinear system

$$x^+ = f(x)$$

Stable equilibrium

Any $\lambda \in \mathbb{C}$ with $|\lambda| \leq 1$ is an eigenvalue with an eigenfunction $\phi \in \mathcal{C}(D^\circ)$

Set of all eigenfunctions dense in $\mathcal{C}(D^\circ)$



Examples: rotation

$$x^{+} = x + \theta \bmod 1$$

Eigenfunctions $e^{i2\pi xk}$, $k \in \mathbb{Z}$

with eigenvalues $e^{i2\pi\theta k}$

θ **rational** $\Rightarrow$ spectrum finite discrete subset of $\mathbb{T}$ $\Rightarrow$ periodic dynamics

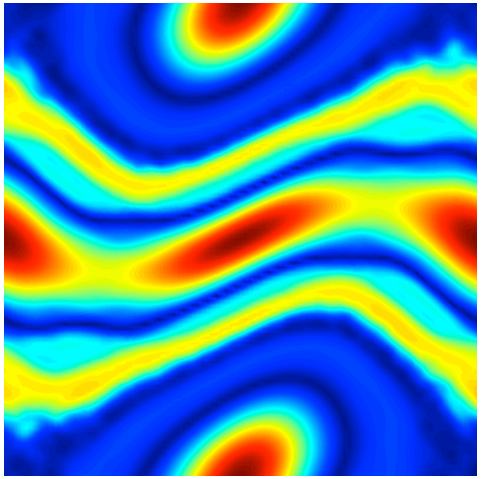
θ **irrational** $\Rightarrow$ spectrum dense in $\mathbb{T}$ $\Rightarrow$ ergodic dynamics

Koopman operator

Eigenfunctions

$$\phi \circ f^{(k)} = \lambda^k \phi$$

$$\lambda = 1 \quad \Rightarrow \quad \{x \,:\, \phi(x) = \gamma\}$$ invariant set



Chirikov standard map

Koopman operator

Eigenfunctions

$$\phi \circ f^{(k)} = \lambda^k \phi$$

$$|\lambda| \leq 1 \quad \Rightarrow \quad \{x \;:\; |\phi|(x) \leq \gamma\}$$ invariant set

Chirikov standard map

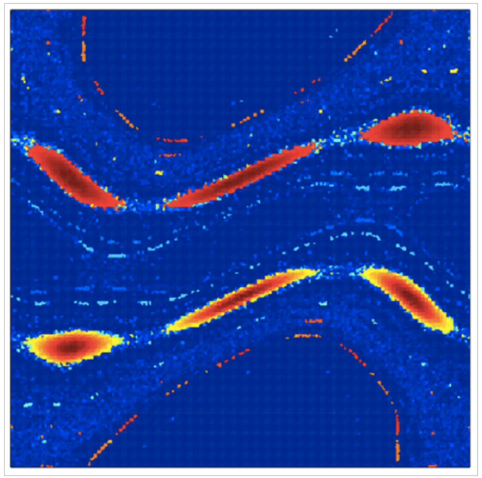
Koopman operator

Eigenfunctions

$$\phi \circ f^{(k)} = \lambda^k \phi$$

$$\lambda = e^{i\omega} \quad \Rightarrow \quad \{x \ : \ \phi(x) = \gamma\}$$ periodic set

(ω rational)



Chirikov standard map

[Budisic et al. 2012]

Koopman operator

Eigenfunctions

$$\phi \circ f^{(k)} = \lambda^k \phi$$

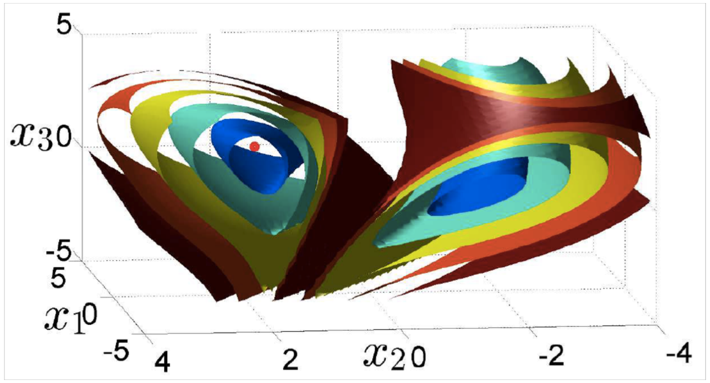
Isostables

Isochrons

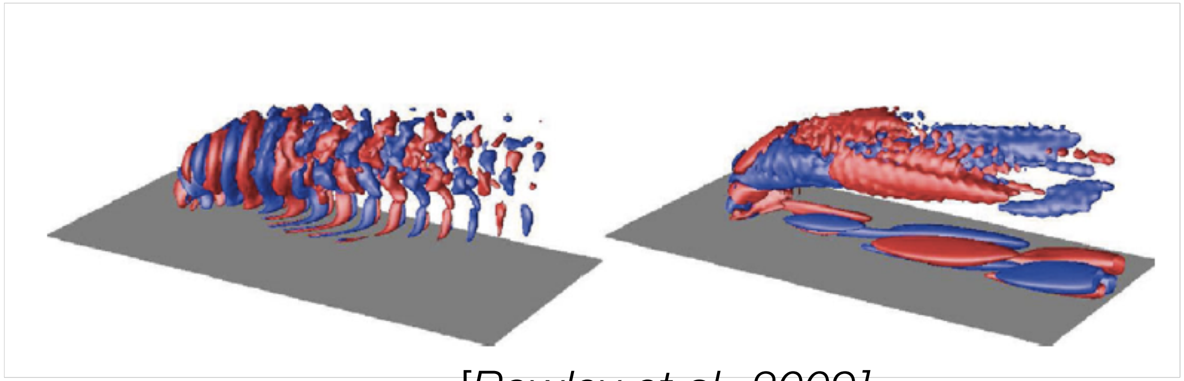
Stability

Model reduction

⋮



[Mauroy et al. 2013]



[Rowley et al. 2009]

Prediction

Prediction

Eigenfunctions $\phi_1, \ldots, \phi_N$

$$\boldsymbol{g} = \sum_{i=1}^{N} c_i \phi_i \quad \Rightarrow \quad \boldsymbol{g} \circ f = \sum_{i=1}^{N} c_i \lambda_i \phi_i$$

(e.g., $\boldsymbol{g}(x) = x$)

Prediction

Eigenfunctions $\phi_1, \ldots, \phi_N$

$$\boldsymbol{g} = \sum_{i=1}^{N} c_i \phi_i \quad \Rightarrow \quad \boldsymbol{g} \circ f = \sum_{i=1}^{N} c_i \lambda_i \phi_i$$

(e.g., $\boldsymbol{g}(x) = x$)

$$\boldsymbol{g} \circ f^{(2)} = \sum_{i=1}^{N} c_i \lambda_i^2 \phi_i$$

$$\vdots$$

$$\boldsymbol{g} \circ f^{(k)} = \sum_{i=1}^{N} c_i \lambda_i^k \phi_i$$

Prediction

Eigenfunctions $\phi_1, \ldots, \phi_N$

$$\boldsymbol{g} = \sum_{i=1}^{N} c_i \phi_i \quad \Rightarrow \quad \boldsymbol{g} \circ f = \sum_{i=1}^{N} c_i \lambda_i \phi_i$$

(e.g., $\boldsymbol{g}(x) = x$)

$$\boldsymbol{g} \circ f^{(k)} = \underbrace{[c_1, \ldots, c_N]}_{C} \underbrace{\begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_N \end{bmatrix}^k}_{A} \underbrace{\begin{bmatrix} \phi_1 \\ \vdots \\ \phi_N \end{bmatrix}}_{\boldsymbol{\psi}}$$

$$\boldsymbol{g} \circ f^{(k)} = C A^k \boldsymbol{\psi}$$

Prediction

Eigenfunctions $\phi_1, \ldots, \phi_N$

$$\boldsymbol{g} = \sum_{i=1}^{N} c_i \phi_i \quad \Rightarrow \quad \boldsymbol{g} \circ f = \sum_{i=1}^{N} c_i \lambda_i \phi_i$$

(e.g., $\boldsymbol{g}(x) = x$)

New coordinates $z = \boldsymbol{\psi}(x)$ ("eigencoordiantes")

$$\Rightarrow z_{k+1} = A z_k$$

Prediction

Eigenfunctions $\phi_1, \ldots, \phi_N$

$$\boldsymbol{g} = \sum_{i=1}^{N} c_i \phi_i \quad \Rightarrow \quad \boldsymbol{g} \circ f = \sum_{i=1}^{N} c_i \lambda_i \phi_i$$

(e.g., $\boldsymbol{g}(x) = x$)

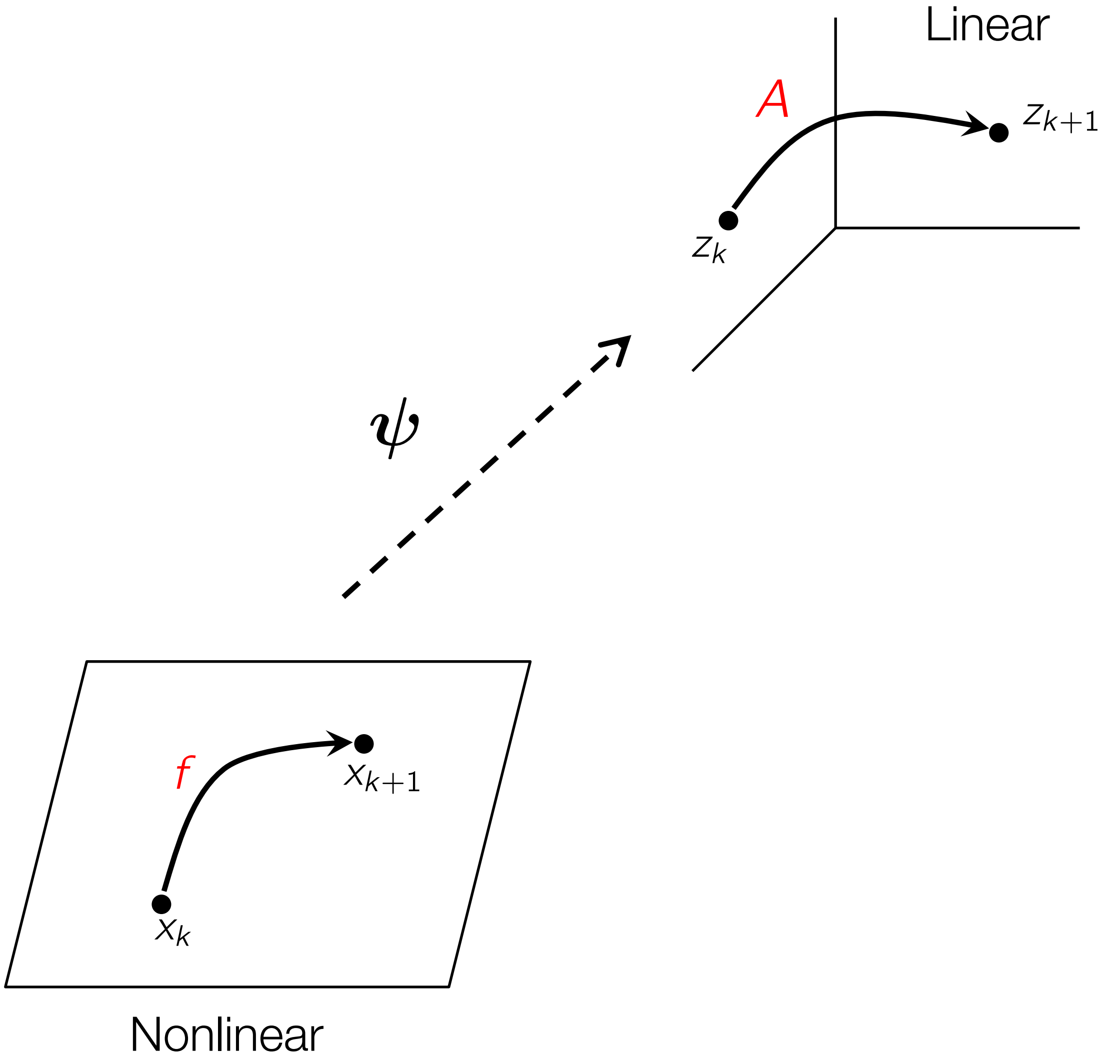
Linear predictor

$$z_{k+1} = A z_k$$
$$y_k = C z_k$$
$$z_0 = \boldsymbol{\psi}(x_0)$$

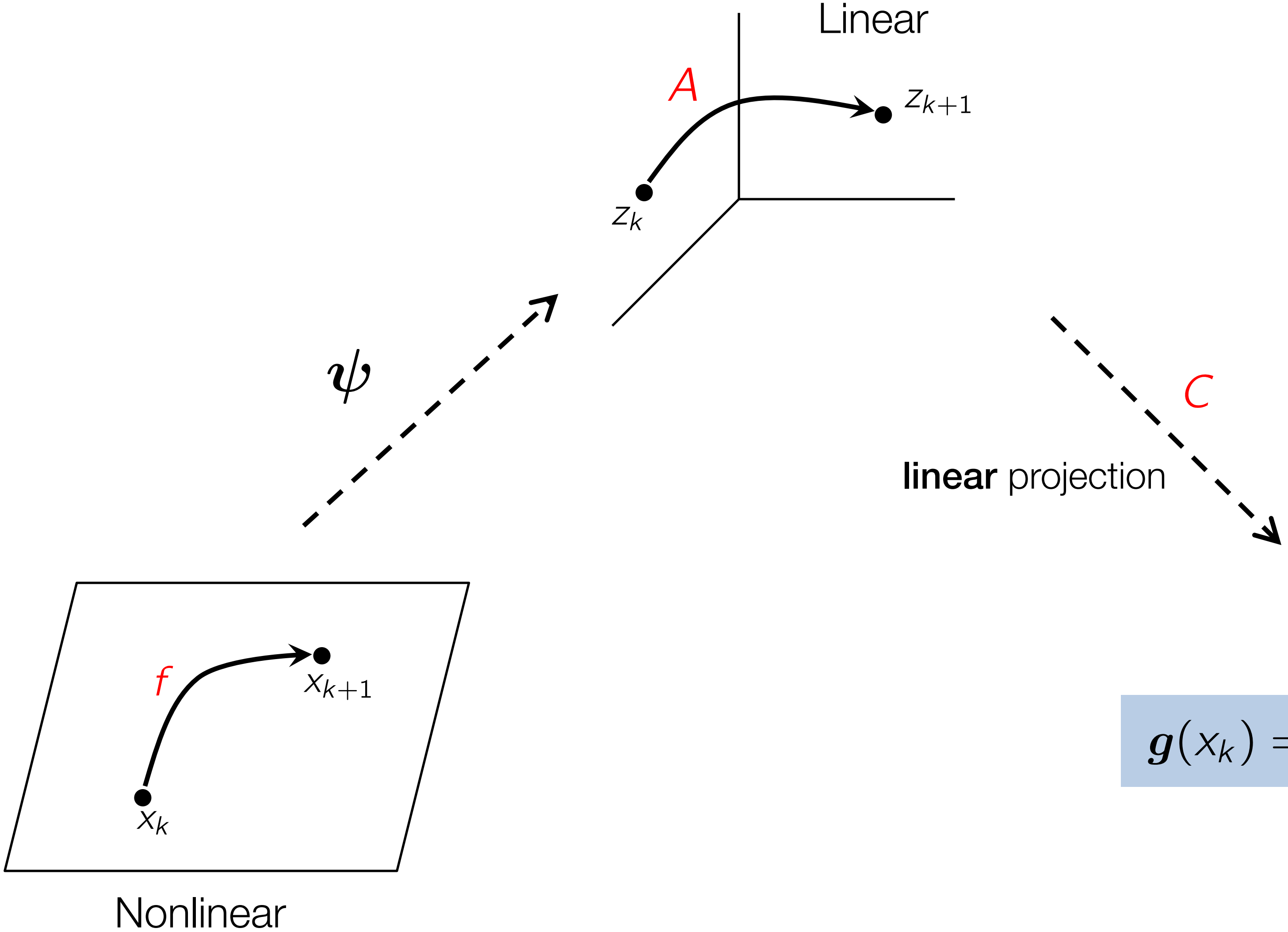
$\Rightarrow$ $y_k = \boldsymbol{g}(x_k)$

$$C = [c_1, \ldots, c_N] \qquad \boldsymbol{\psi}(x_0) = \begin{bmatrix} \phi_1(x_0) \\ \vdots \\ \phi_N(x_0) \end{bmatrix} \qquad A = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$$

Nonlinear embedding



Nonlinear embedding



$$\boldsymbol{g}(x_k) = C z_k$$

Are eigenfunctions the only choice?

$$z_{k+1} = A z_k$$
$$z_0 = \boldsymbol{\psi}(x_0)$$
$$y_k = C z_k$$

$$y_k = \boldsymbol{g}(x_k)$$

Exact **linear** prediction possible if

$\mathrm{span}\{\psi_1, \ldots, \psi_N\}$ is Koopman invariant   & $\boldsymbol{g} \in \mathrm{span}\{\psi_1, \ldots, \psi_N\}$

$\Rightarrow$ Eigenfunctions and **generalized** eigenfunctions
(or linear combinations thereof)

This talk: Assume ψ given

Constructing good ψ: *[Korda, Mezić, 2018]*

Getting A and B from data

Data $(x_i)_{i=1}^{M}$ $(x_i^+)_{i=1}^{M}$ $x_i^+ = f(x_i)$

Basis functions $\psi = [\psi_1, \ldots, \psi_N]^\top$

LS problem

$$\min_{A \in \mathbb{R}^{N\times N}} \sum_{i=1}^{M} \|\boldsymbol{\psi}(x_i^+) - A\boldsymbol{\psi}(x_i)\|_2^2$$

LS problem

$$\min_{C \in \mathbb{R}^{N\times N}} \sum_{i=1}^{M} \|\boldsymbol{g}(x_i) - C\boldsymbol{\psi}(x_i)\|_2^2$$

Extended dynamic mode decomposition *[Williams et al., 2015]*

Convergence of predictions

Finite-horizon predictions converge!

Theorem *[Korda, Mezić, 2018]*

- $\mathcal{K} : L_2(\mu) \to L_2(\mu)$, bounded
- $\overline{\mathrm{span}\{\psi_i\}_{i=1}^{\infty}} = L_2(\mu)$

$$\lim_{N\to\infty} \lim_{M\to\infty} \int_X |C A_{N,M}^k \psi_N - g \circ f^k|^2 \, d\mu \to 0$$

for any $k \in \mathbb{N}$

Convergence of predictions

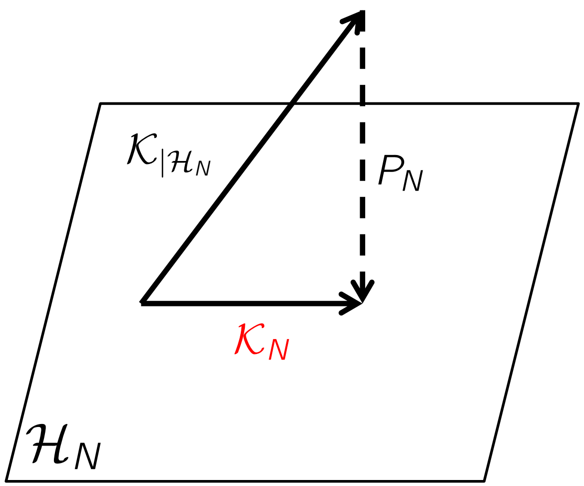
Finite-horizon predictions converge!

Theorem *[Korda, Mezić, 2018]*

- $\mathcal{K} : L_2(\mu) \to L_2(\mu)$, bounded
- $\overline{\mathrm{span}\{\psi_i\}_{i=1}^{\infty}} = L_2(\mu)$

$$\lim_{N\to\infty} \lim_{M\to\infty} \int_X |C A_{N,M}^k \psi_N - g \circ f^k|^2 \, d\mu \to 0$$

for any $k \in \mathbb{N}$



$$\mathcal{K}_N := P_N \mathcal{K}_{|\mathcal{H}_N}$$

$$\mathcal{H}_N := \mathrm{span}\{\psi_1, \ldots, \psi_N\}$$

Application – State estimation *[Surana et al. 2016]*

$$\boldsymbol{\psi}$$

	$x^+ = f(x)$	$\xrightarrow{\boldsymbol{\psi}}$	$z^+ = Az$
measured	$y_1 = \boldsymbol{h}(x)$		$y_1 = C_1 z$
to estimate	$y_2 = \boldsymbol{g}(x)$		$y_2 = C_2 z$

$$\boldsymbol{h} \in \mathrm{span}\{\boldsymbol{\psi}\} \quad \Leftrightarrow \quad \boldsymbol{h}(x) = C_1 \boldsymbol{\psi}(x)$$

$$\boldsymbol{g} \in \mathrm{span}\{\boldsymbol{\psi}\} \quad \Leftrightarrow \quad \boldsymbol{g}(x) = C_2 \boldsymbol{\psi}(x)$$

Kalman filter $\quad \hat{z}^+ = A\hat{z} + L(y_1 - C_1 \hat{z})$

$C_2 \hat{z}$ is an estimate of $\boldsymbol{g}(x)$

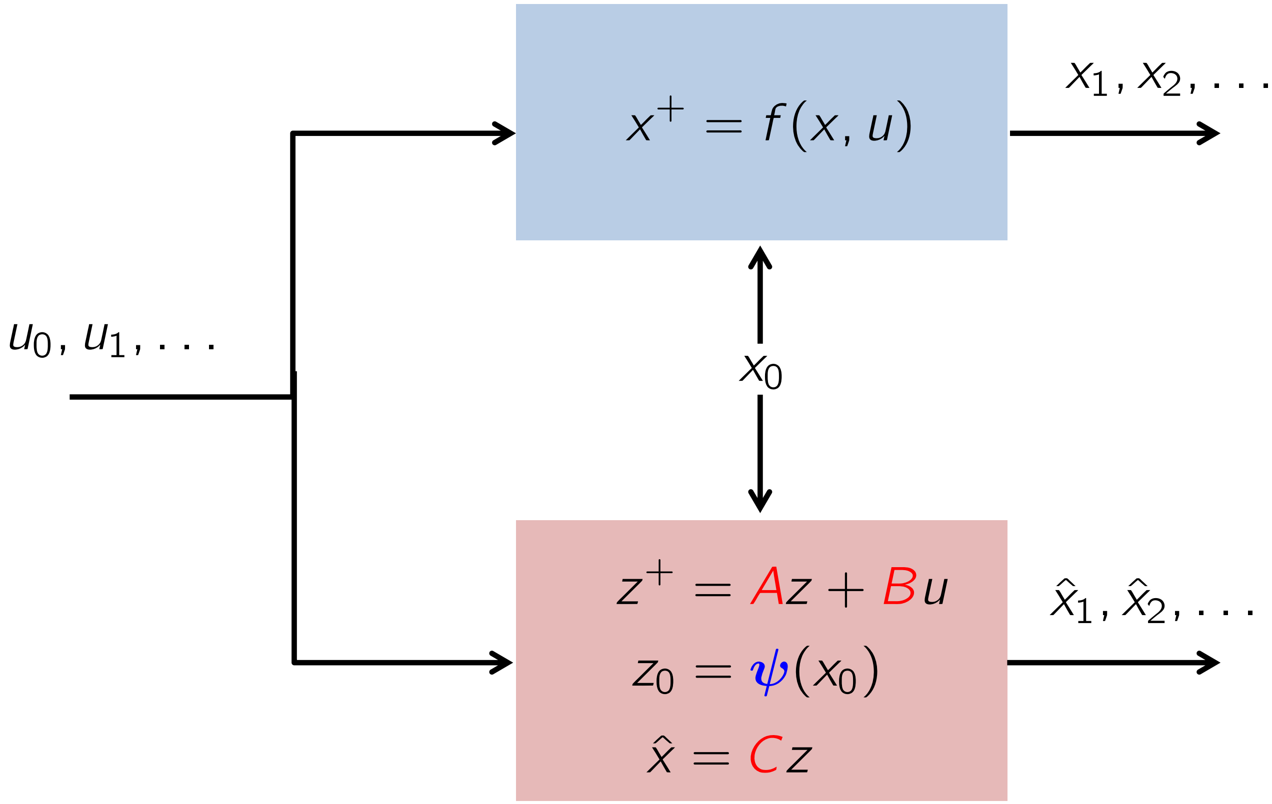
$(A - LC_1)$ stable and $\mathrm{span}\{\boldsymbol{\psi}\}$ Koopman invariant $\Rightarrow \|C_2 \hat{z}_k - \boldsymbol{g}(x_k)\| \to 0$

Control

Control

Joint work with Igor Mezić

Linear predictor



$$\psi : \mathbb{R}^n \to \mathbb{R}^N, \qquad N \gg n$$

Koopman operator for controlled systems

$$x^+ = f(x, u),\ x \in \mathbb{R}^n,\ u \in \mathbb{R}^m$$

Koopman operator for controlled systems

$$x^+ = f(x,u),\ x \in \mathbb{R}^n,\ u \in \mathbb{R}^m$$

$$\Downarrow$$

$$\chi^+ = F(\chi) := \begin{bmatrix} f(x, \boldsymbol{u}(0)) \\ \mathcal{S}\boldsymbol{u} \end{bmatrix}$$

- Extended state $\chi := (x, \boldsymbol{u}) \in \mathcal{X} := \mathbb{R}^n \times \ell(\mathbb{R}^m)$
- Shift operator $(\mathcal{S}\boldsymbol{u})(i) = \boldsymbol{u}(i+1)$

Space of all control sequences $(u_i)_{i=0}^{\infty} =: \boldsymbol{u}$

Koopman operator for controlled systems

$$x^+ = f(x, u),\ x \in \mathbb{R}^n,\ u \in \mathbb{R}^m$$

$$\Downarrow$$

$$\chi^+ = F(\chi) := \begin{bmatrix} f(x, \boldsymbol{u}(0)) \\ \mathcal{S}\boldsymbol{u} \end{bmatrix}$$

- Extended state $\chi := (x, \boldsymbol{u}) \in \mathcal{X} := \mathbb{R}^n \times \ell(\mathbb{R}^m)$
- Shift operator $(\mathcal{S}\boldsymbol{u})(i) = \boldsymbol{u}(i+1)$

Space of all control sequences $(u_i)_{i=0}^{\infty} =: \boldsymbol{u}$

Koopman operator $\mathcal{K}\phi = \phi \circ F$ $\qquad \phi : \mathcal{X} \to \mathbb{R}$

Linear predictors from Koopman - EDMD

Data $(x_i)_{i=1}^{K}$ $(x_i^+)_{i=1}^{K}$ $x_i^+ = F(x_i)$

LS problem

$$\min_{\mathcal{A} \in \mathbb{R}^{N_\phi \times N_\phi}} \sum_{i=1}^{K} \|\phi(x_i^+) - \mathcal{A}\phi(x_i)\|_2^2$$

$$\phi(x) = [\phi_1(x), \ldots, \phi_{N_\phi}(x)]^\top$$

Linear predictors from Koopman - EDMD

Data $(x_i)_{i=1}^K$ $(x_i^+)_{i=1}^K$ $x_i^+ = F(x_i)$

LS problem

$$\min_{\mathcal{A} \in \mathbb{R}^{N_\phi \times N_\phi}} \sum_{i=1}^{K} \|\phi(x_i^+) - \mathcal{A}\phi(x_i)\|_2^2$$

$$\phi(x) = [\phi_1(x), \ldots, \phi_{N_\phi}(x)]^\top$$

linear operator

Predictor linear in u $\Rightarrow$ $\phi_i(x, \boldsymbol{u}) = \psi_i(x) + \mathcal{L}_i(\boldsymbol{u})$

Linear predictors from Koopman - EDMD

Data $(x_i)_{i=1}^K$ $(x_i^+)_{i=1}^K$ $x_i^+ = F(x_i)$

LS problem

$$\min_{\mathcal{A} \in \mathbb{R}^{N_\phi \times N_\phi}} \sum_{i=1}^{K} \|\phi(x_i^+) - \mathcal{A}\phi(x_i)\|_2^2$$

$$\phi(x) = [\phi_1(x), \ldots, \phi_{N_\phi}(x)]^\top$$

linear operator

Predictor linear in u $\Rightarrow$ $\phi_i(x, \boldsymbol{u}) = \psi_i(x) + \mathcal{L}_i(\boldsymbol{u})$

Without loss of generality

$$\phi(x, \boldsymbol{u}) = [\psi_1(x), \ldots, \psi_N(x), \boldsymbol{u}(0)^\top]^\top$$

Linear predictors from Koopman - EDMD

Data $(x_i)_{i=1}^K$ $(x_i^+)_{i=1}^K$ $x_i^+ = F(x_i)$

LS problem

$$\min_{\mathcal{A}\in\mathbb{R}^{N_\phi\times N_\phi}} \sum_{i=1}^{K} \|\phi(x_i^+) - \mathcal{A}\phi(x_i)\|_2^2$$

$$\phi(x) = [\phi_1(x), \dots, \phi_{N_\phi}(x)]^\top$$

linear operator

Predictor linear in u $\Rightarrow$ $\phi_i(x,\boldsymbol{u}) = \psi_i(x) + \mathcal{L}_i(\boldsymbol{u})$

Without loss of generality

$$\phi(x,\boldsymbol{u}) = [\psi_1(x), \dots, \psi_N(x), \boldsymbol{u}(0)^\top]^\top$$

$$\min_{A\in\mathbb{R}^{N\times N},\, B\in\mathbb{R}^{N\times m}} \sum_{i=1}^{K} \|\boldsymbol{\psi}(x_i^+) - A\boldsymbol{\psi}(x_i) - B\boldsymbol{u}_i(0)\|_2^2$$

Algorithm summary

Data $\mathbf{X} = \left[x_1, \ \ldots, x_M\right], \ \mathbf{X}^+ = \left[x_1^+, \ldots, x_M^+\right], \mathbf{U} = \left[u_1, \ldots, u_M\right]$

Embedding $\mathbf{X}_{\text{embed}} = \left[\psi(x_1), \ \ldots, \psi(x_M)\right], \ \mathbf{X}^+_{\text{embed}} = \left[\psi(x_1^+), \ldots, \psi(x_M^+)\right]$

LS problem $\min_{A,B} \|\mathbf{X}^+{}_{\text{embed}} - A\mathbf{X}_{\text{embed}} - B\mathbf{U}\|_F, \quad \min_C \|\mathbf{X} - C\mathbf{X}_{\text{embed}}\|_F$

Solution $[A, B] = \mathbf{X}^+_{\text{embed}}[\mathbf{X}_{\text{embed}}, \mathbf{U}]^\dagger, \quad C = \mathbf{X}\mathbf{X}^\dagger_{\text{embed}}$

$$z^+ = Az + Bu$$
$$\hat{x} = Cz$$
$$z_0 = \psi(x_0)$$

MPC design

Koopman MPC

Nonlinear MPC

$$
\begin{array}{lll}
\underset{u_i, x_i}{\text{minimize}} & \sum_{i=0}^{N_p-1} l_x(x_i) + u_i^\top R u_i + r^\top u_i & \\
\text{subject to} & x_{i+1} = f(x_i, u_i), & i = 0, \ldots, N_p - 1 \\
& c_x(x_i) + C_u u_i \le b, & i = 0, \ldots, N_p - 1 \\
\text{parameter} & x_0 = x &
\end{array}
$$

Nonconvex

x

$$\kappa(x) = \{u_0^\star, u_1^\star, \ldots, u_{N_p-1}^\star\}$$

$$x^+ = f(x, u)$$

Koopman MPC

Koopman MPC

$$
\begin{array}{lll}
\underset{u_i, z_i}{\text{minimize}} & \sum_{i=0}^{N_p-1} z_i^\top Q z_i + u_i^\top R u_i + q^\top z_i + r^\top u_i & \\
\text{subject to} & z_{i+1} = A z_i + B u_i, & i = 0, \ldots, N_p - 1 \\
& E z_i + F u_i \le b, & i = 0, \ldots, N_p - 1 \\
\text{parameter} & z_0 = \psi(x) &
\end{array}
$$

Convex

x

$$\kappa(x) = \{u_0^\star, u_1^\star, \ldots, u_{N_p-1}^\star\} \longrightarrow \quad x^+ = f(x, u)$$

Can handle **nonlinear constraints** and **costs** in a linear fashion

Koopman MPC

Dense-form Koopman MPC

$$\begin{array}{ll} \underset{\mathbf{u}\in\mathbb{R}^{mN_p}}{\text{minimize}} & \mathbf{u}^\top H \mathbf{u}^\top + h^\top \mathbf{u} + z_0^\top G \mathbf{u} \\ \text{subject to} & L\mathbf{u} + M z_0 \leq c \end{array}$$

parameter $z_0 = \psi(x)$

$$\kappa(x) = \begin{bmatrix} u_0^\star \\ \vdots \\ u_{N_p-1}^\star \end{bmatrix}$$

$$x^+ = f(x, u)$$

x

Computation cost **independent** of the size of the embedding!

Koopman MPC summary

At each step k of closed-loop operation

- Set $z_0 = \boldsymbol{\psi}(x_k)$

- Solve

$$\begin{array}{ll} \underset{\mathbf{u}\in\mathbb{R}^{mN_p}}{\text{minimize}} & \mathbf{u}^\top H \mathbf{u}^\top + h^\top \mathbf{u} + z_0^\top G \mathbf{u} \\ \text{subject to} & L\mathbf{u} + M z_0 \leq c \end{array} \quad \Rightarrow \quad \mathbf{u}^\star = \begin{bmatrix} u_0^\star \\ \vdots \\ u_{N_p-1}^\star \end{bmatrix}$$

- Apply $u_0^\star$ to the system

Koopman MPC summary

At each step k of closed-loop operation

- Set $z_0 = \boldsymbol{\psi}(x_k)$

- Solve

$$\begin{array}{ll} \underset{\mathbf{u}\in\mathbb{R}^{mN_p}}{\text{minimize}} & \mathbf{u}^\top H \mathbf{u}^\top + h^\top \mathbf{u} + z_0^\top G \mathbf{u} \\ \text{subject to} & L\mathbf{u} + M z_0 \leq c \end{array} \quad \Rightarrow \quad \mathbf{u}^\star = \begin{bmatrix} u_0^\star \\ \vdots \\ u_{N_p-1}^\star \end{bmatrix}$$

- Apply $u_0^\star$ to the system

Main benefits

Data-driven: No model required

Fast & **simple**: only small **convex quadratic program** solved online

Nonlinear constraints and **costs** handled in a linear fashion

Extensions

- Input-output systems

$$x^+ = f(x, u)$$
$$y = h(x)$$

Solution: Use nonlinear functions of y and its time-delays as basis functions

(cf. Takens theorem, system id)

- Systems with disturbances

$$x^+ = f(x, u, w)$$

Solution: Treat w as an additional input

Numerical examples

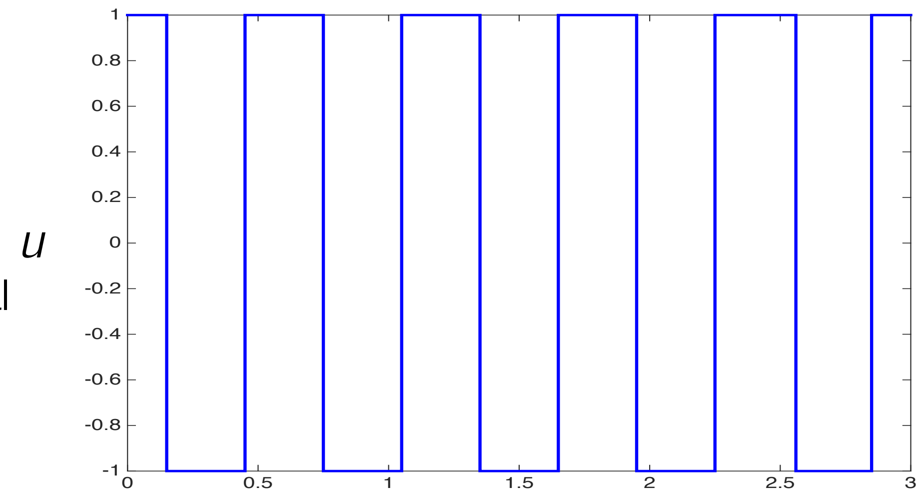
Van der Pol oscillator

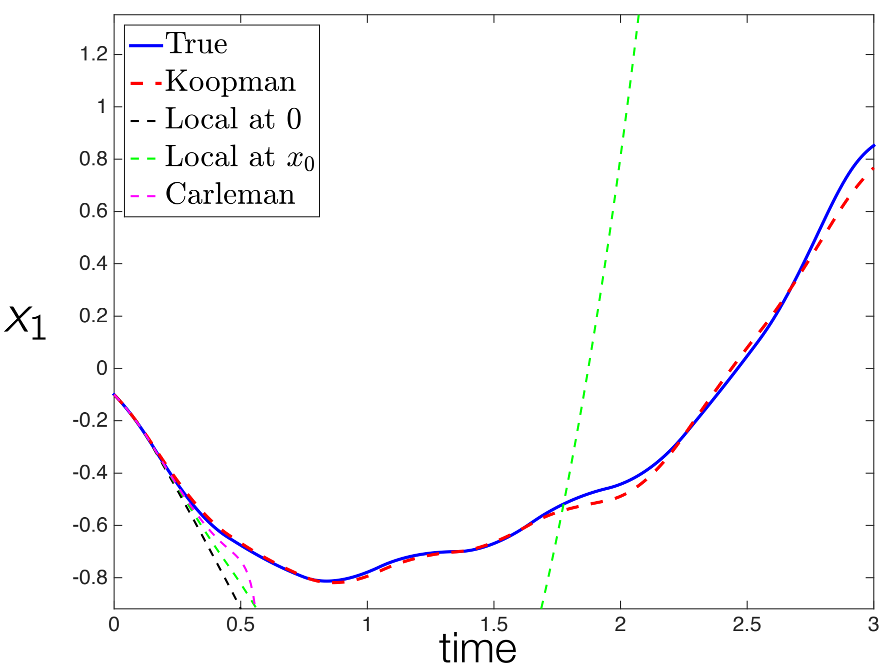
$$\dot{x}_1 = 2x_2$$
$$\dot{x}_2 = -0.8x_1 + 2x_2 - 10x_1^2 x_2 + u$$

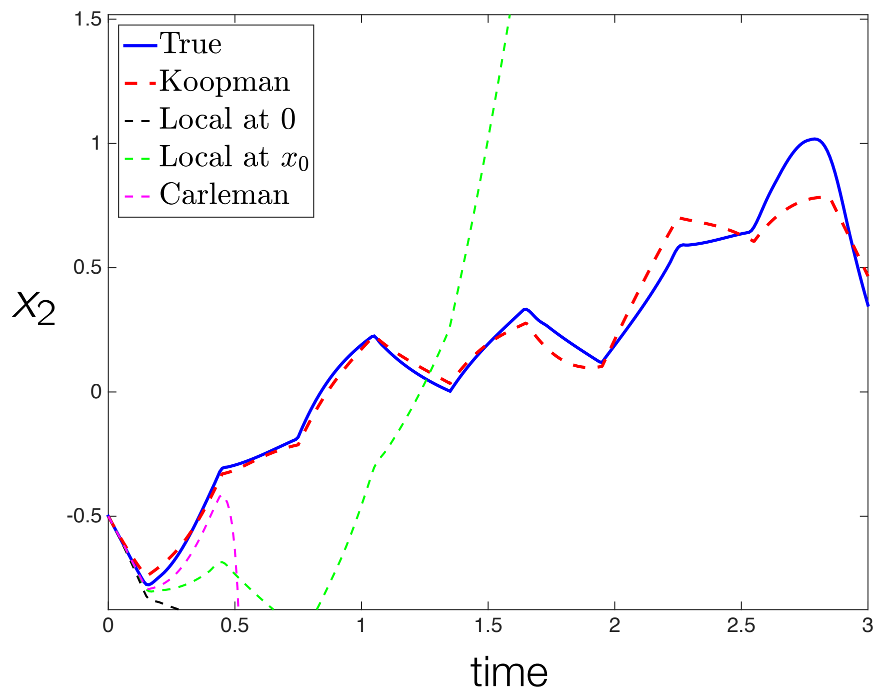
RK-4 discretization with 0.01 s sampling interval

Data: 20 trajectories with 1000 samples each

Lifting: state observable + 100 RBFs







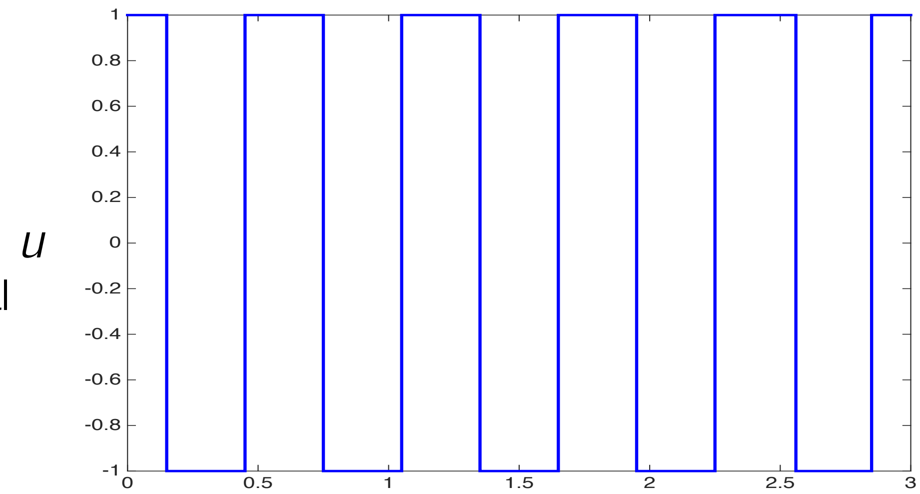
Van der Pol oscillator

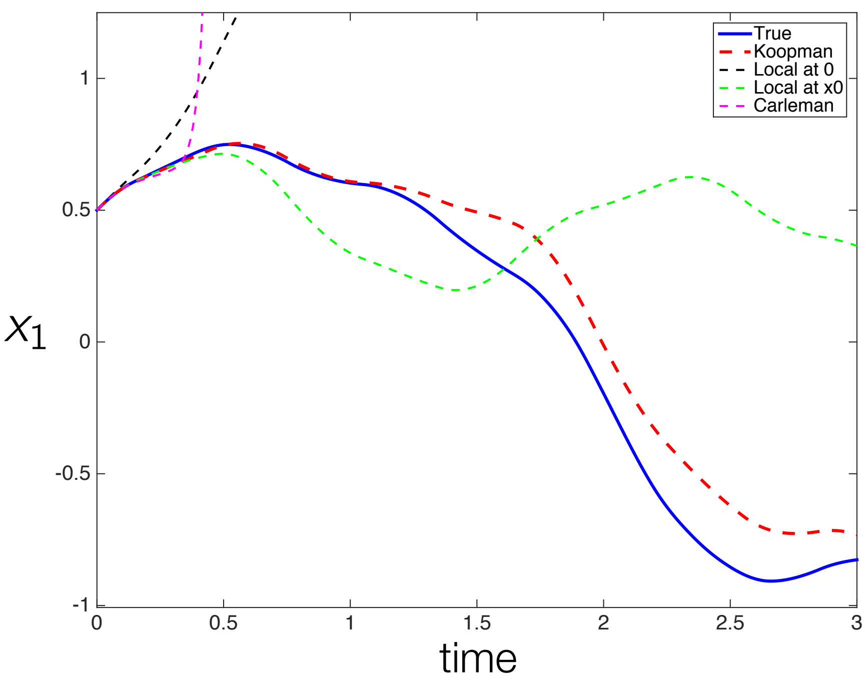
$$\dot{x}_1 = 2x_2$$
$$\dot{x}_2 = -0.8x_1 + 2x_2 - 10x_1^2 x_2 + u$$

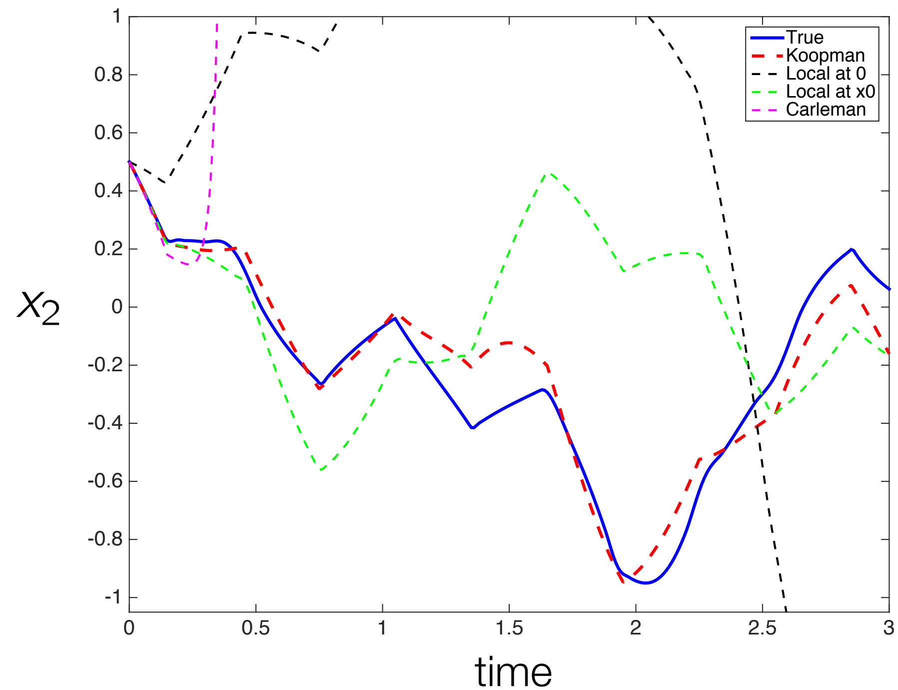
RK-4 discretization with 0.01 s sampling interval

Data: 20 trajectories with 1000 samples each

Lifting: state observable + 100 RBFs







Van der Pol oscillator

$$\dot{x}_1 = 2x_2$$
$$\dot{x}_2 = -0.8x_1 + 2x_2 - 10x_1^2 x_2 + u$$

$$\text{RMSE [\%]} = 100 \cdot \frac{\|x_{\text{true}} - x_{\text{pred}}\|}{\|x_{\text{true}}\|}$$

RK-4 discretization with 0.01 s sampling interval

Data: 20 trajectories with 1000 samples each

Lifting: state observable + 100 RBFs

x_0	Average RMSE
Koopman	24.4 %
Local linearization at x_0	$2.83 \cdot 10^3$ %
Local linearization at 0	912.5 %
Carleman	$5.08 \cdot 10^{22}$ %

Van der Pol oscillator

$$\dot{x}_1 = 2x_2$$
$$\dot{x}_2 = -0.8x_1 + 2x_2 - 10x_1^2 x_2 + u$$

$$\text{RMSE}\,[\%] = 100 \cdot \frac{\|x_{\text{true}} - x_{\text{pred}}\|}{\|x_{\text{true}}\|}$$

RK-4 discretization with 0.01 s sampling interval

Data: 20 trajectories with 1000 samples each

Lifting: state observable + 100 RBFs

N	5	10	25	50	75	100
Average RMSE	66.5 %	44.9 %	47.0 %	38.7 %	30.6 %	24.4 %

Power grid stabilization

Join work with Yoshi Susuki

Problem setup

New England power grid model

$$\dot{\delta}_i = \omega_i$$

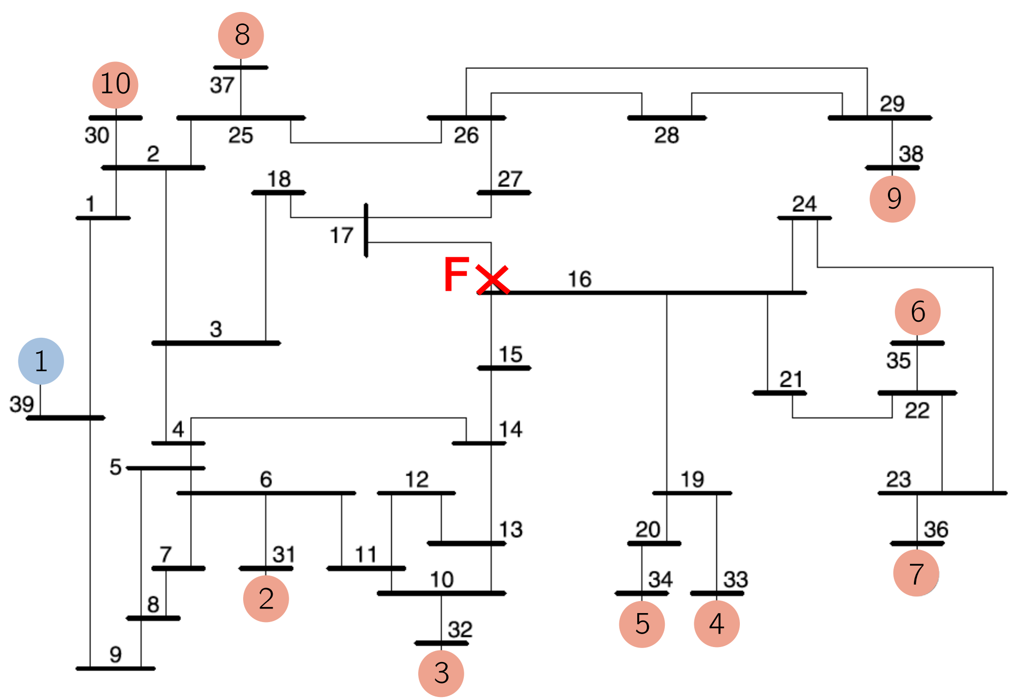
$$\frac{H_i}{\pi f_\mathrm{b}}\dot{\omega}_i = -D_i\omega_i + P_{\mathrm{m}_i}$$

$$-G_{ii}V_i^2 - \sum_{j=1, j\neq i}^{10} V_iV_j\{G_{ij}\cos(\delta_i - \delta_j) + B_{ij}\sin(\delta_i - \delta_j)\}$$

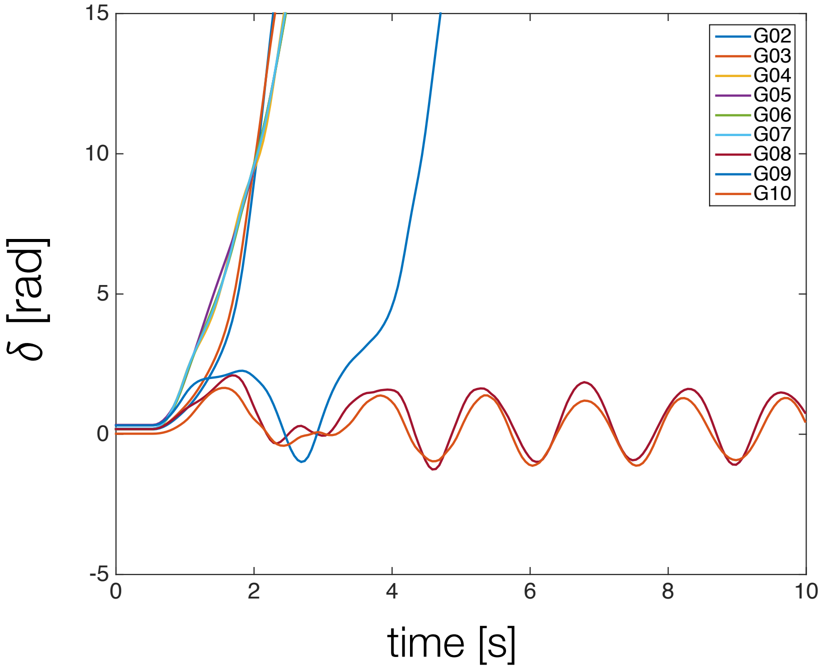
Setup from *[Susuki et al, 2011]*

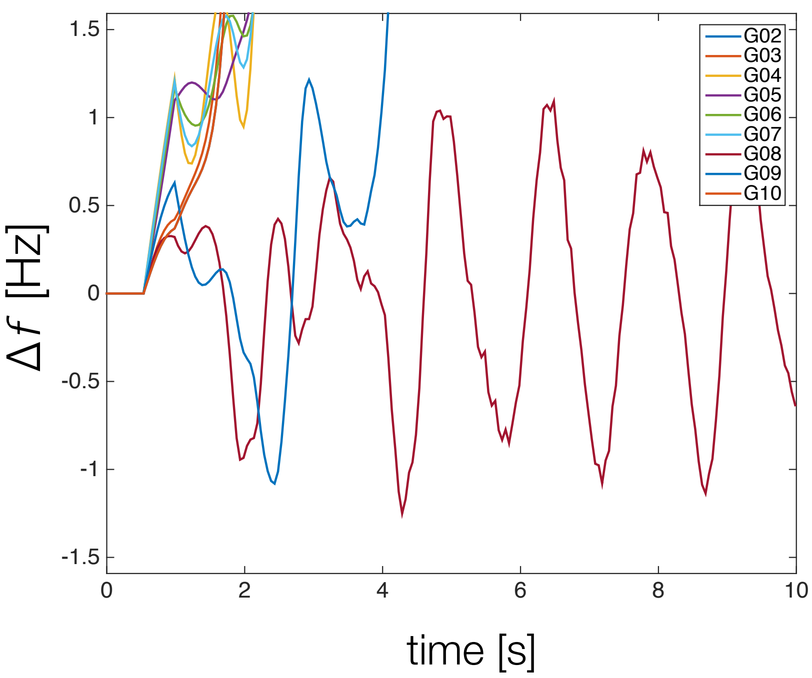
$t = 0.67\,\mathrm{s}$ – fault occurs

$t = 1\,\mathrm{s}$ – faulted line removed



Fault causes instability





Setting up Koopman MPC

New England power grid model

$$\dot{\delta}_i = \omega_i$$

$$\frac{H_i}{\pi f_\mathrm{b}}\dot{\omega}_i = -D_i\omega_i + P_{\mathrm{m}_i} - G_{ii}V_i^2 - \sum_{j=1, j\neq i}^{10} V_i V_j \{G_{ij}\cos(\delta_i - \delta_j) + B_{ij}\sin(\delta_i - \delta_j)\}$$

Actuation: P_{m_i} mechanical power

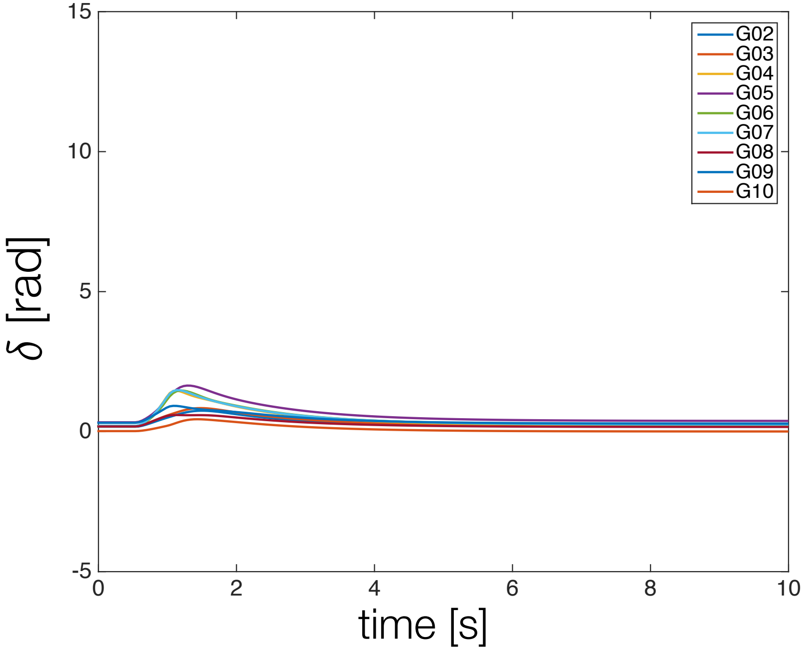
Cost: $\sum_i \omega_i^2$ – frequency deviation

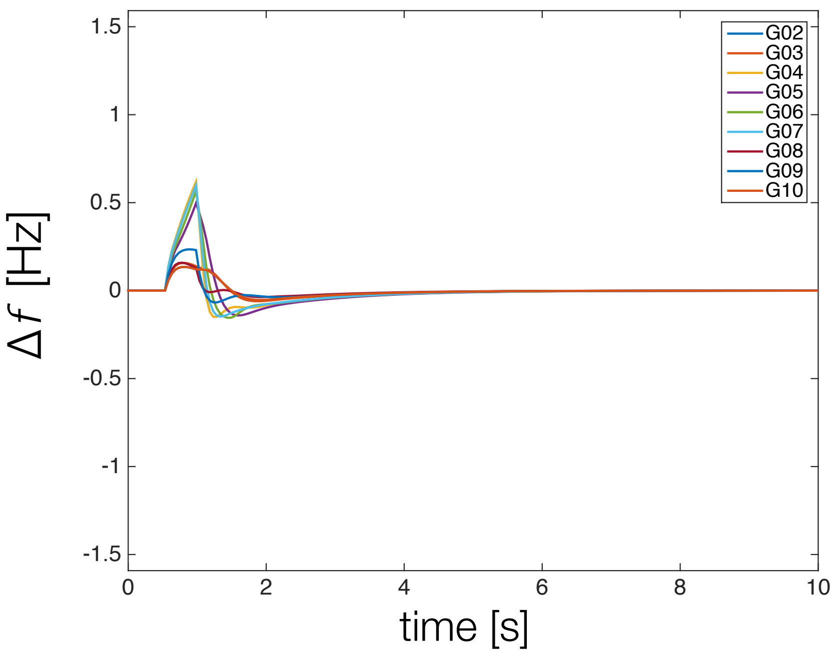
Pred. horizon: 1 second

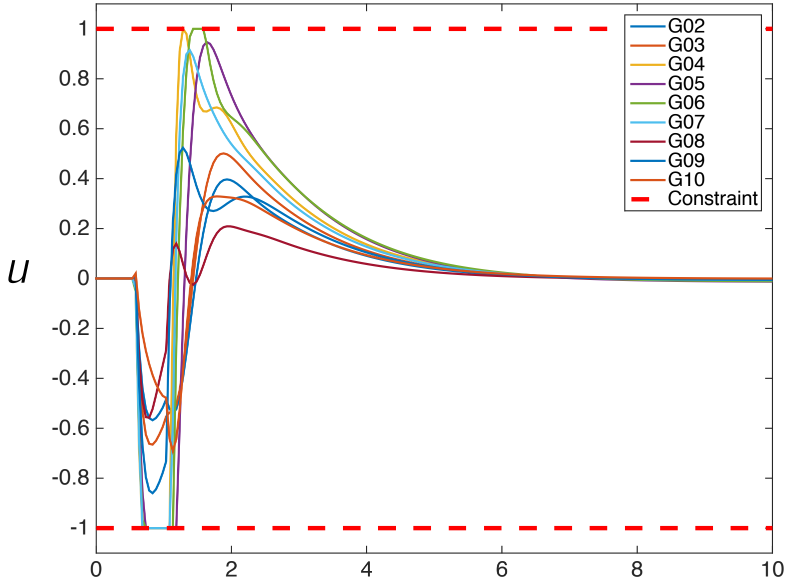
Sampling: 50 ms

Embedding: $\psi = \begin{bmatrix} \cos(\delta) \\ \sin(\delta) \\ \omega \end{bmatrix}$ $\qquad \psi : \mathbb{R}^{18} \to \mathbb{R}^{27}$

Instability suppression





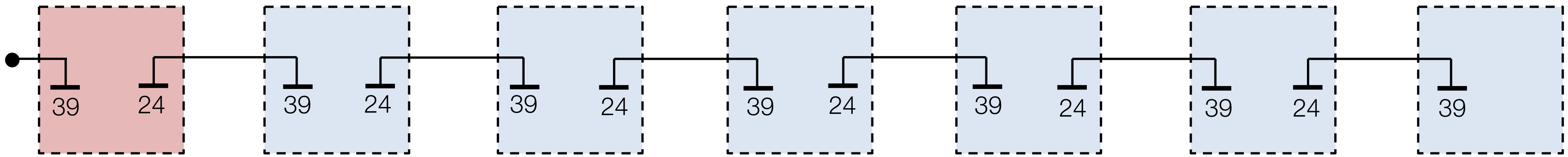


NE grid cascade

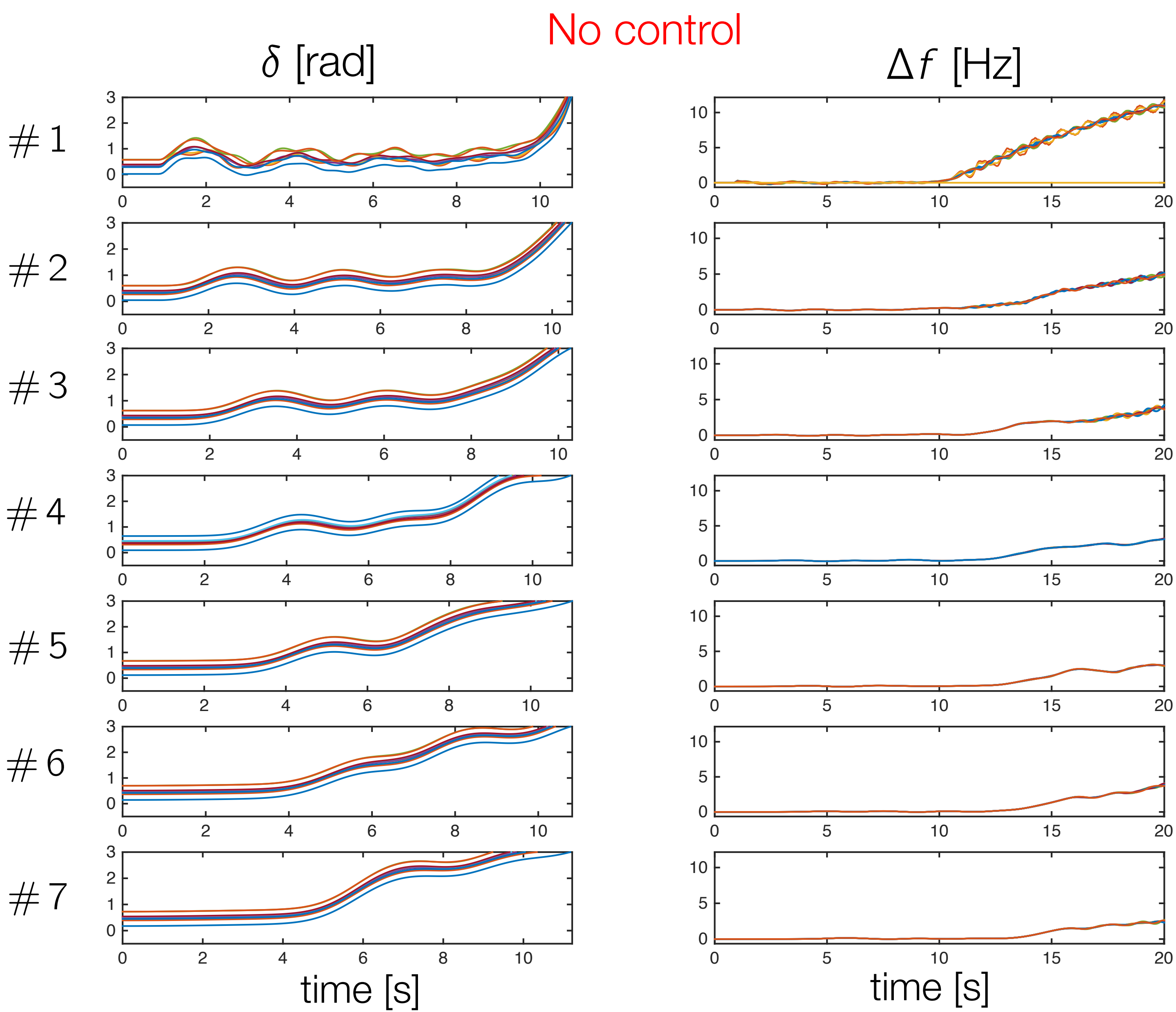
$t = 0.87\,\text{s}$ – fault occurs in grid $\#1$

$t = 1\,\text{s}$ – faulted line removed

Setup from *[Susuki et al, 2012]*

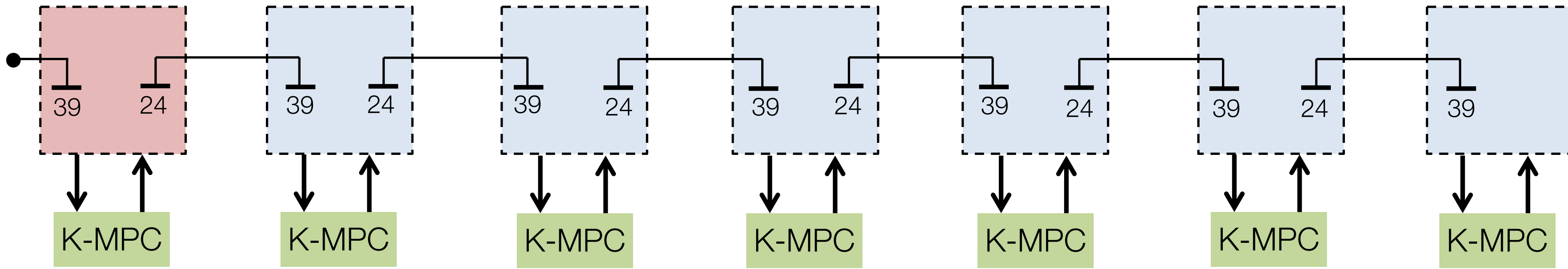


Cascade instability occurs without control

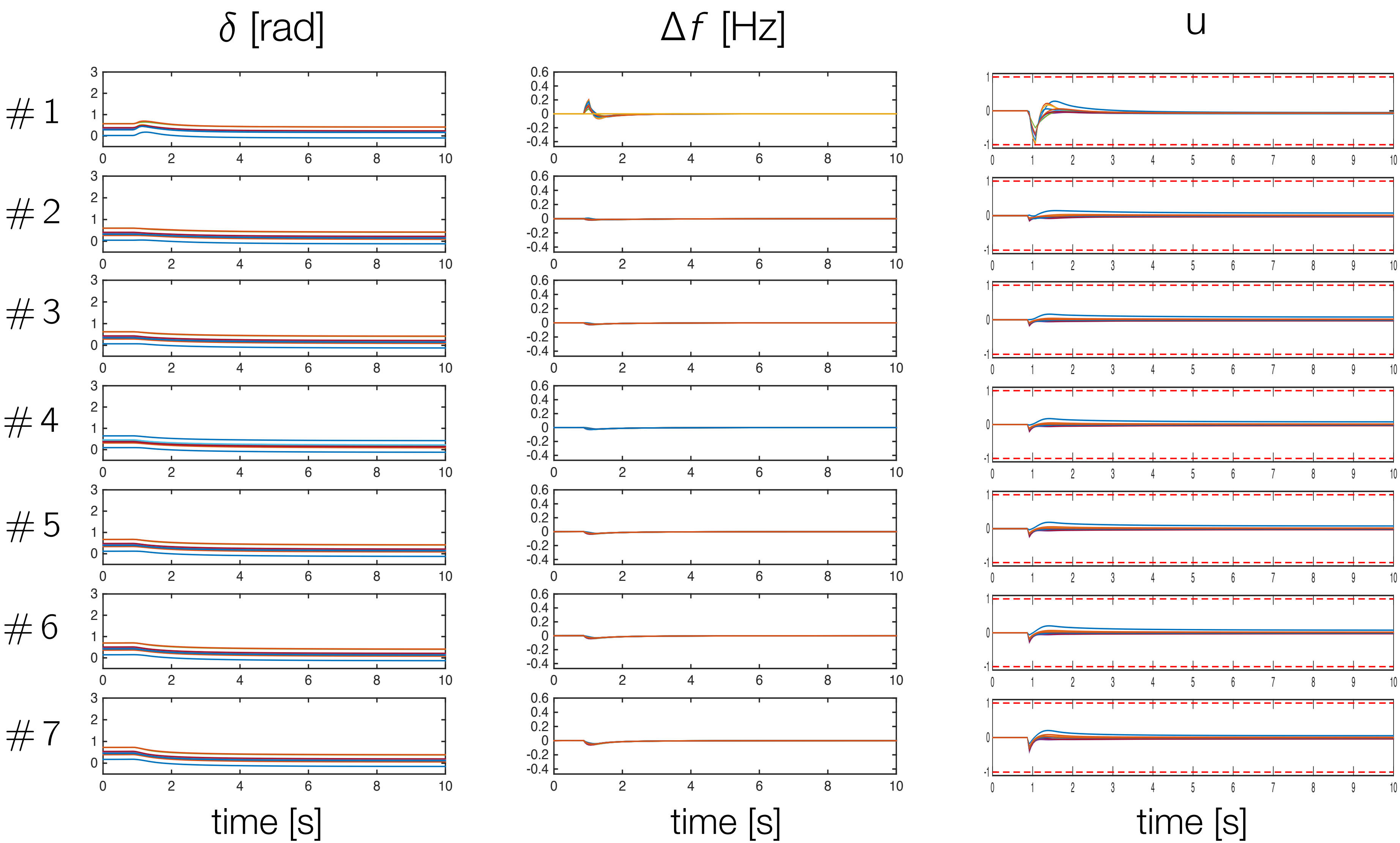


Can we suppress cascade instability?

Case 1: Each grid controlled separately

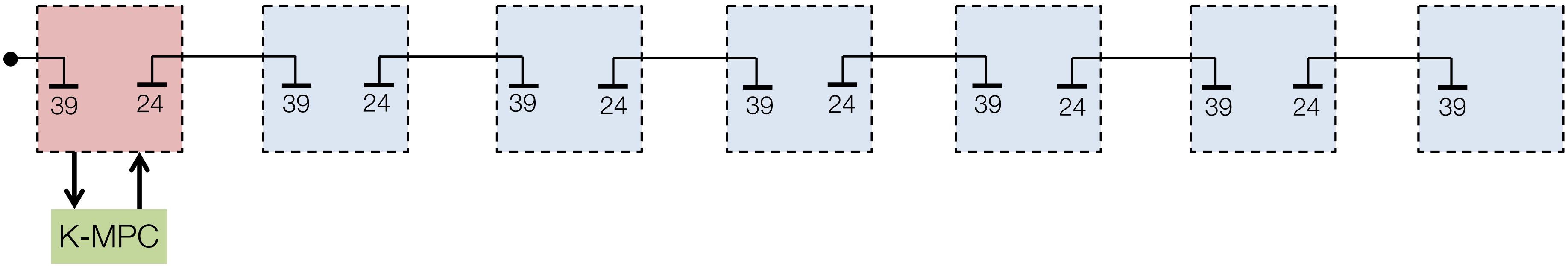


Koopman MPC suppresses the instability

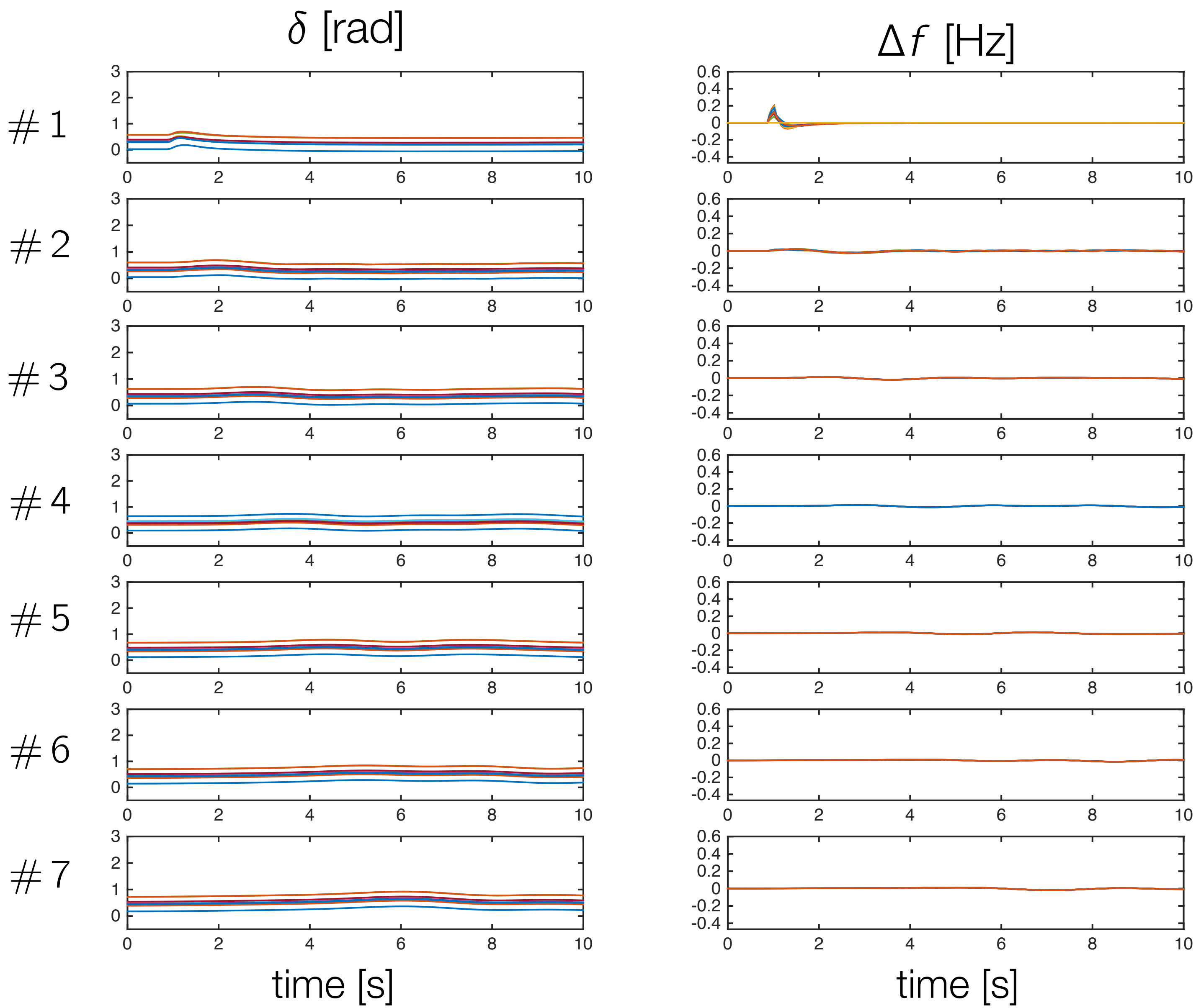


What if only the first grid is controlled?

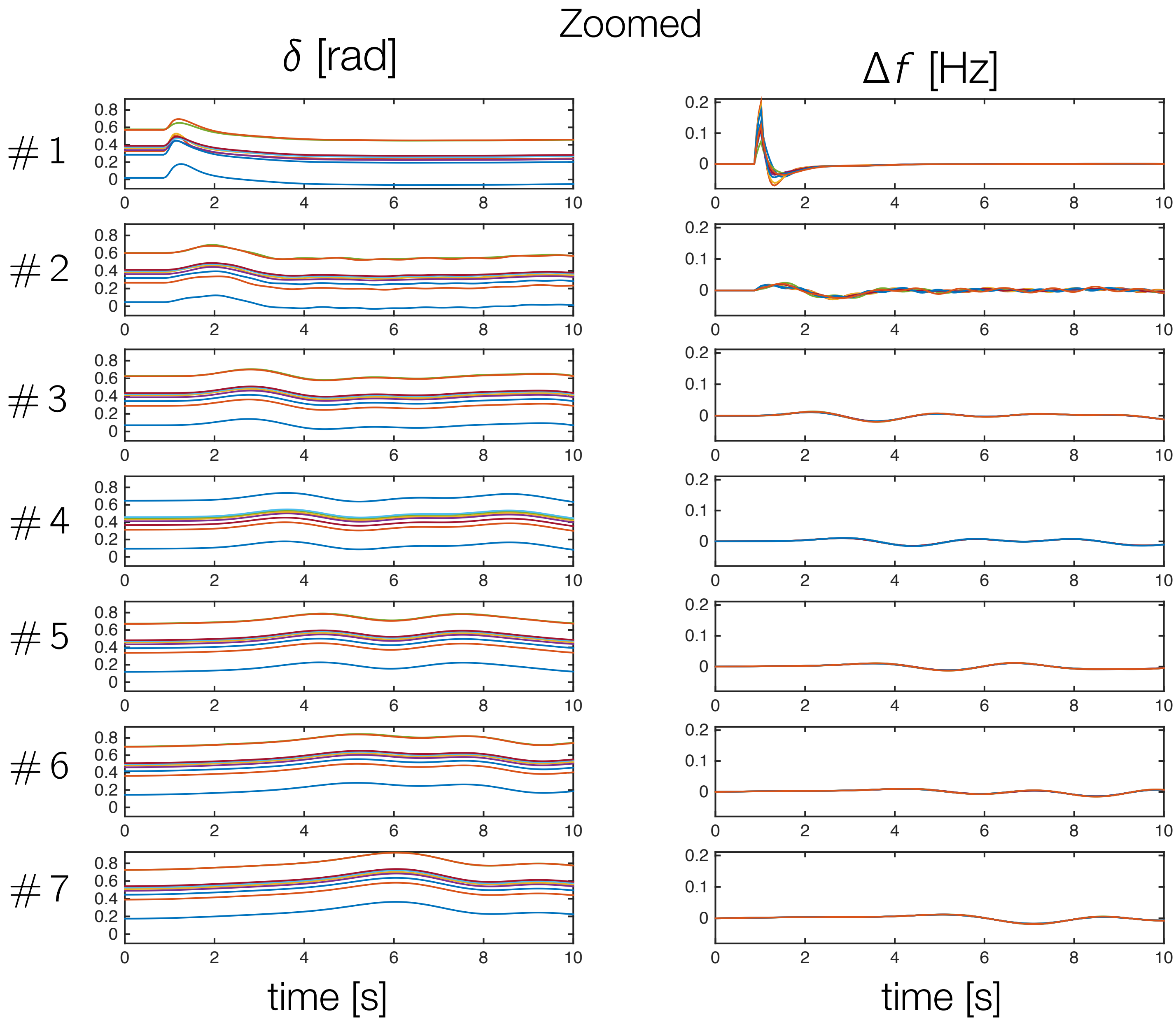
Case 2: Only the grid where the fault occurred controlled



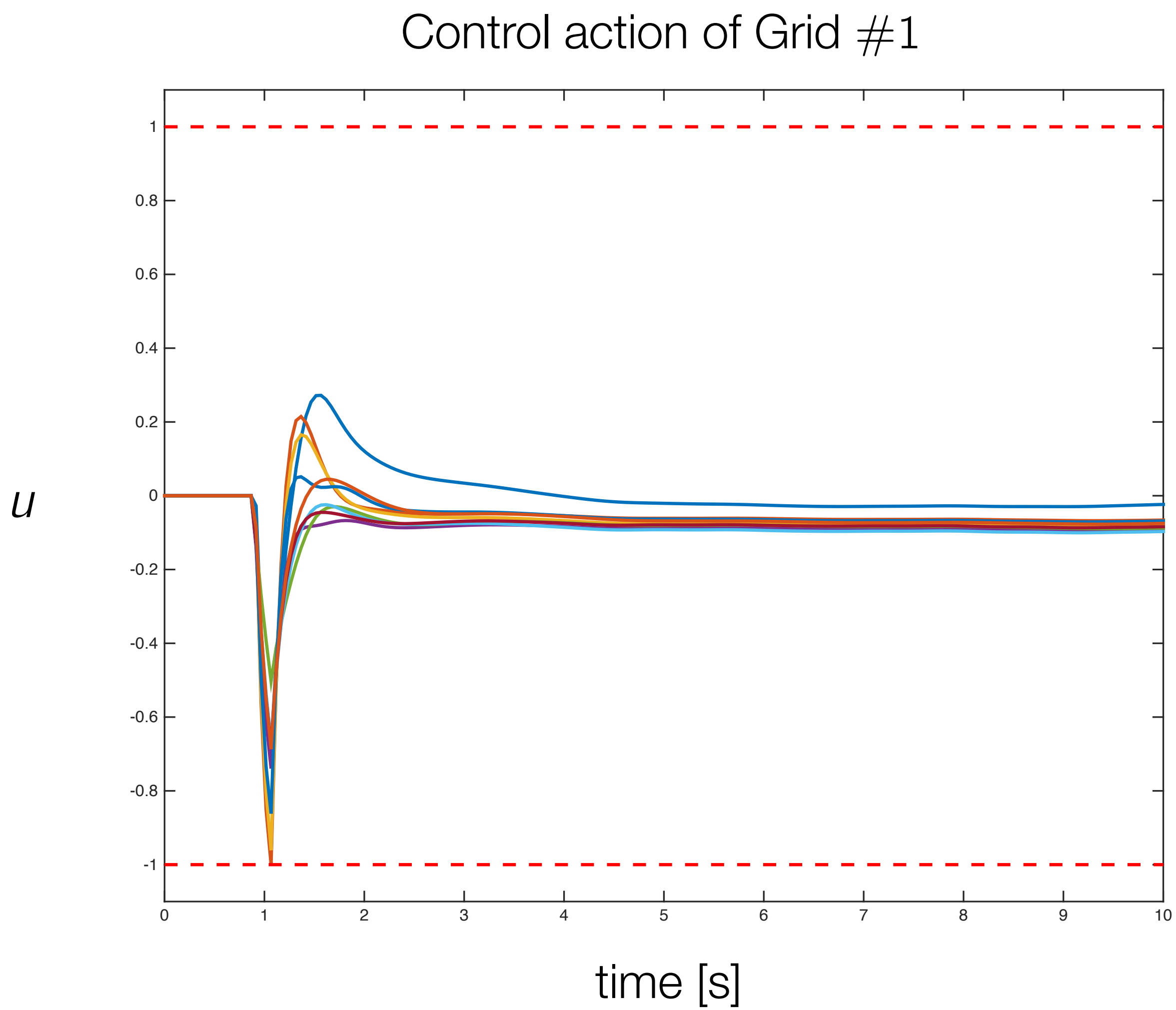
Even one grid control suppresses the instability



Even one grid control suppresses the instability



Control input



Numerical examples - NE cascade

Computation time $\approx$ 10ms per grid

(Matlab + qpOASES, 2GHz i7)

PDE control

Joint work with Hassan Arbabi

Burgers' equation

$$\frac{\partial y}{\partial t} - \nu \frac{\partial^2 y}{\partial x^2} + y \frac{\partial y}{\partial x} = u(t, x)$$

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

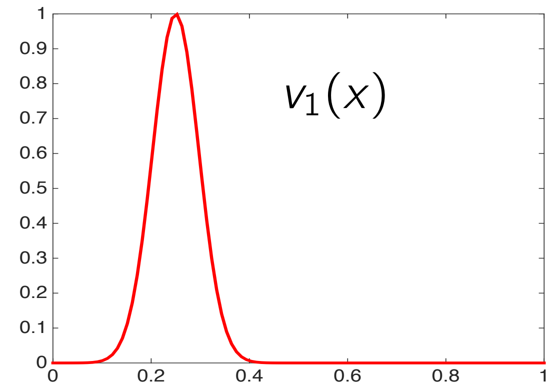
Burgers' equation

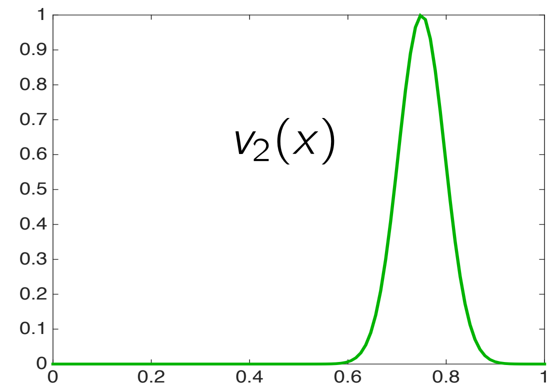
$$\frac{\partial y}{\partial t} - \nu \frac{\partial^2 y}{\partial x^2} + y \frac{\partial y}{\partial x} = u(t,x)$$

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

Setup from *[Peitz, Klus 2017]*

$$u(t,x) = u_1(t) v_1(x) + u_2(t) v_2(x)$$





Burgers' equation

$$\frac{\partial y}{\partial t} - \nu \frac{\partial^2 y}{\partial x^2} + y \frac{\partial y}{\partial x} = u(t,x)$$

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

$$u(t,x) = u_1(t) v_1(x) + u_2(t) v_2(x)$$

$$|u_1(t)| \le 0.1 \ , \quad |u_2(t)| \le 0.1$$

Tracking piecewise-constant reference

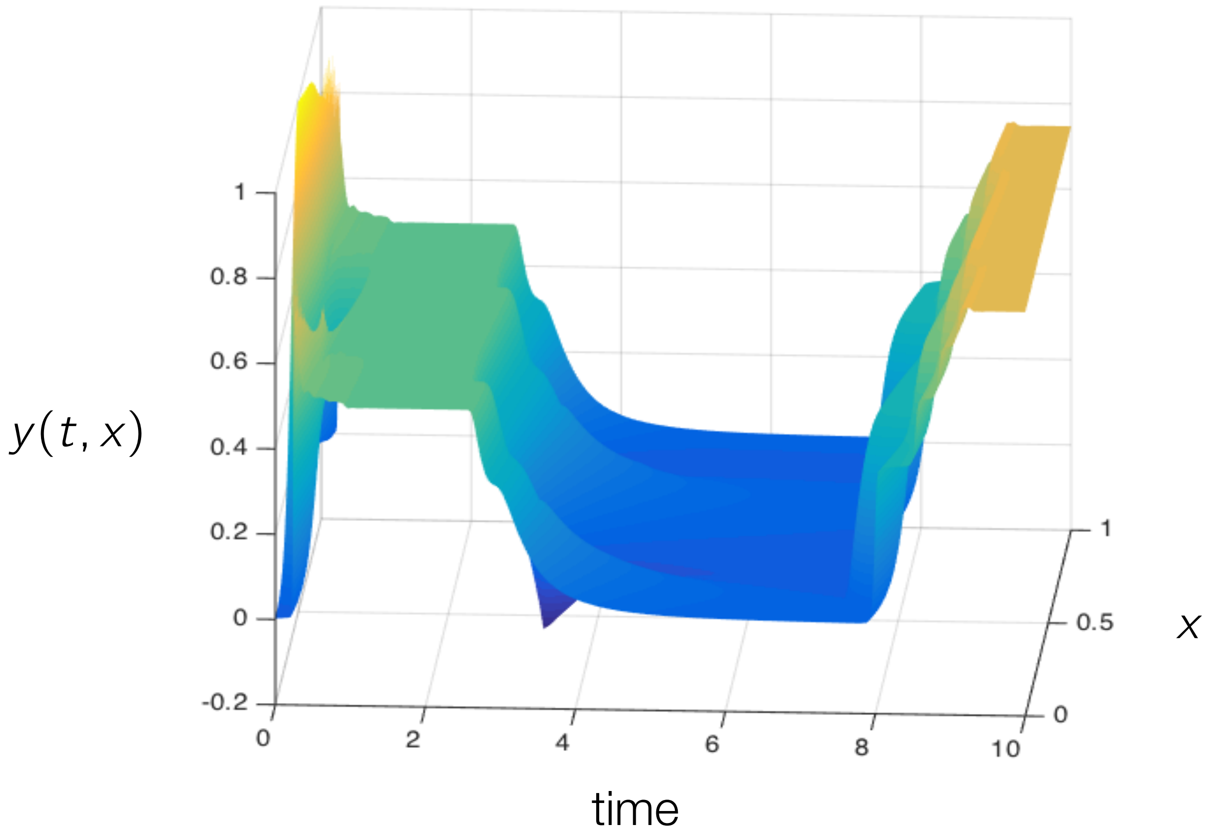
Burgers' equation

$$\frac{\partial y}{\partial t} - \nu \frac{\partial^2 y}{\partial x^2} + y \frac{\partial y}{\partial x} = u(t,x)$$

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

$$u(t,x) = u_1(t)v_1(x) + u_2(t)v_2(x)$$

$$|u_1(t)| \le 0.1 \;, \quad |u_2(t)| \le 0.1$$



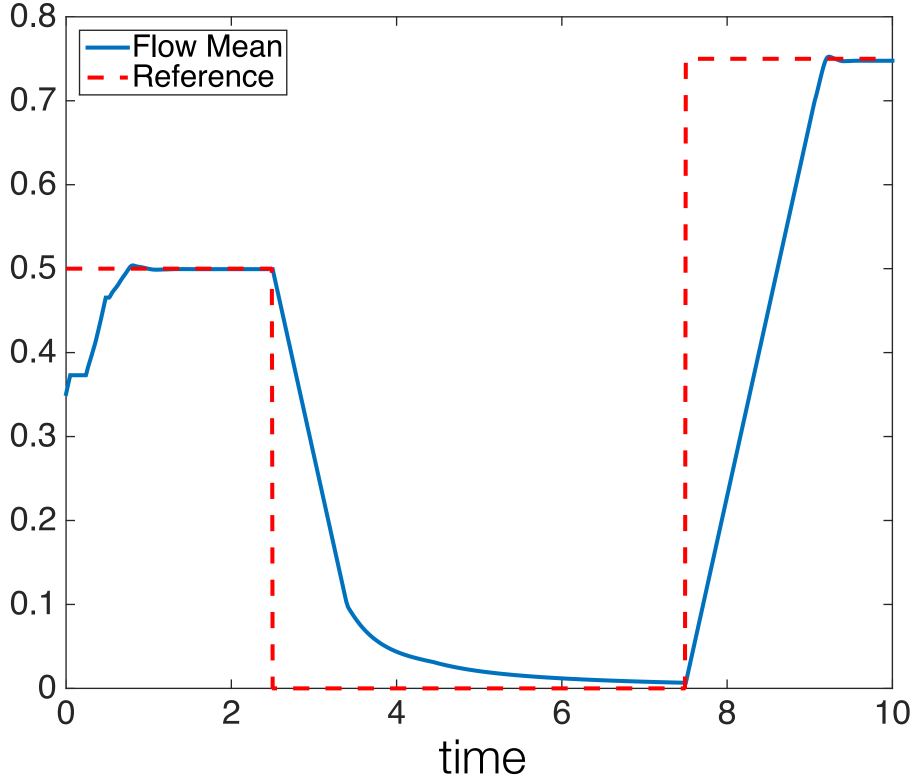
Burgers' equation

$$\frac{\partial y}{\partial t} - \nu \frac{\partial^2 y}{\partial x^2} + y \frac{\partial y}{\partial x} = u(t, x)$$

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

$$u(t, x) = u_1(t) v_1(x) + u_2(t) v_2(x)$$

$$|u_1(t)| \leq 0.1 \;, \quad |u_2(t)| \leq 0.1$$



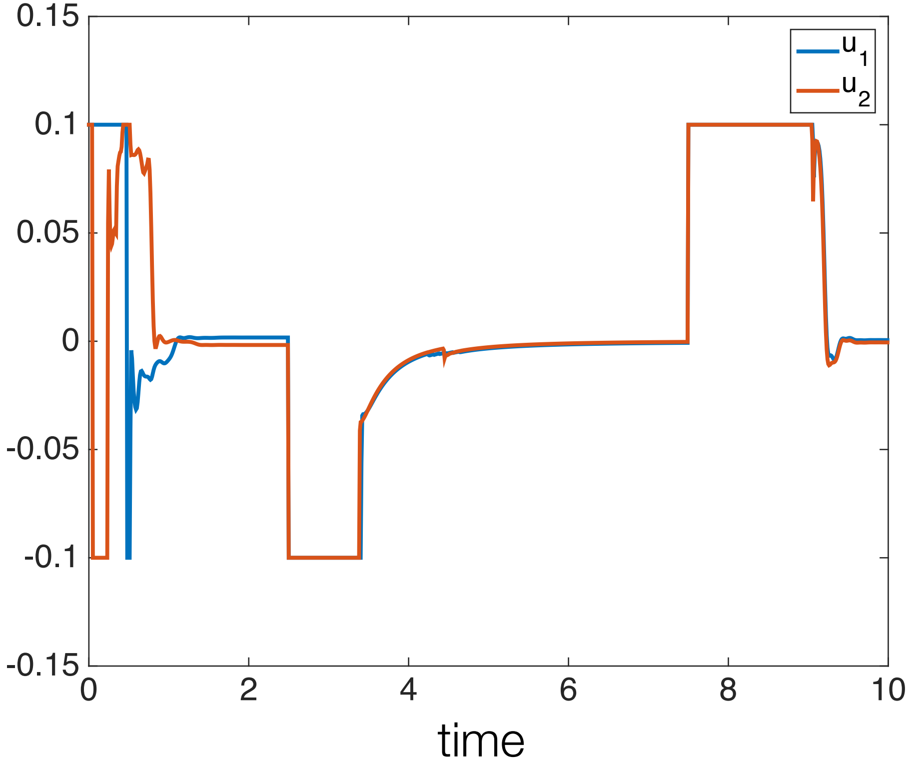
Burgers' equation

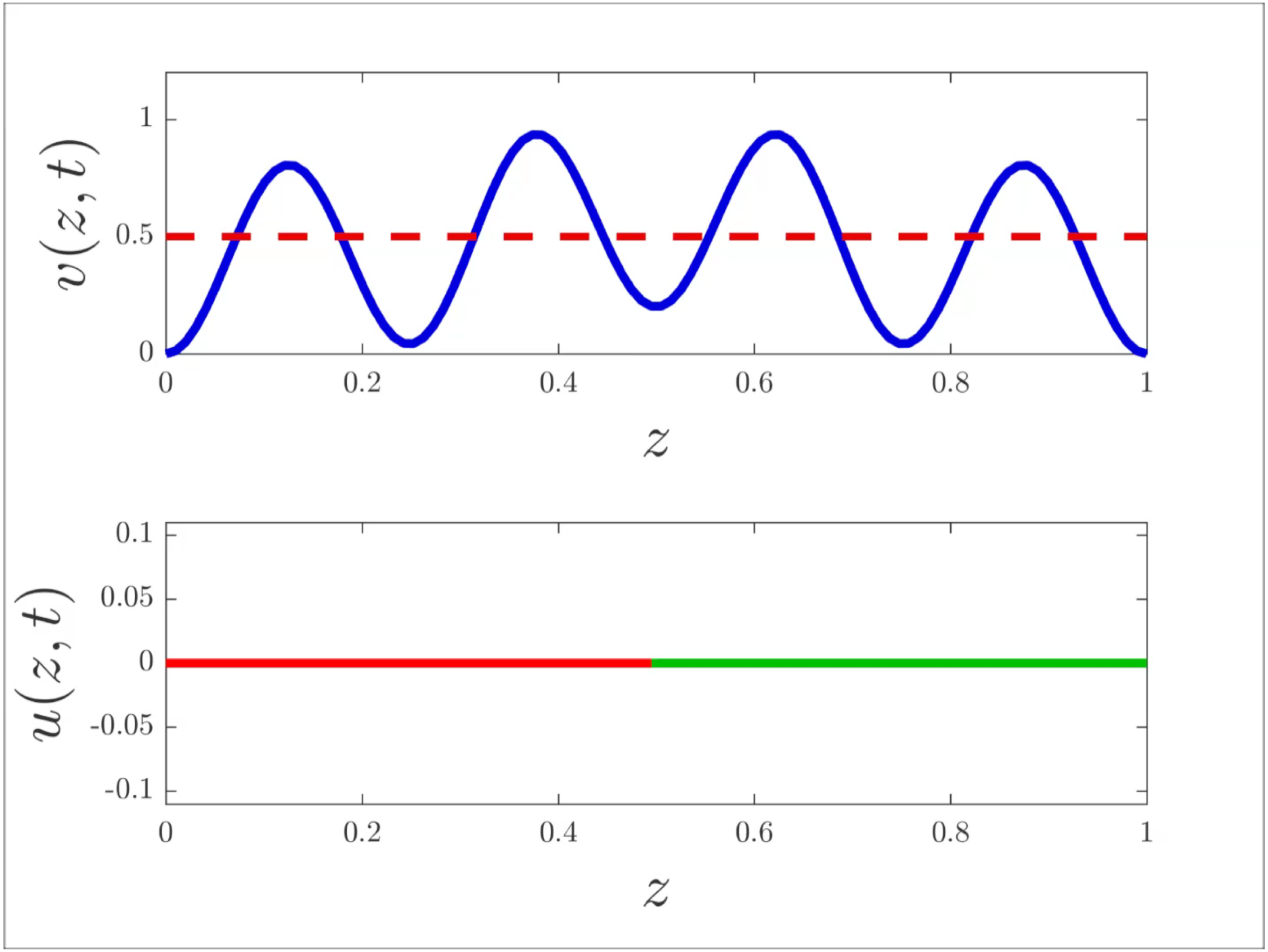
$$\frac{\partial y}{\partial t} - \nu \frac{\partial^2 y}{\partial x^2} + y \frac{\partial y}{\partial x} = u(t,x)$$

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

$$u(t,x) = u_1(t) v_1(x) + u_2(t) v_2(x)$$

$$|u_1(t)| \leq 0.1 \ , \quad |u_2(t)| \leq 0.1$$

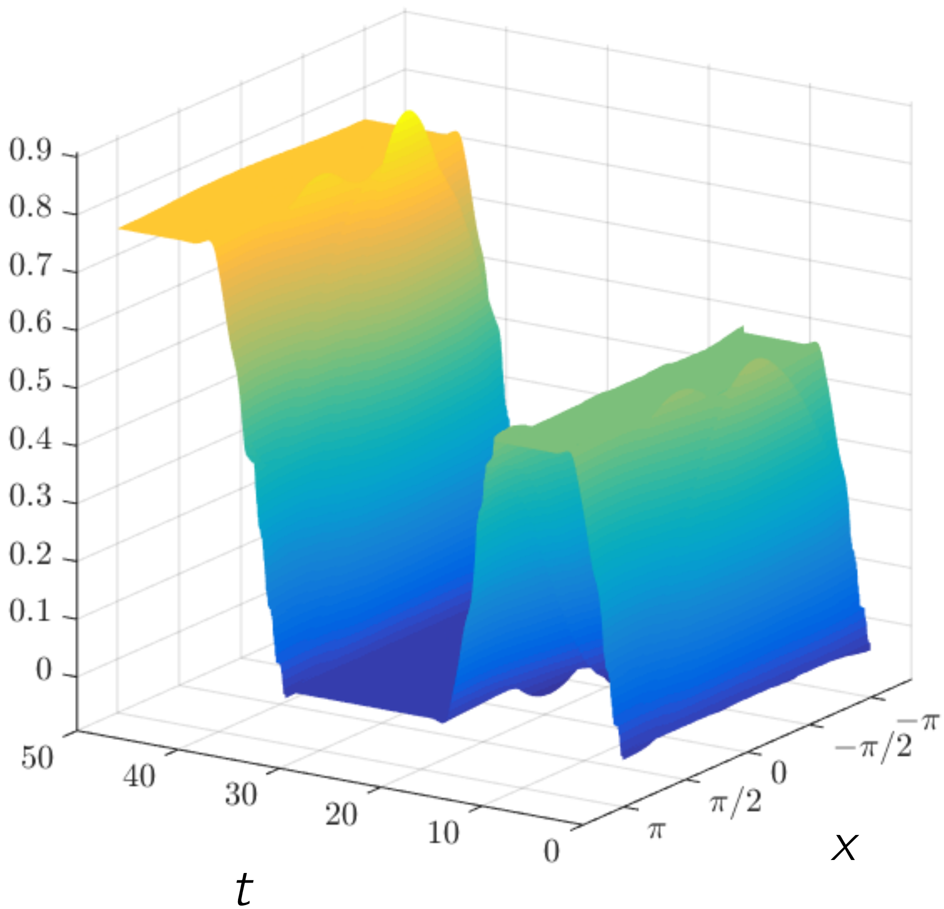




1
0.5
0
$v(z,t)$
0
0.2
0.4
0.6
0.8
1
z
0.1
0.05
0
-0.05
-0.1
$u(z,t)$
0
0.2
0.4
0.6
0.8
1
z

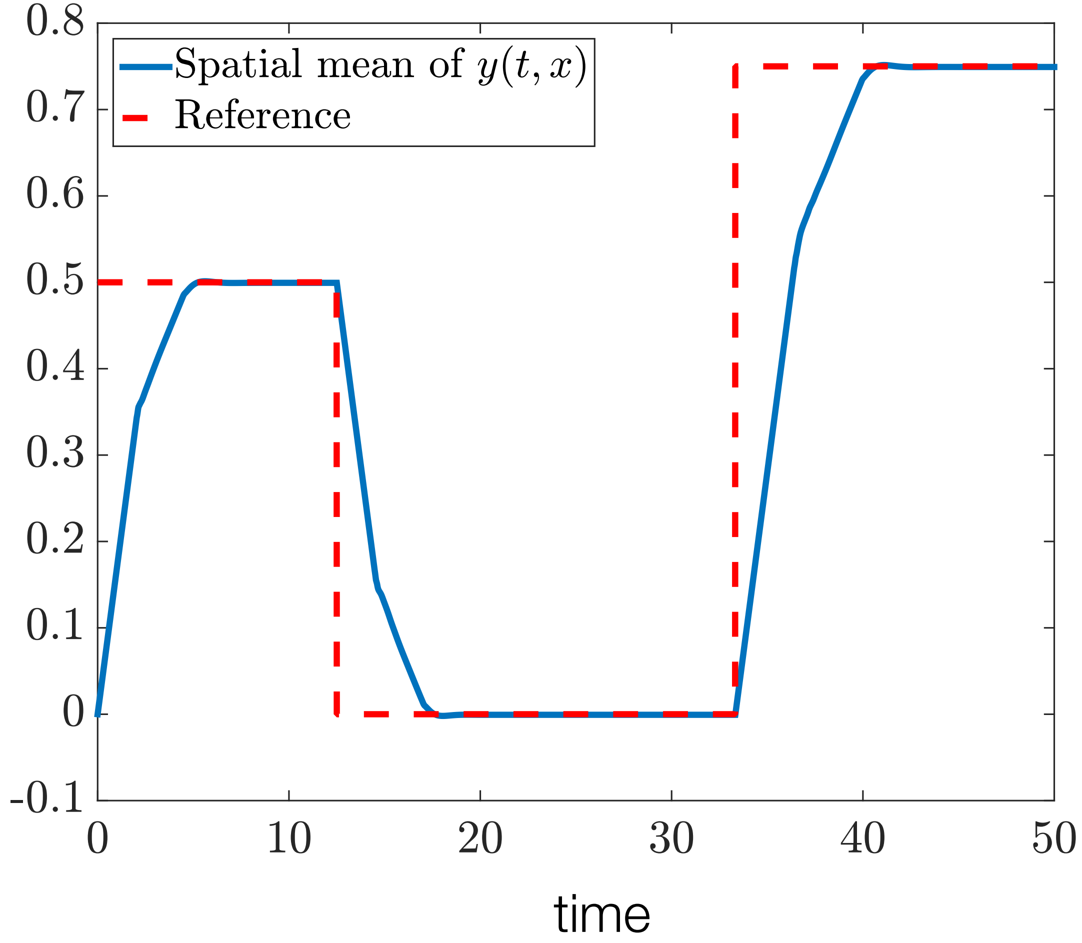
Korteweg–de Vries equation

Similar control setup as for Burgers’



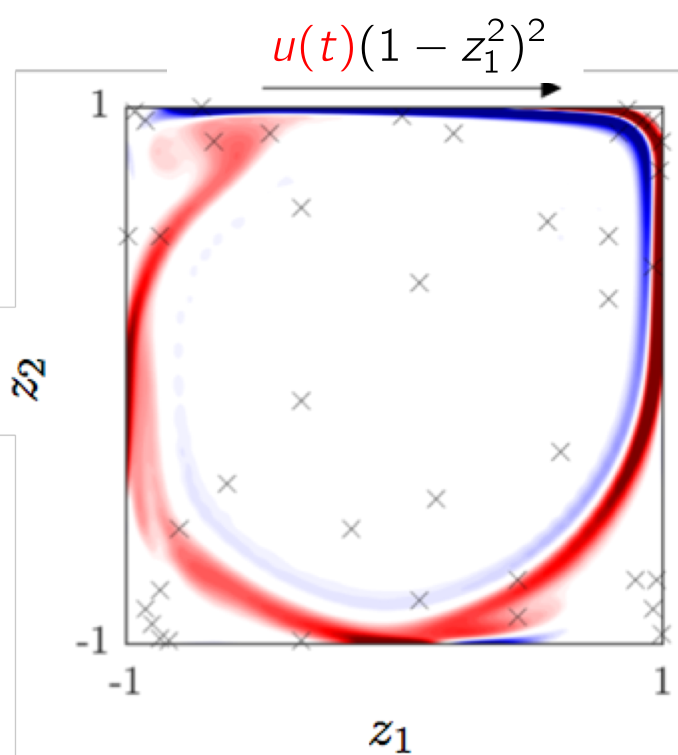
Korteweg–de Vries equation

Similar control setup as for Burgers'



Cavity flow – problem setup

$$\frac{\partial v}{\partial t} + v \cdot \nabla v = -\nabla p + \frac{1}{Re}\nabla^2 v, \quad \nabla \cdot v = 0$$



Control input: Amplitude of top lid velocity

Measurements: Velocity at randomly selected points

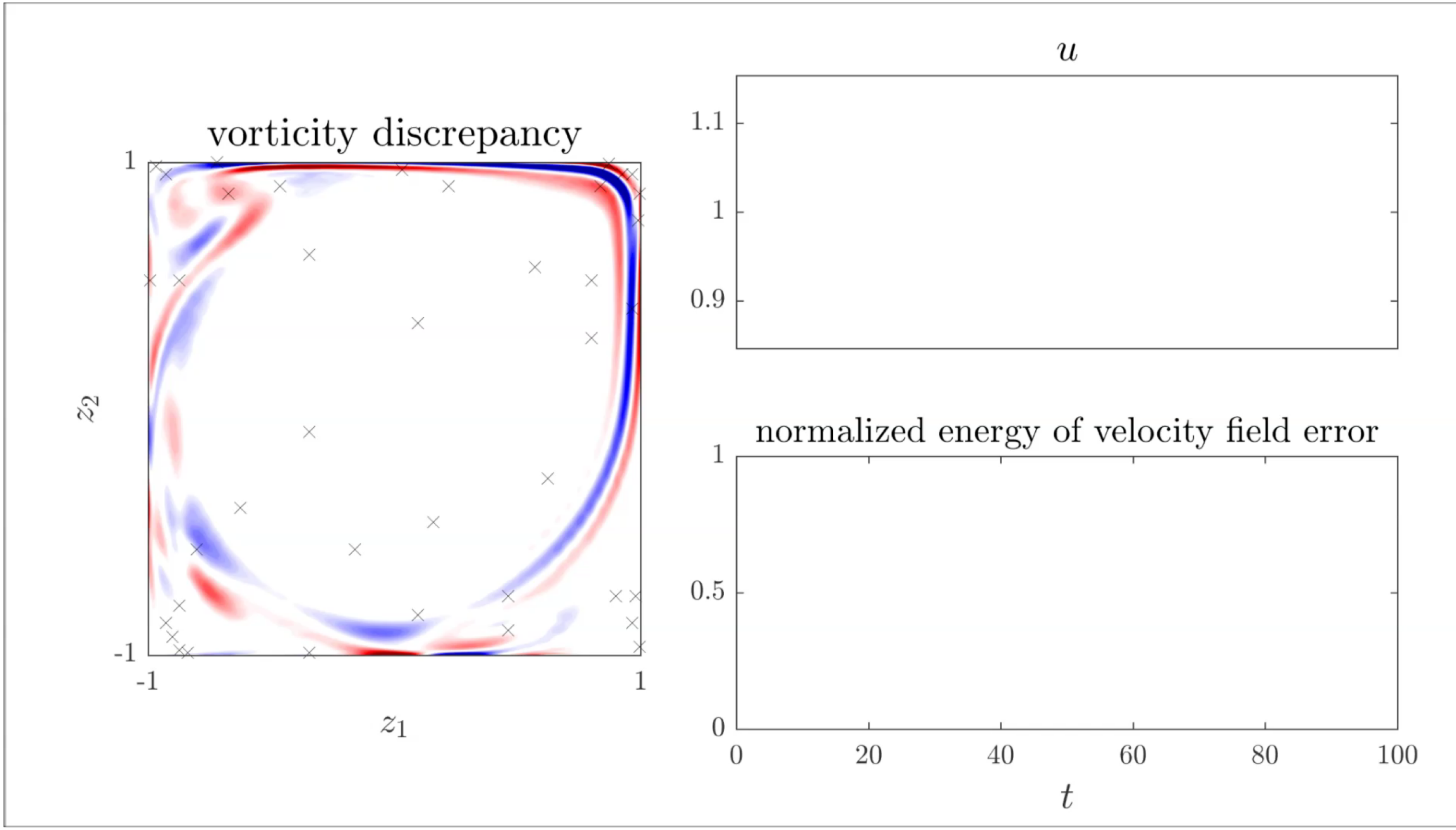
Training data: 300 two-second-long trajectories

Control task: Re 13k (limit cycle) → Re 10k (stable fixed point)

Re 13k (limit cycle) → mean flow (unstable fixed point)

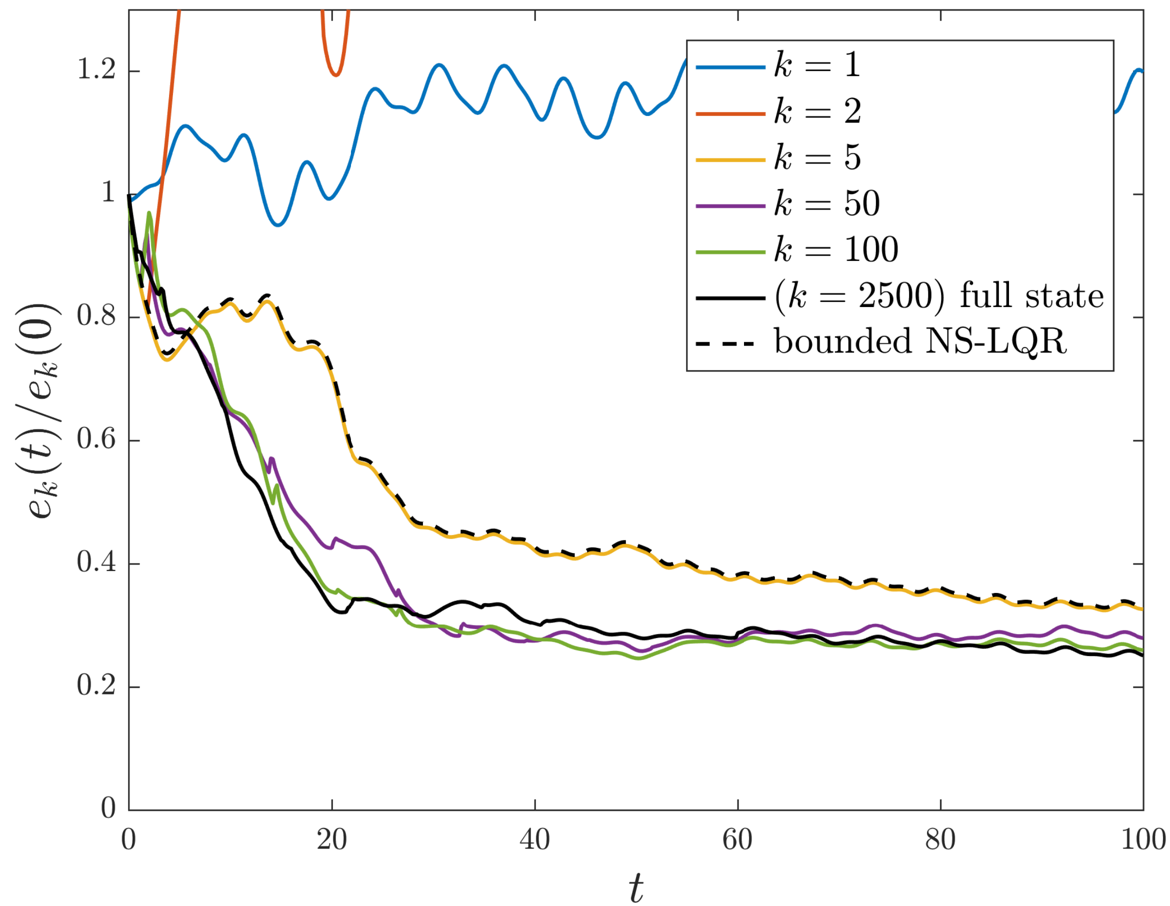
Control performance

Control task: Re 13k (limit cycle) → Re 10k (stable fixed point)



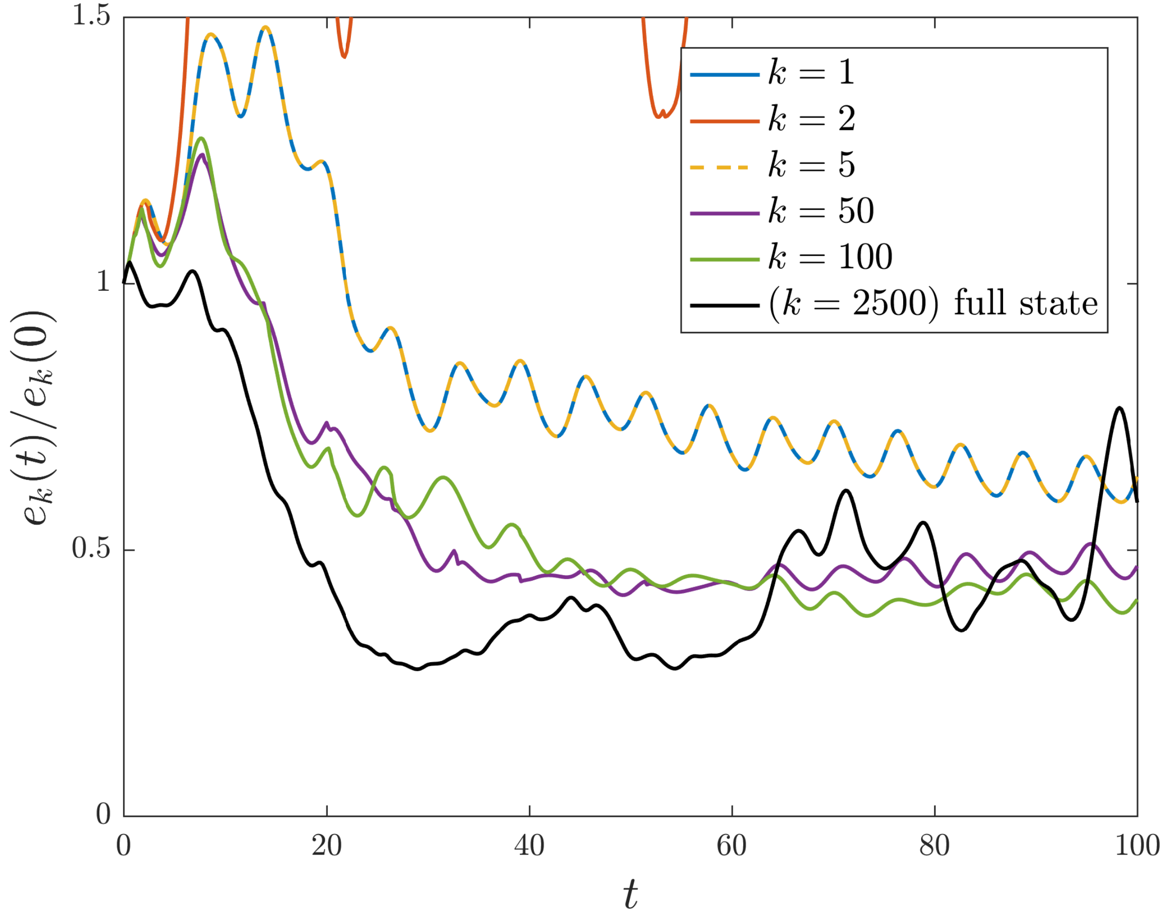
Comparison for different # of measurements

Control task: Re 13k (limit cycle) → Re 10k (locally stable fixed point)



Comparison for different # of measurements

Control task: Re 13k (limit cycle) → Mean flow (unstabilizable fixed point)



Computation time: 10^{-4} second per step

Summary

- Embedding method for analysis & control of nonlinear dynamical systems
- Data-driven
- Fast & scalable (only small convex QPs solved online)

Open problems

- Accuracy of the predictors for finite N – Some answers? *[Kurdila, Bobade, 2018]*
- Choice of the embedding ψ – partly solved *[Korda, Mezić 2018]*
- Guarantees on the controllers (stability, optimality)
- Control for other classes of predictors (bilinear)

References

Koopman MPC — M. Korda, I. Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. Automatica, 2018.

Learning ψ — M. Korda, I. Mezić. Learning Koopman eigenfunctions for prediction and control: the transient case, arXiv, 2018.

Convergence — M. Korda, I. Mezić. On convergence of extended dynamic mode decomposition to the Koopman operator. Journal of Nonlinear Science, 2018.

Rates — A. Kurdila, P. Bobade. Koopman Theory and Linear Approximation Spaces, arxiv 2018.

State estimation — A. Surana, A. Banaszuk. Linear observer synthesis for nonlinear systems using Koopman operator framework. IFAC, 2016.

PDE control — H. Arbabi, M. Korda, I. Mezić. A data-driven Koopman model predictive control framework for nonlinear flows. arXiv, 2018.

Power grid — M. Korda, Y. Susuki, I. Power grid transient stabilization using Koopman predictive control. IFAC CPES & arXiv, 2018.

Numerical examples

Bilinear motor

$$\dot{x}_1 = -(R_a/L_a)x_1 - (k_m/L_a)x_2 u + u_a/L_a$$

$$\dot{x}_2 = -(B/J)x_2 + (k_m/J)x_1 u - \tau_l/J$$

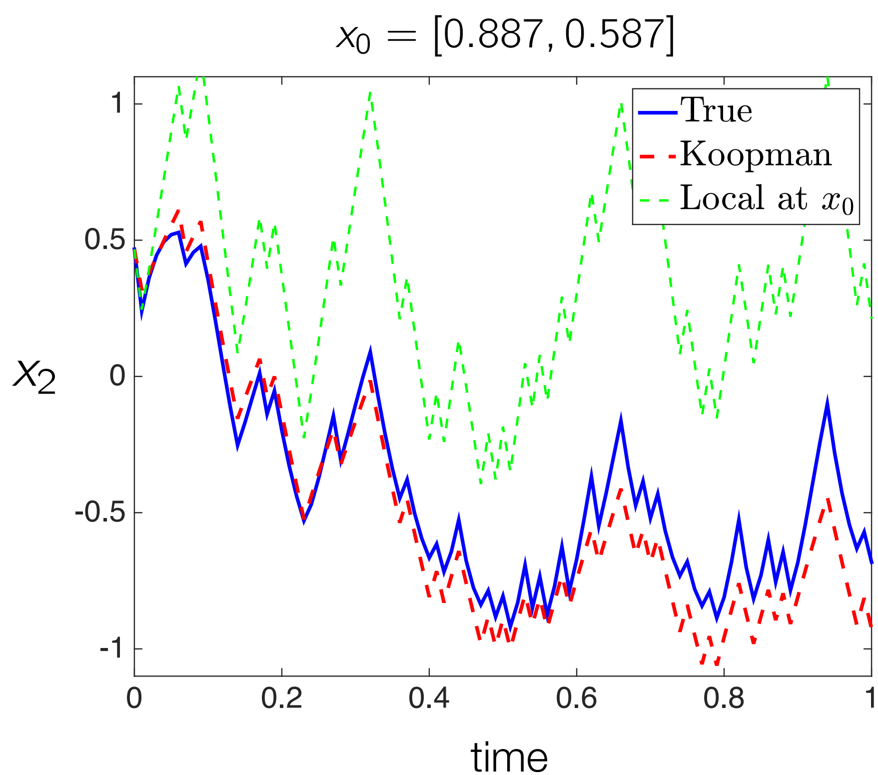
RK-4 discretization with 0.01 s sampling interval

Only x_2 (= angular velocity) measured

Data: 20 trajectories with 1000 samples each

Lifting: state observable + 100 RBFs

Prediction



Numerical examples

Bilinear motor

$$\dot{x}_1 = -(R_a/L_a)x_1 - (k_m/L_a)x_2 u + u_a/L_a$$

$$\dot{x}_2 = -(B/J)x_2 + (k_m/J)x_1 u - \tau_l/J$$

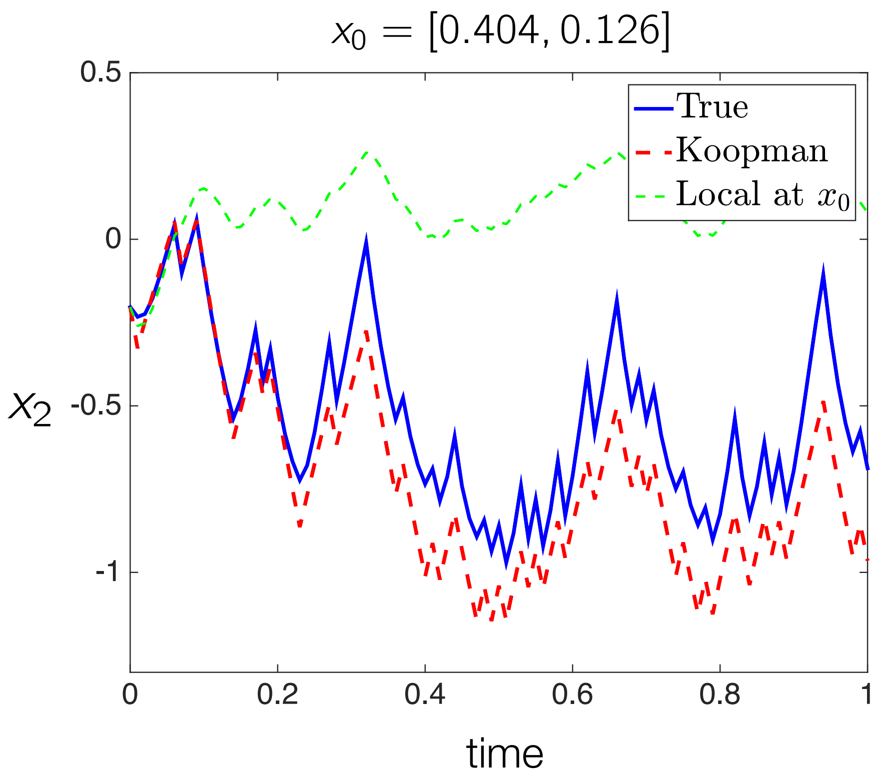
RK-4 discretization with 0.01 s sampling interval

Only x_2 (= angular velocity) measured

Data: 20 trajectories with 1000 samples each

Lifting: state observable + 100 RBFs

Prediction



Numerical examples

Bilinear motor

$$\dot{x}_1 = -(R_a/L_a)x_1 - (k_m/L_a)x_2 u + u_a/L_a$$

$$\dot{x}_2 = -(B/J)x_2 + (k_m/J)x_1 u - \tau_l/J$$

RK-4 discretization with 0.01 s sampling interval

Only x_2 (= angular velocity) measured

Data: 20 trajectories with 1000 samples each

Lifting: state observable + 100 RBFs

Prediction

	Koopman	Local linearization at x_0
Average RMSE	32.3 %	135.5 %

Numerical examples

Bilinear motor

$$\dot{x}_1 = -(R_a/L_a)x_1 - (k_m/L_a)x_2 u + u_a/L_a$$

$$\dot{x}_2 = -(B/J)x_2 + (k_m/J)x_1 u - \tau_l/J$$

RK-4 discretization with 0.01 s sampling interval

Only x_2 (= angular velocity) measured

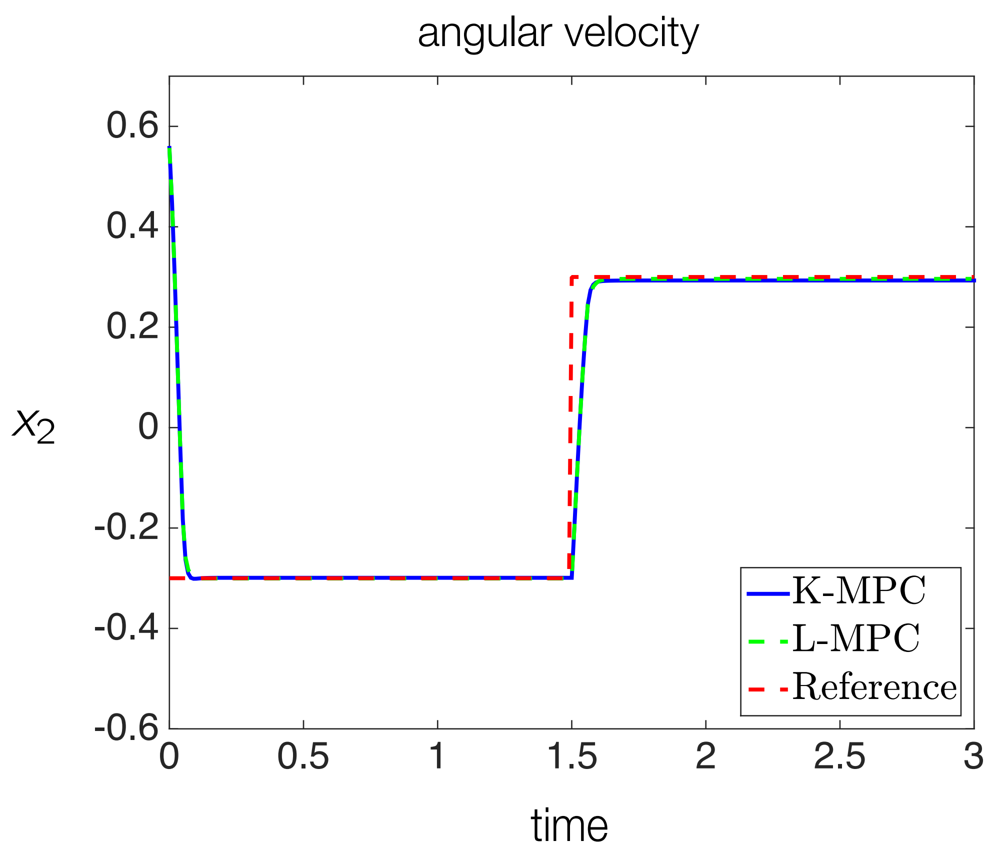
Data: 20 trajectories with 1000 samples each

Lifting: state observable + 100 RBFs

Feedback control

$T_{\text{pred}} = 1\,\text{s}$

$Q = 1,\ R = 0.01$



Numerical examples

Bilinear motor

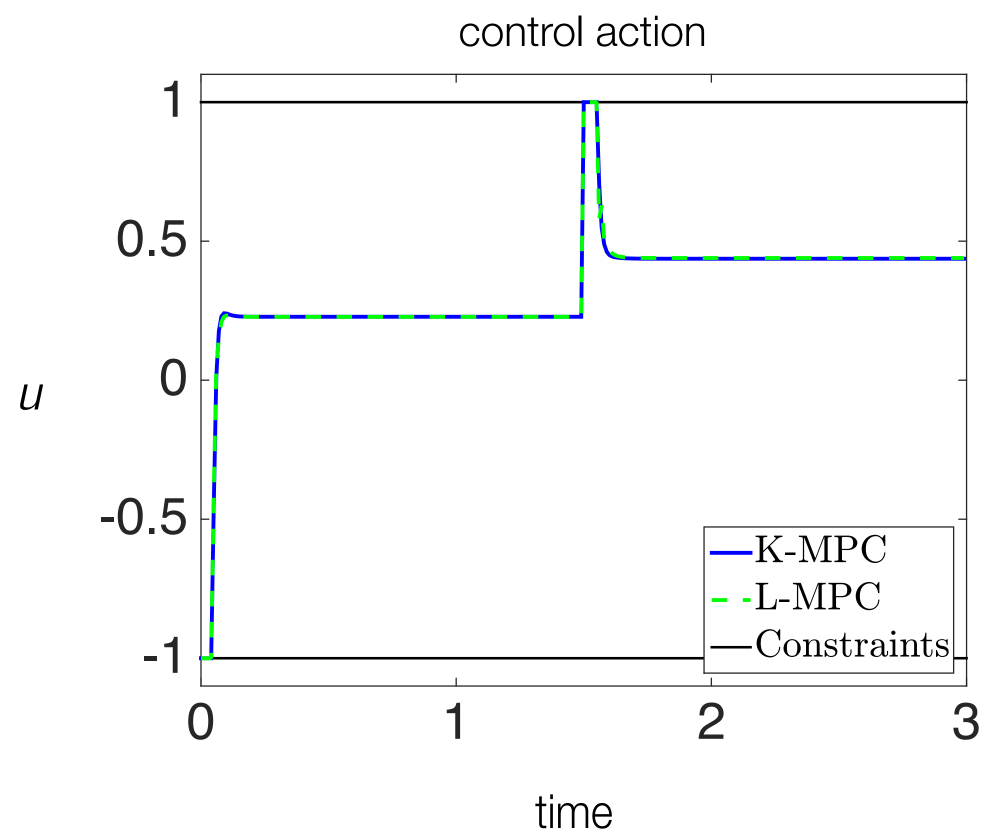
$$\dot{x}_1 = -(R_a/L_a)x_1 - (k_m/L_a)x_2 u + u_a/L_a$$

$$\dot{x}_2 = -(B/J)x_2 + (k_m/J)x_1 u - \tau_l/J$$

RK-4 discretization with 0.01 s sampling interval
Only x_2 (= angular velocity) measured
Data: 20 trajectories with 1000 samples each
Lifting: state observable + 100 RBFs

Feedback control

$T_{\text{pred}} = 1\,\text{s}$
$Q = 1,\ R = 0.01$



Numerical examples

Bilinear motor

$$\dot{x}_1 = -(R_a/L_a)x_1 - (k_m/L_a)x_2 u + u_a/L_a$$

$$\dot{x}_2 = -(B/J)x_2 + (k_m/J)x_1 u - \tau_l/J$$

RK-4 discretization with 0.01 s sampling interval

Only x_2 (= angular velocity) measured

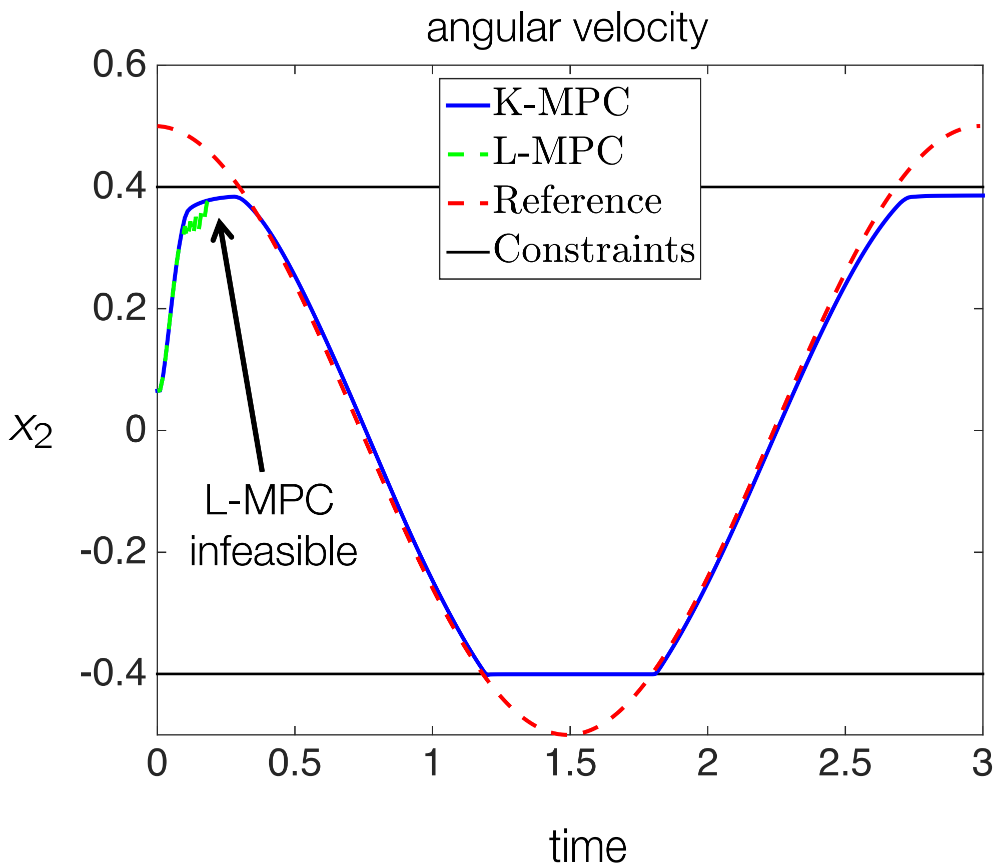
Data: 20 trajectories with 1000 samples each

Lifting: state observable + 100 RBFs

Feedback control

$T_{\text{pred}} = 1\,\text{s}$

$Q = 1,\ R = 0.01$



Numerical examples

Bilinear motor

$$\dot{x}_1 = -(R_a/L_a)x_1 - (k_m/L_a)x_2 u + u_a/L_a$$

$$\dot{x}_2 = -(B/J)x_2 + (k_m/J)x_1 u - \tau_l/J$$

RK-4 discretization with 0.01 s sampling interval

Only x_2 (= angular velocity) measured

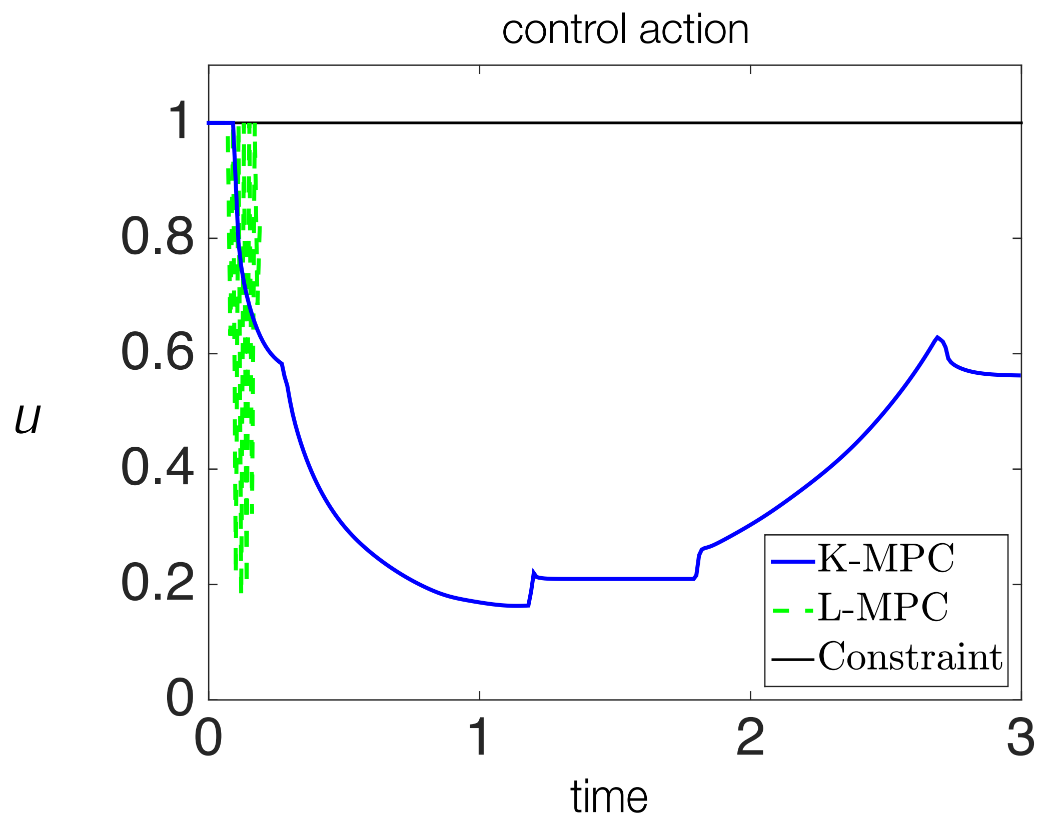
Data: 20 trajectories with 1000 samples each

Lifting: state observable + 100 RBFs

Feedback control

$T_{\text{pred}} = 1\,\text{s}$

$Q = 1,\ R = 0.01$



Numerical examples

Bilinear motor

$$\dot{x}_1 = -(R_a/L_a)x_1 - (k_m/L_a)x_2 u + u_a/L_a$$

$$\dot{x}_2 = -(B/J)x_2 + (k_m/J)x_1 u - \tau_l/J$$

RK-4 discretization with 0.01 s sampling interval

Only x_2 (= angular velocity) measured

Data: 20 trajectories with 1000 samples each

Lifting: state observable + 100 RBFs

Feedback control

$$T_{\text{pred}} = 1\,\text{s}$$

$$Q = 1,\ R = 0.01$$

Average computation time = 6.83 ms

(Matlab + qpOASES, 2GHz i7)