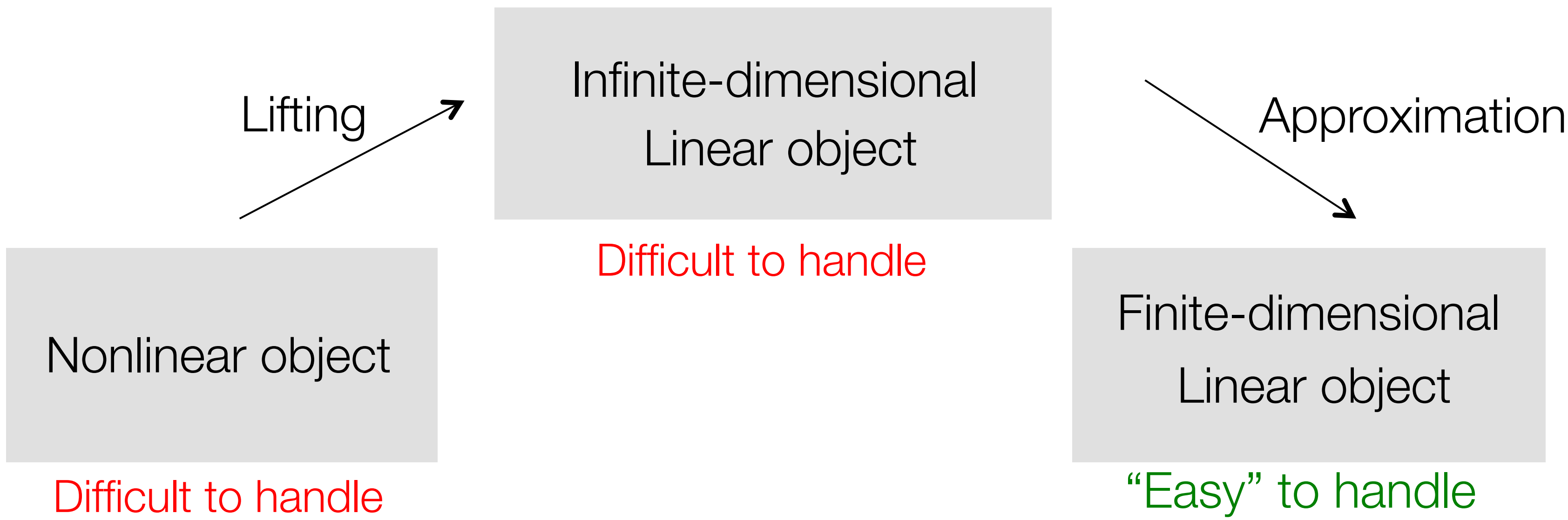
Operator-theoretic methods for prediction and control of nonlinear dynamical systems

Milan Korda

UNIVERSITY OF CALIFORNIA
SANTA BARBARA

Big picture


Infinite-dimensional
Linear object
Lifting
Approximation
Difficult to handle
Nonlinear object
Finite-dimensional
Linear object
Difficult to handle
“Easy” to handle

Big picture

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

Nonlinear dynamics

Lifting

Linear operator

Infinite-dimensional operator

Approximation

Approximation of linear operator

Matrix

Koopman operator

Koopman operator

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

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

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

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

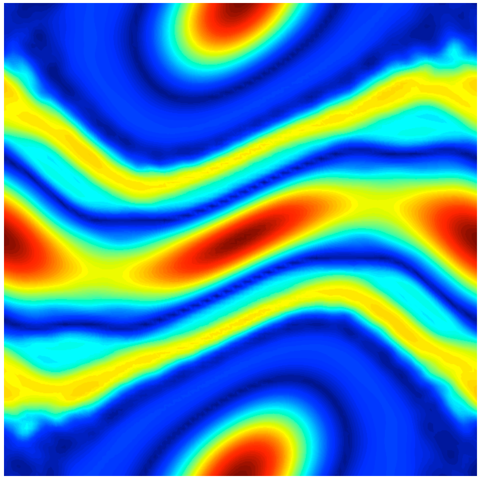
$$\phi \circ f^k = \lambda^k \phi$$ Linear coordinate

Koopman operator

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

$$\phi \circ f^k = \lambda^k \phi$$

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

Chirikov standard map

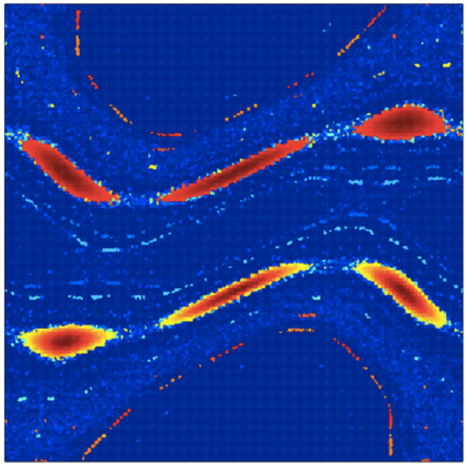
Koopman operator

Eigenfunctions

$$\phi \circ f^k = \lambda^k \phi$$

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

(ω rational)



Chirikov standard map
[Budisic et al. 2012]

Koopman operator

Eigenfunctions

$$\phi \circ f^k = \lambda^k \phi$$

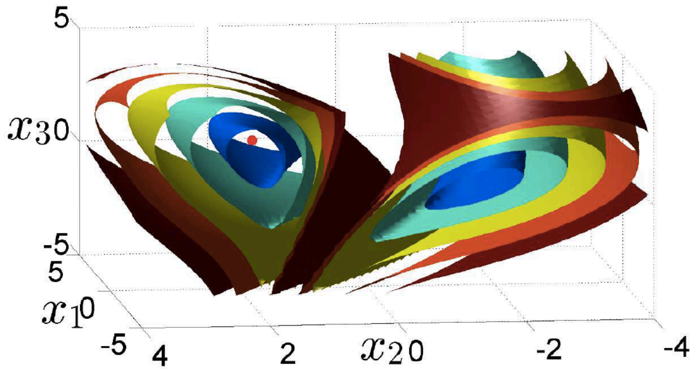
Isostables

Isochrons

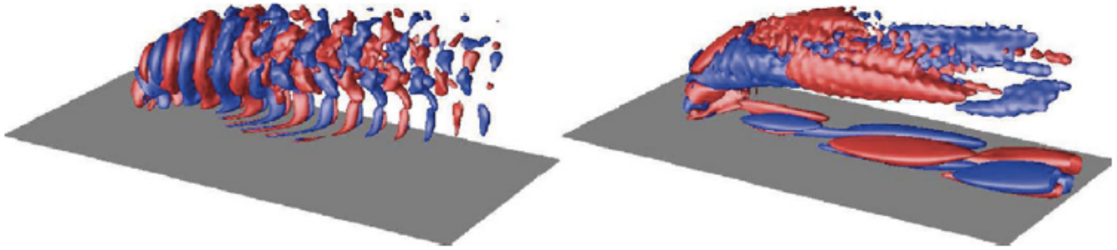
Ergodic partition

Model reduction

⋮



[Mauroy et al. 2013]



[Rowley et al. 2009]

Prediction

Prediction

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

$$g = \sum_{i=1}^{N} c_i \psi_i \in \mathcal{H}_N$$

Prediction

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

$$g = \sum_{i=1}^{N} c_i \psi_i \in \mathcal{H}_N$$

Linear predictor

$$g(x_k) = C A^k z_0$$

$$C = [c_1, \ldots, c_N] \qquad z_0 = \boldsymbol{\psi}(x_0) = \begin{bmatrix} \psi_1(x_0) \\ \vdots \\ \psi_N(x_0) \end{bmatrix}$$

Prediction

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

$$g = \sum_{i=1}^{N} c_i \psi_i \in \mathcal{H}_N$$

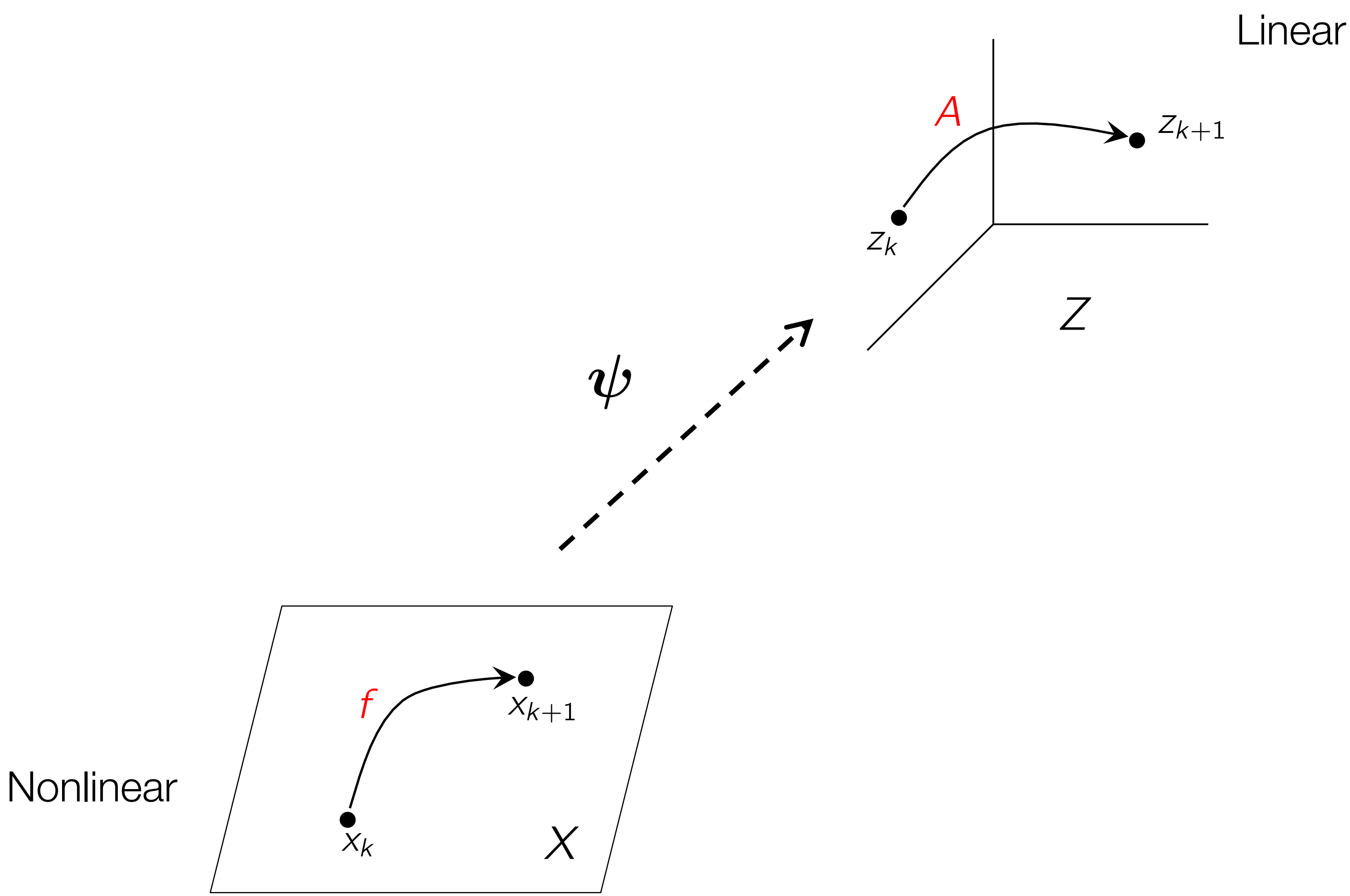
Linear predictor

$$g(x_k) = C A^k z_0$$

$$C = [c_1, \ldots, c_N] \qquad z_0 = \psi(x_0) = \begin{bmatrix} \psi_1(x_0) \\ \vdots \\ \psi_N(x_0) \end{bmatrix}$$

ψ_i's eigenfunctions $\Rightarrow$ $A = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$

Linear prediction



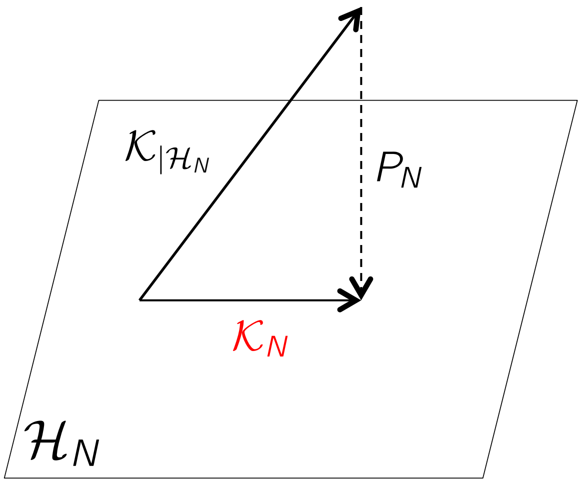
Approximation of the Koopman operator

Approximation

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

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

- Construct
- Analyze



Extended dynamic mode decomposition

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

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

LS problem

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

Koopman approximaton

$$\mathcal{K}_{N,K} g := c^\top A_{N,K} \boldsymbol{\psi}$$

$$g = c^\top \psi$$

[Williams et al., 2015]

Convergence of EDMD

$$\mathcal{K}_{N,K} g := c^\top A_{N,K} \boldsymbol{\psi}$$

$$g = c^\top \boldsymbol{\psi}$$

$\hat{\mu}_K$ = empirical measure on $x_1, \ldots, x_K$

Fact: $\mathcal{K}_{N,K} = P_N^{\hat{\mu}_K} \mathcal{K}_{|\mathcal{H}_N}$

Convergence of EDMD

$$\mathcal{K}_{N,K}\, g := c^\top A_{N,K} \boldsymbol{\psi} \qquad g = c^\top \boldsymbol{\psi}$$

$\hat{\mu}_K =$ empirical measure on $x_1, \ldots, x_K$

Fact: $\mathcal{K}_{N,K} = P_N^{\hat{\mu}_K} \mathcal{K}_{|\mathcal{H}_N}$

Fact: $\lim_{K\to\infty} \mathcal{K}_{N,K} \to \mathcal{K}_N = P_N^{\mu} \mathcal{K}_{|\mathcal{H}_N}$

(iid or ergodic sampling from μ)

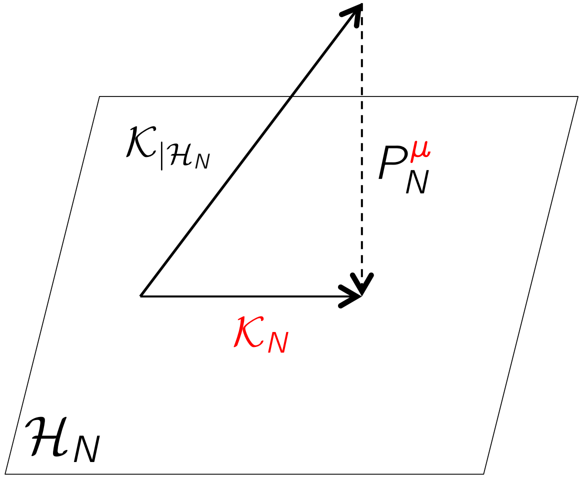
Convergence of EDMD

$$\mathcal{K}_{N,K}\, g := c^\top A_{N,K} \boldsymbol{\psi}$$

$$g = c^\top \boldsymbol{\psi}$$

Fact: $\lim_{K\to\infty} \mathcal{K}_{N,K} \to \mathcal{K}_N = P_N^\mu \mathcal{K}_{|\mathcal{H}_N}$

(iid or ergodic sampling from μ)



Convergence of EDMD

$\mathcal{H} = L_2(\mu)$

Theorem

- $\overline{\mathrm{span}\{\psi_i\}_{i=1}^{\infty}} = \mathcal{H}$
- $\mathcal{K} : \mathcal{H} \to \mathcal{H}$ bounded

$$\lim_{N\to\infty} \|\mathcal{K}_N g - \mathcal{K} g\| = 0 \ \text{ for all } \ g \in \mathcal{H}$$

(Converge in strong operator topology)

Convergence of EDMD

$\mathcal{H} = L_2(\mu)$

Theorem

- $\overline{\text{span}\{\psi_i\}_{i=1}^{\infty}} = \mathcal{H}$
- $\mathcal{K} : \mathcal{H} \to \mathcal{H}$ bounded

$$\lim_{N\to\infty} \|\mathcal{K}_N g - \mathcal{K} g\| = 0 \quad \text{for all} \quad g \in \mathcal{H}$$

(Converge in strong operator topology)

As a result, any finite-horizon predictions converge!

Corollary

Under the same assumptions

For any $N_p \in \mathbb{N}$: $\lim_{N\to\infty} \sup_{i\in\{1,\ldots,N_p\}} \|\mathcal{K}_N^i g - \mathcal{K}^i g\| = 0$

Convergence of EDMD

$\mathcal{H} = L_2(\mu)$

Theorem

- $\overline{\mathrm{span}\{\psi_i\}_{i=1}^{\infty}} = \mathcal{H}$
- $\mathcal{K} : \mathcal{H} \to \mathcal{H}$ bounded

$$\lim_{N\to\infty} \|\mathcal{K}_N g - \mathcal{K} g\| = 0 \ \text{ for all } \ g \in \mathcal{H}$$

(Converge in strong operator topology)

As a result, any finite-horizon predictions converge!

Corollary

Under the same assumptions

For any $N_p \in \mathbb{N}$: $\lim_{N\to\infty} \sup_{i\in\{1,\ldots,N_p\}} \|\mathcal{K}_N^i g - \mathcal{K}^i g\| = 0$

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

Convergence of EDMD

$\mathcal{H} = L_2(\mu)$

Theorem

- $\overline{\mathrm{span}\{\psi_i\}_{i=1}^{\infty}} = \mathcal{H}$
- $\mathcal{K} : \mathcal{H} \to \mathcal{H}$ bounded

$$\lim_{N\to\infty} \|\mathcal{K}_N g - \mathcal{K} g\| = 0 \quad \text{for all} \quad g \in \mathcal{H}$$

(Converge in strong operator topology)

As a result, any finite-horizon predictions converge!

Corollary

Under the same assumptions

For any $N_p \in \mathbb{N}$: $\lim_{N\to\infty} \sup_{i\in\{1,\ldots,N_p\}} \|\mathcal{K}_N^i g - \mathcal{K}^i g\| = 0$

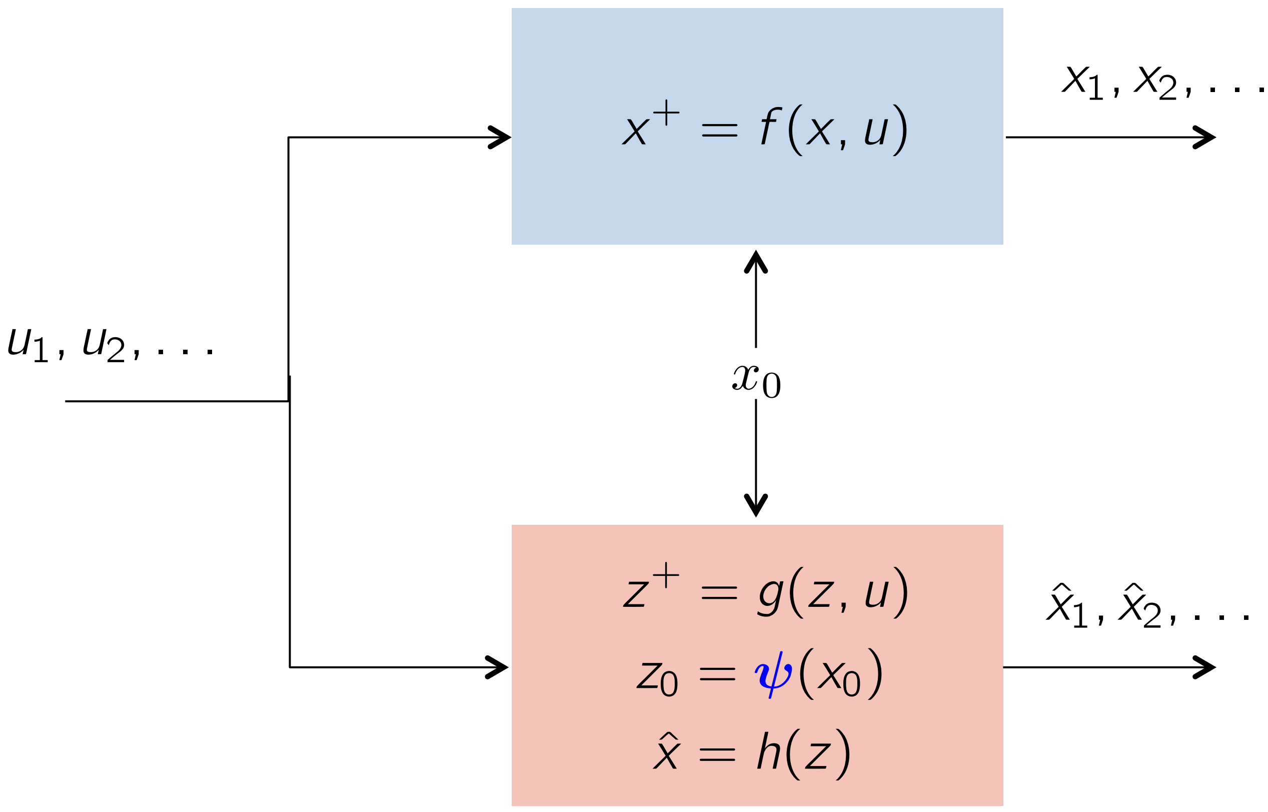
Spectral convergence more delicate. See

Korda M. and Mezić I. *On Convergence of Extended Dynamic Mode Decomposition to the Koopman Operator*, Journal of Nonlinear Science, 2017

Control

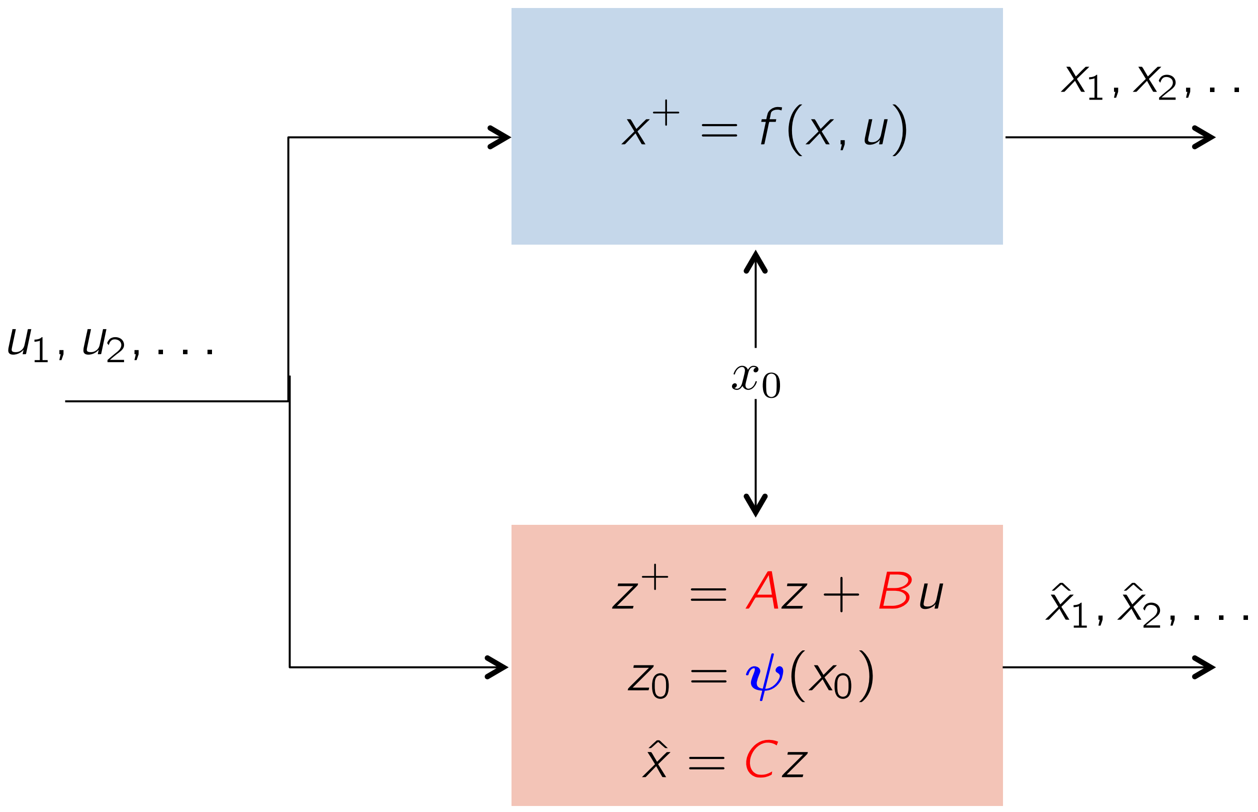
(Joint work with Igor Mezić)

Predictor



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

Predictor



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

Why linear predictors?

Can design controllers using **linear** methods

$$u = \kappa_{\text{lift}}(z)$$

$$\Downarrow$$

$$u = \kappa(x) := \kappa_{\text{lift}}(\boldsymbol{\psi}(x))$$

Why linear predictors?

Can design controllers using **linear** methods

$$u = \kappa_{\text{lift}}(z)$$

$$\Downarrow$$

$$u = \kappa(x) := \kappa_{\text{lift}}(\psi(x))$$

Especially suited for Model predictive control (MPC)

- Optimization-based controller
- **Fast** and **effective** for constrained linear systems
- Computation speed **independent** of the size of the lift N

Designing the predictors

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 $\chi := (x, \boldsymbol{u}) \in \mathcal{X} := \mathbb{R}^n \times \underbrace{\ell(\mathbb{R}^m)}$
- 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 $\chi := (x, \boldsymbol{u}) \in \mathcal{X} := \mathbb{R}^n \times \underbrace{\ell(\mathbb{R}^m)}$
- 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 $\mathcal{K}\phi = \phi \circ F$ $\phi : \mathcal{X} \to \mathbb{R}$

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

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

Other work

[Proctor et al, 2016] [Proctor et al, 2016]

[Williams et al, 2016] [Brunton et al, 2016]

Linear predictors from Koopman - EDMD

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

LS problem

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

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

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

$$\phi(x, \boldsymbol{u}) = [\psi_1(x), \ldots, \psi_N(x), \boldsymbol{u}(0)_1, \ldots, \boldsymbol{u}(0)_m]^\top$$

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

Without loss of generality

$$\phi(x,\boldsymbol{u}) = [\psi_1(x),\ldots,\psi_N(x),\boldsymbol{u}(0)_1,\ldots,\boldsymbol{u}(0)_m]^\top$$

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

Algorithm summary

Data $\mathbf{X} = \left[x_1, \ \ldots, x_K\right], \ \mathbf{Y} = \left[x_1^+, \ldots, x_K^+\right], \mathbf{U} = \left[u_1, \ldots, u_K\right]$

Lifting $\mathbf{X}_{\text{lift}} = \left[\boldsymbol{\psi}(x_1), \ \ldots, \boldsymbol{\psi}(x_K)\right], \ \mathbf{Y}_{\text{lift}} = \left[\boldsymbol{\psi}(x_1^+), \ldots, \boldsymbol{\psi}(x_K^+)\right]$

LS problem $\min_{A,B} \|\mathbf{Y}_{\text{lift}} - A\mathbf{X}_{\text{lift}} - B\mathbf{U}\|_F \ , \quad \min_{C} \|\mathbf{X} - C\mathbf{X}_{\text{lift}}\|_F$

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

$$z^+ = Az + Bu$$
$$\hat{x} = Cz$$

$$z_0 = \boldsymbol{\psi}(x_0) = \begin{bmatrix} \psi_1(x_0) \\ \vdots \\ \psi_N(x_0) \end{bmatrix}$$

MPC design

Koopman MPC

Nonlinear MPC

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

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

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

x

Koopman MPC

Koopman MPC

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

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

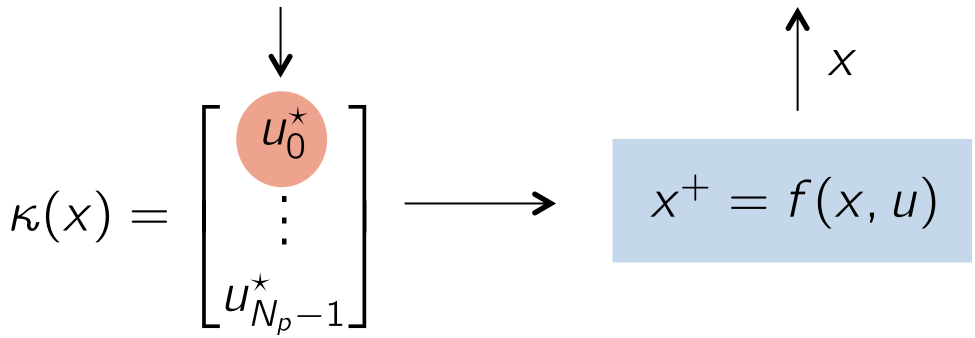
x

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

Koopman MPC

Dense-form Koopman MPC

$$\begin{array}{ll} \underset{U \in \mathbb{R}^{mN_p}}{\text{minimize}} & U^\top H U^\top + h^\top U + z_0^\top G U \\ \text{subject to} & LU + Mz_0 \leq c \\ & \\ \text{parameter} & z_0 = \psi(x) \end{array}$$



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

Koopman MPC summary

At each step k of closed-loop operation

- Set $z_0 = \psi(x_k)$

- Solve

$$\begin{array}{ll} \underset{U \in \mathbb{R}^{mN_p}}{\text{minimize}} & U^\top H U^\top + h^\top U + z_0^\top G U \\ \text{subject to} & LU + Mz_0 \leq c \end{array}$$

- Apply $U^\star_{1:m}$ to the system

Koopman MPC summary

At each step k of closed-loop operation

- Set $z_0 = \psi(x_k)$

- Solve

$$\begin{array}{ll} \underset{U \in \mathbb{R}^{mN_p}}{\text{minimize}} & U^\top H U^\top + h^\top U + z_0^\top G U \\ \text{subject to} & LU + Mz_0 \leq c \end{array}$$

- Apply $U^\star_{1:m}$ to the system

Main benefits

Computation cost **independent** of the size of the lift

Can handle **nonlinear constraints** and **costs** 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

- Systems with disturbances

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

Solution: Treat w as an additional input

Numerical examples

Numerical examples

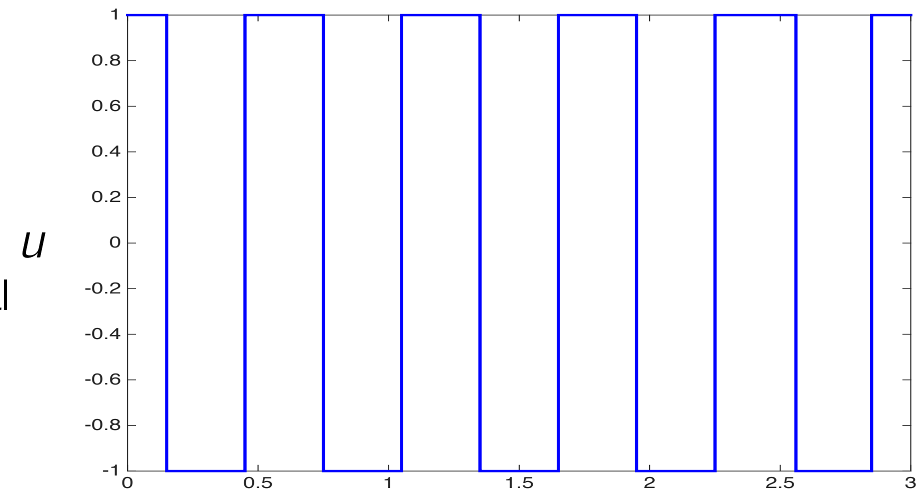
Van der Pol oscillator

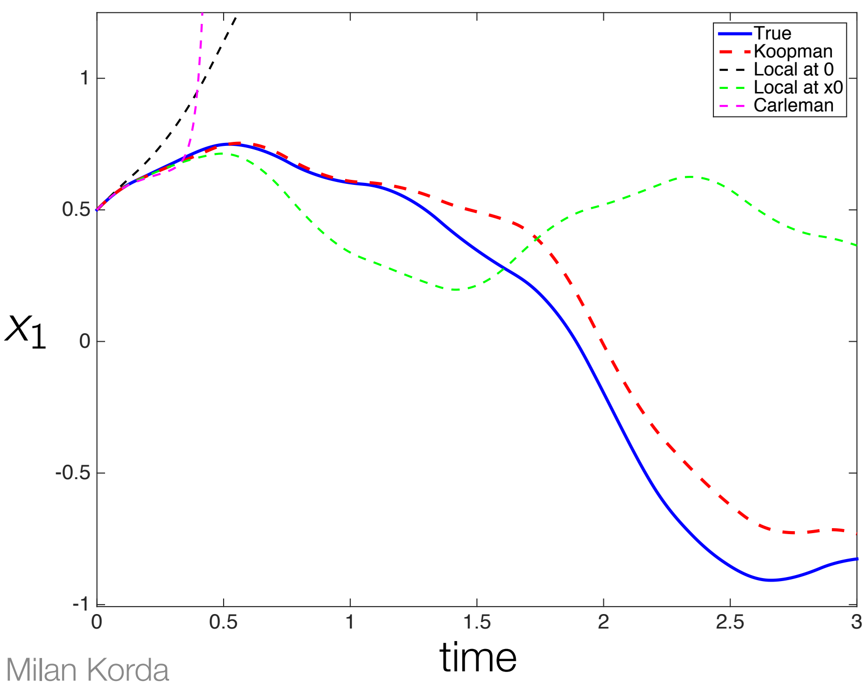
$$\dot{x}_1 = 2x_2$$
$$\dot{x}_2 = -0.8x_1 + 2x_2 - 10x_1^2 x_2 + u$$

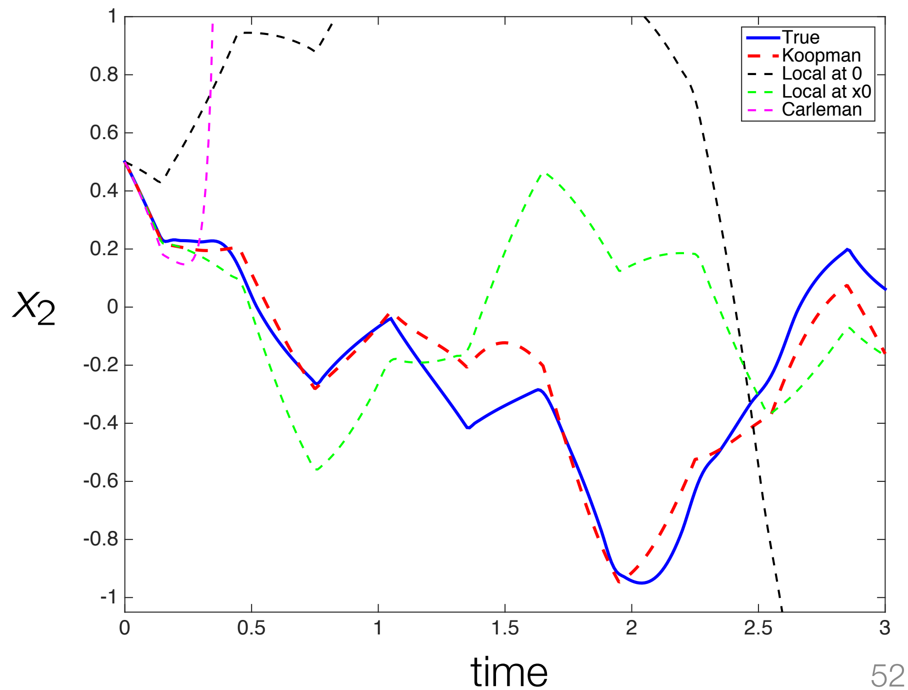
RK-4 discretization with 0.01 s sampling interval

Data: 20 trajectories with 1000 samples each

Lifting: state observable + 100 RBFs







Numerical examples

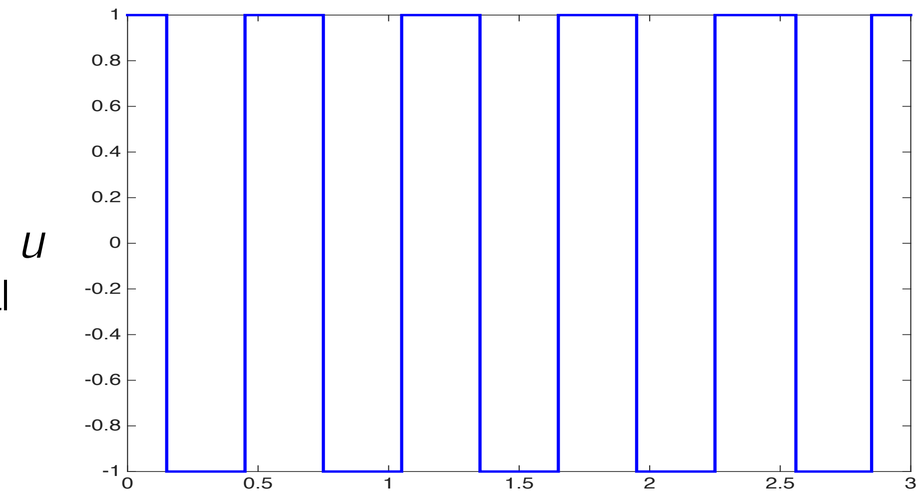
Van der Pol oscillator

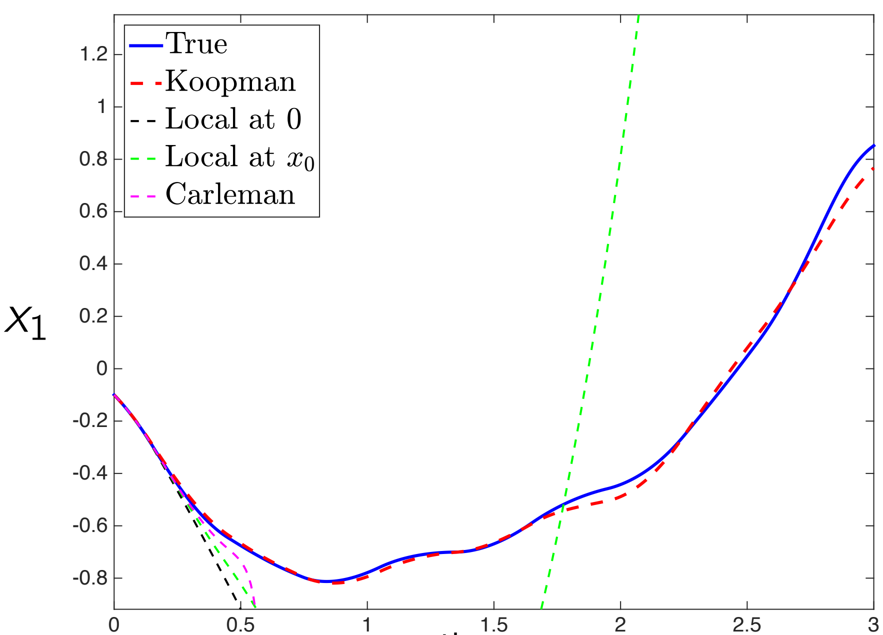
$$\dot{x}_1 = 2x_2$$
$$\dot{x}_2 = -0.8x_1 + 2x_2 - 10x_1^2x_2 + u$$

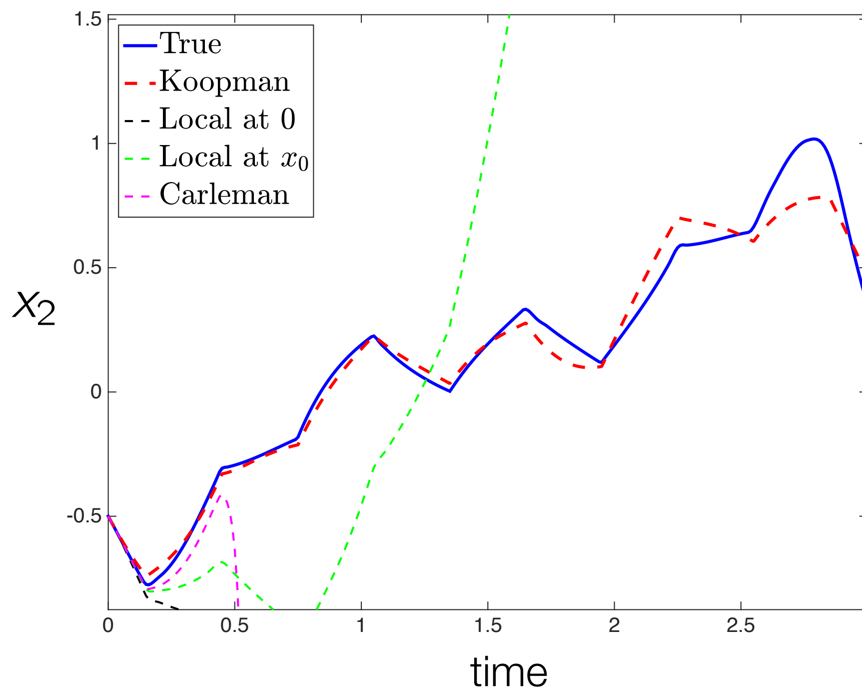
RK-4 discretization with 0.01 s sampling interval

Data: 20 trajectories with 1000 samples each

Lifting: state observable + 100 RBFs







Numerical examples

Van der Pol oscillator

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

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

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 %

Numerical examples

Van der Pol oscillator

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

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

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	$5.08 \cdot 10^{22}$ %

Numerical examples

Bilinear motor

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

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

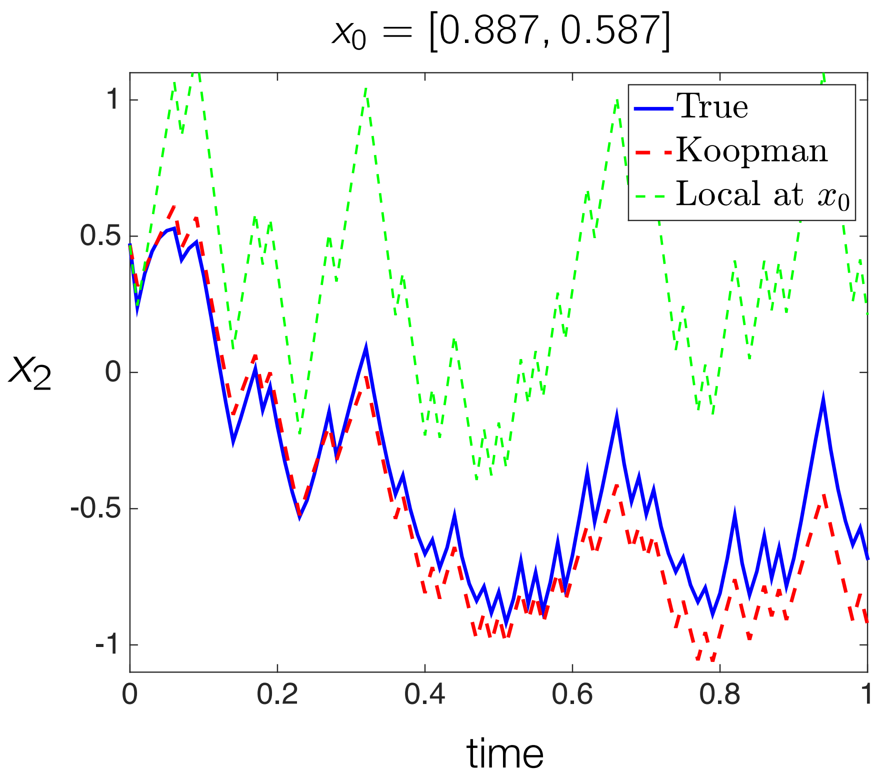
RK-4 discretization with 0.01 s sampling interval

Only x_2 (= angular velocity) measured

Data: 20 trajectories with 1000 samples each

Lifting: state observable + 100 RBFs

Prediction



Numerical examples

Bilinear motor

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

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

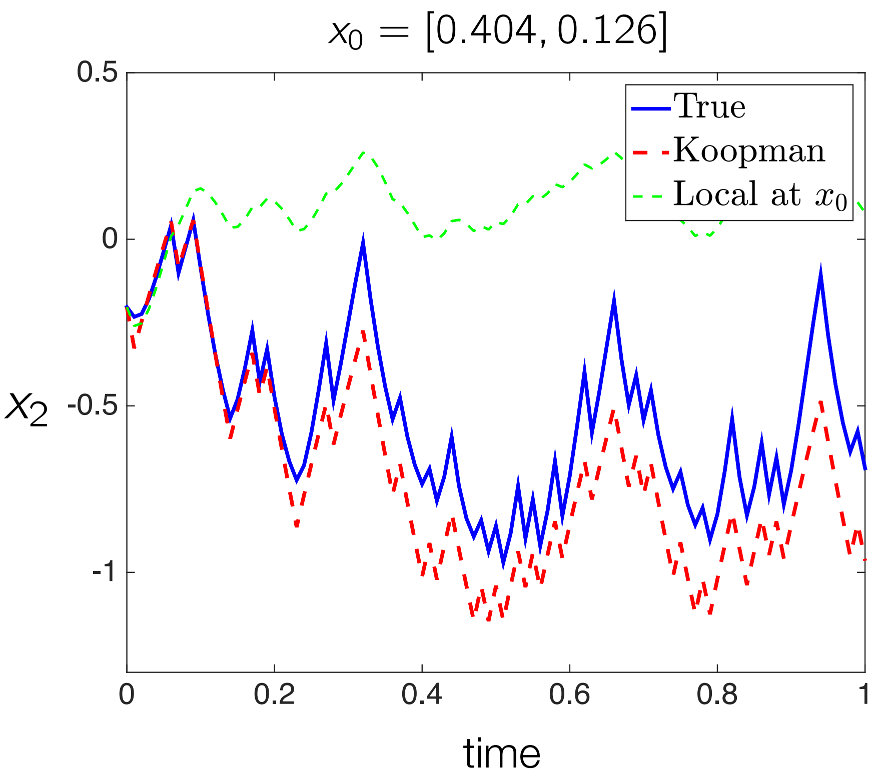
RK-4 discretization with 0.01 s sampling interval

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

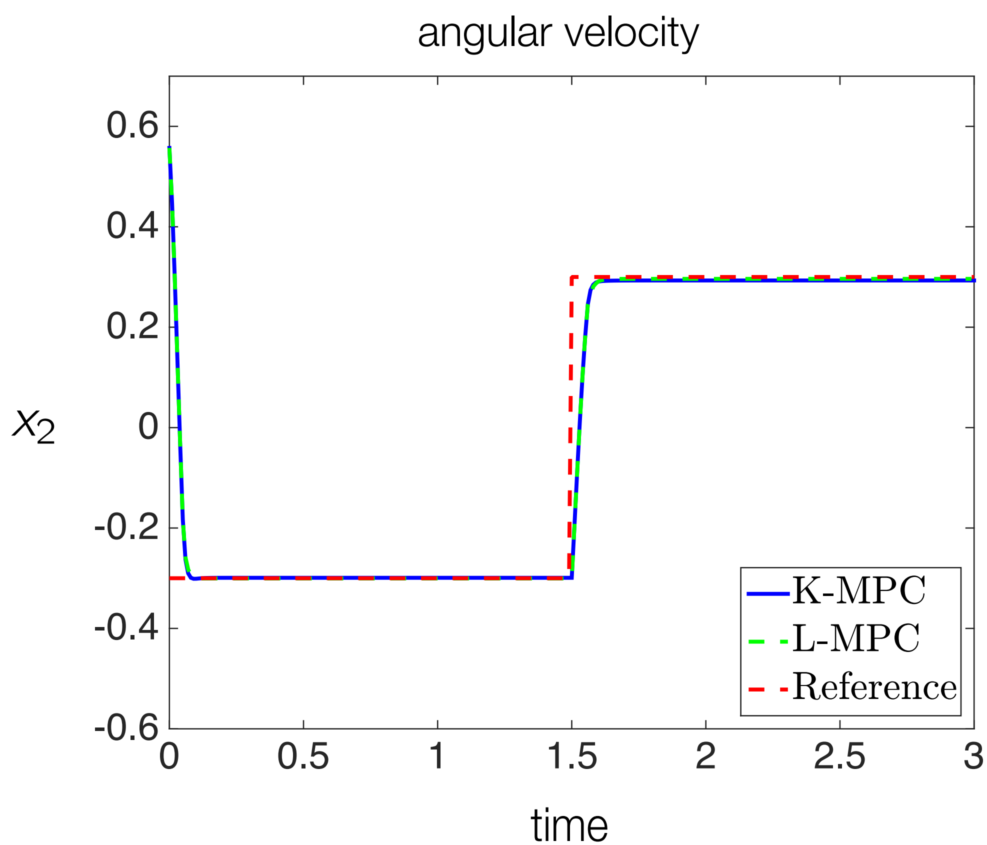
Data: 20 trajectories with 1000 samples each

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

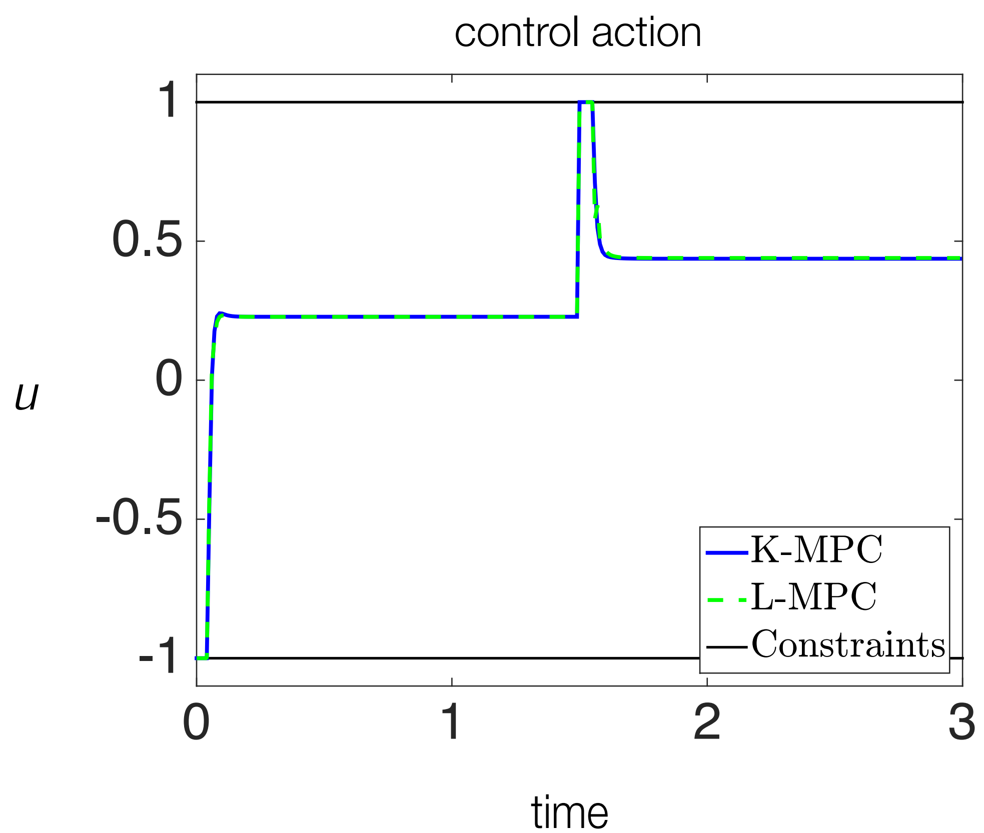
Data: 20 trajectories with 1000 samples each

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

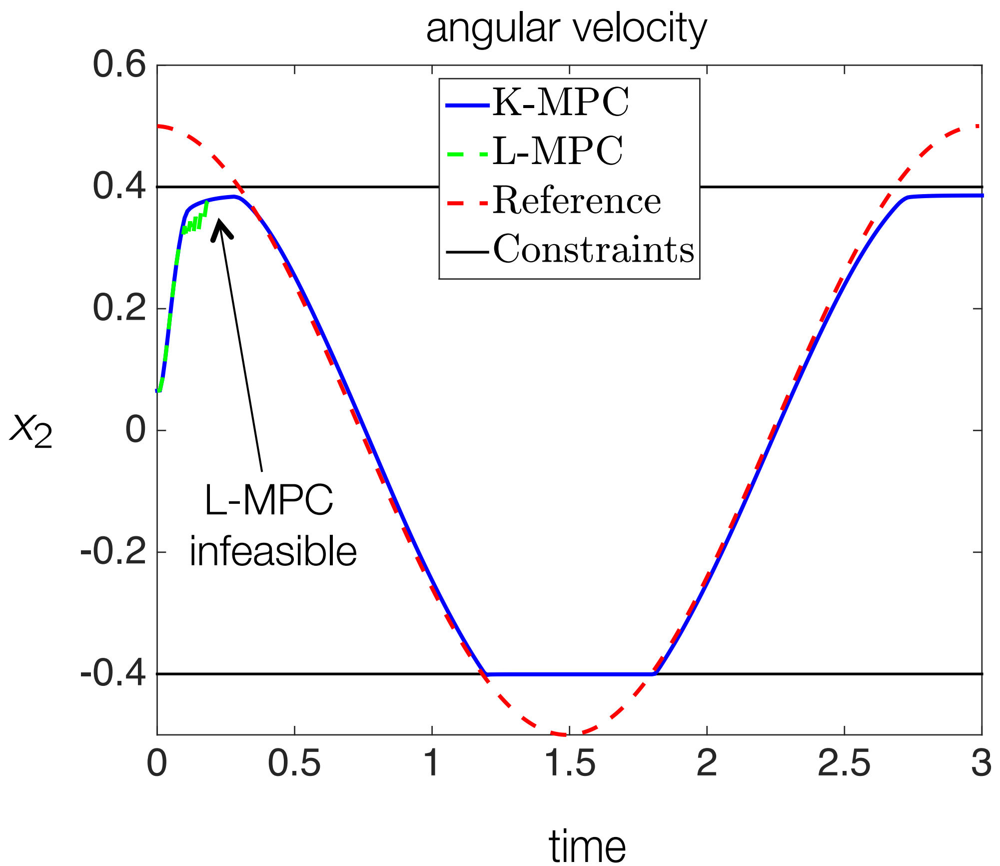
Data: 20 trajectories with 1000 samples each

Lifting: state observable + 100 RBFs

Feedback control

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

$Q = 1,\ R = 0.01$



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

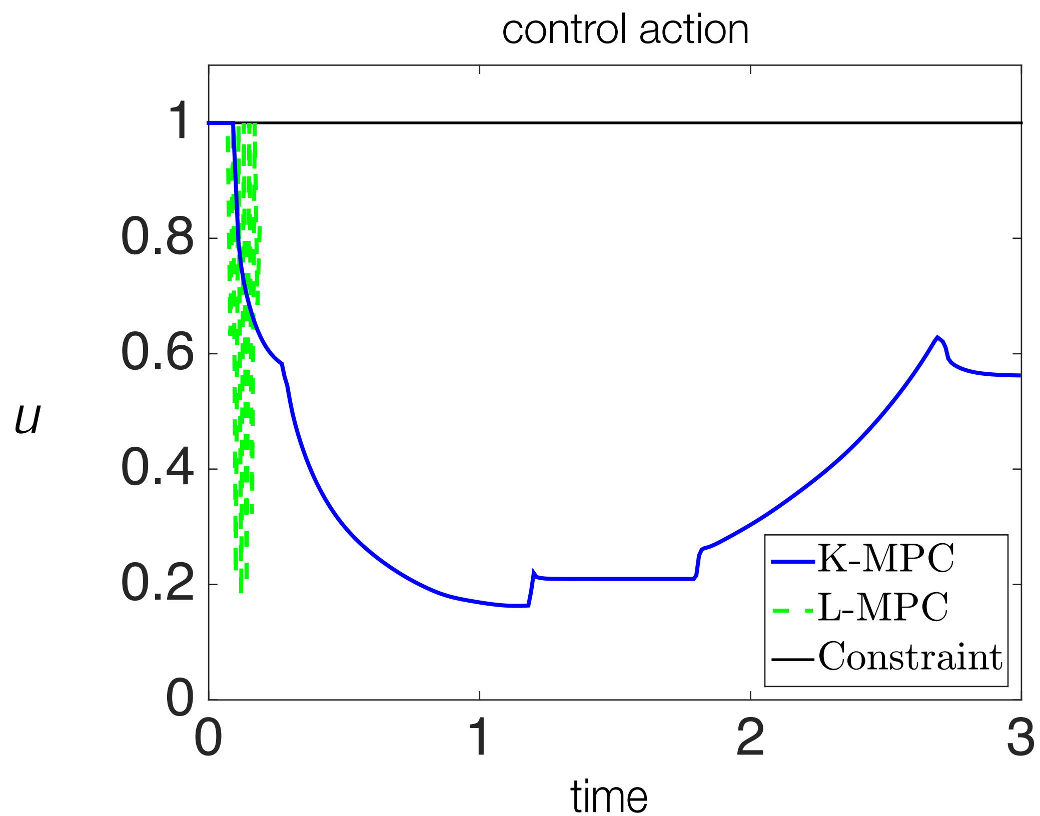
Data: 20 trajectories with 1000 samples each

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)

Powergrid control

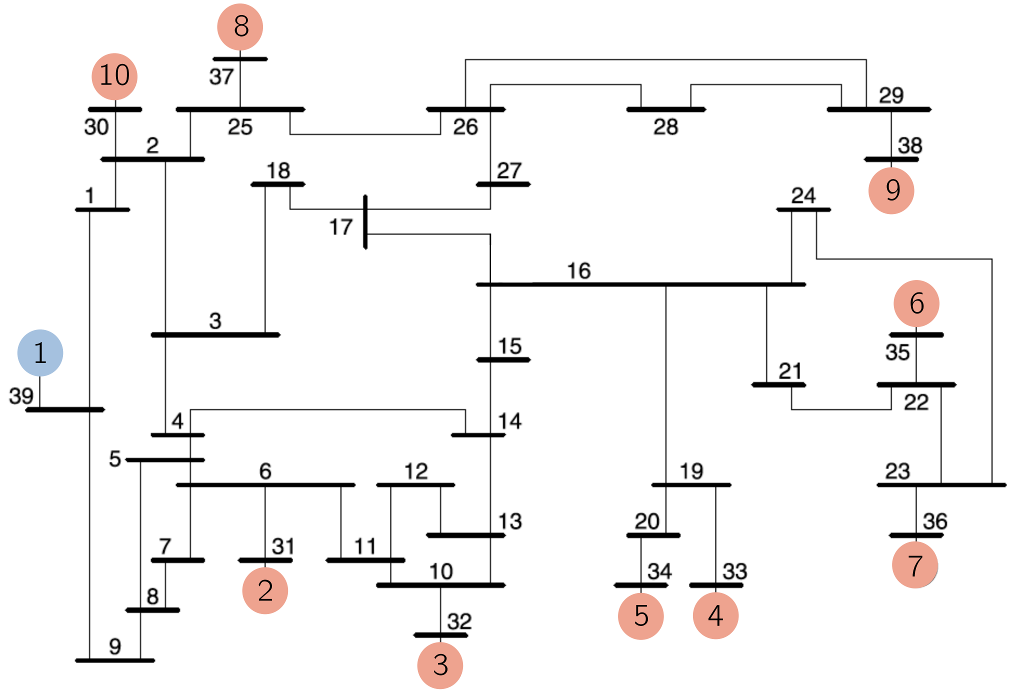
(Joint work with Yoshihiko Susuki)

Numerical examples

New England power grid

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

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



Numerical examples

New England power grid

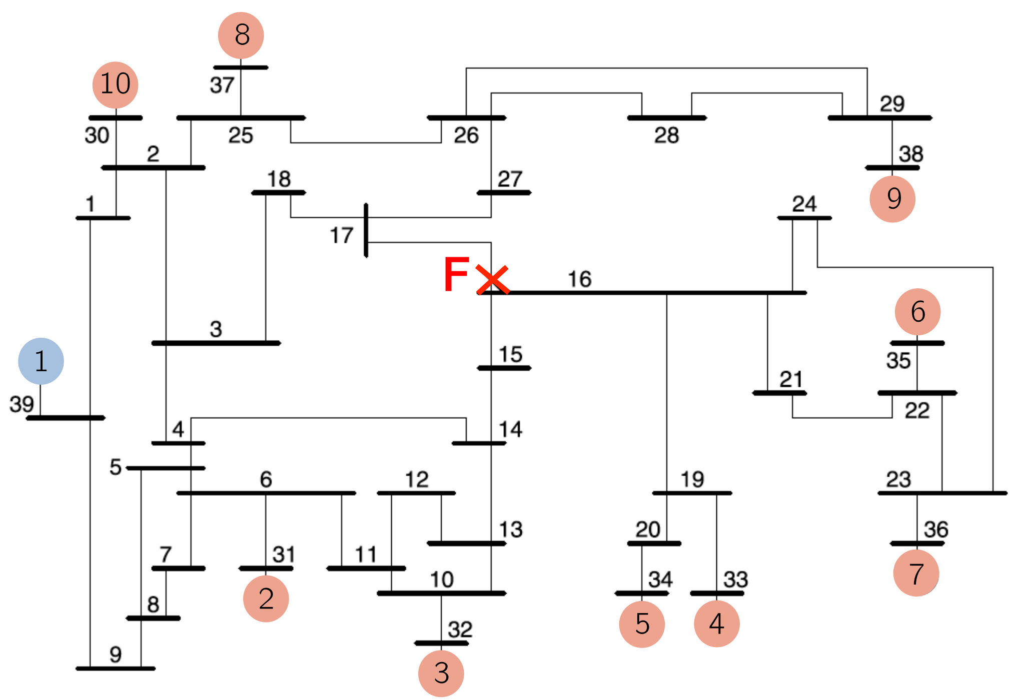
$$\dot{\delta}_i = \omega_i$$

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

Setup from *[Susuki et al, 2011]*

$t = 0.67\,\mathrm{s}$ – fault occurs

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



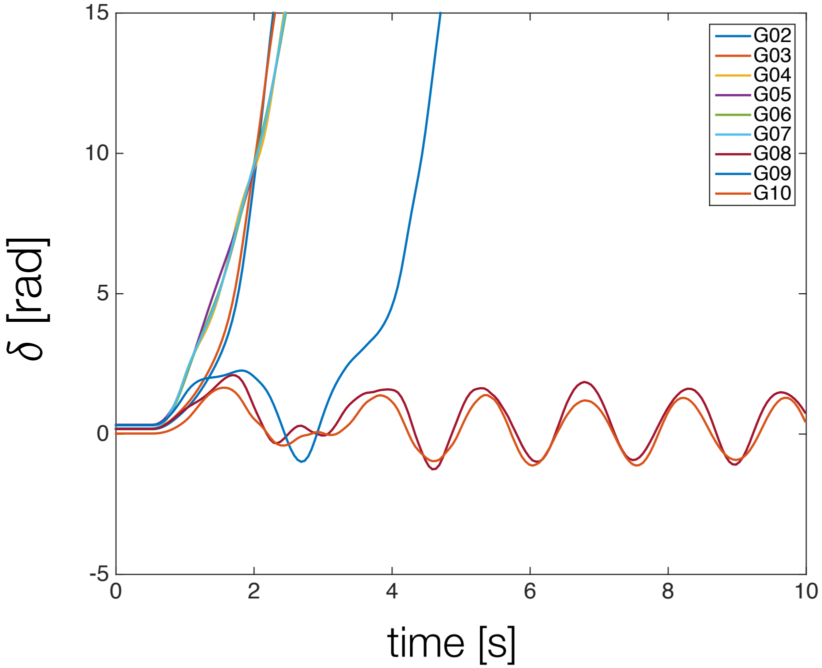
Numerical examples

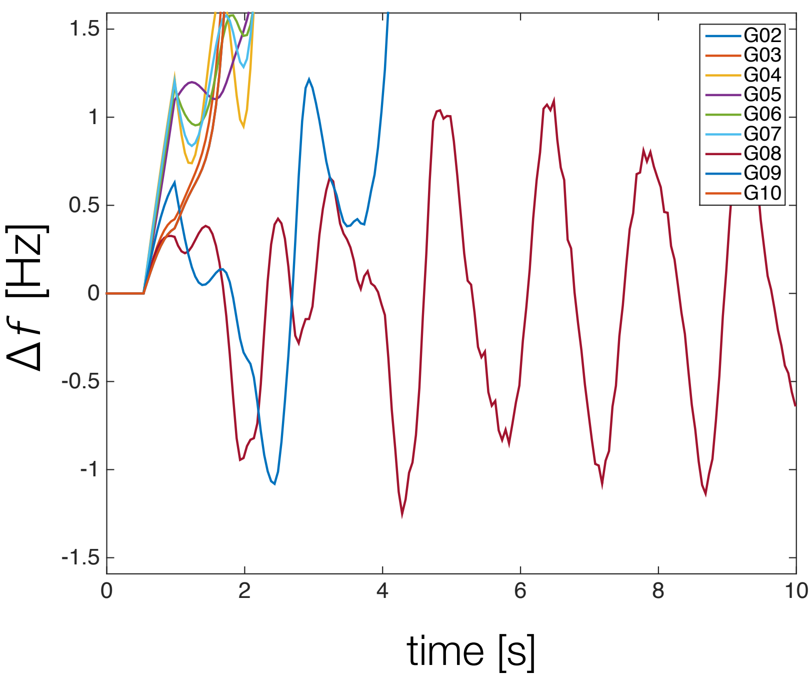
New England power grid

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

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

No control





Numerical examples

New England power grid

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

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

Control

Actuation: P_{m_i} or V_i - mechanical power or generator voltage

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

Pred. horizon: 1 second

Sampling: 50 ms

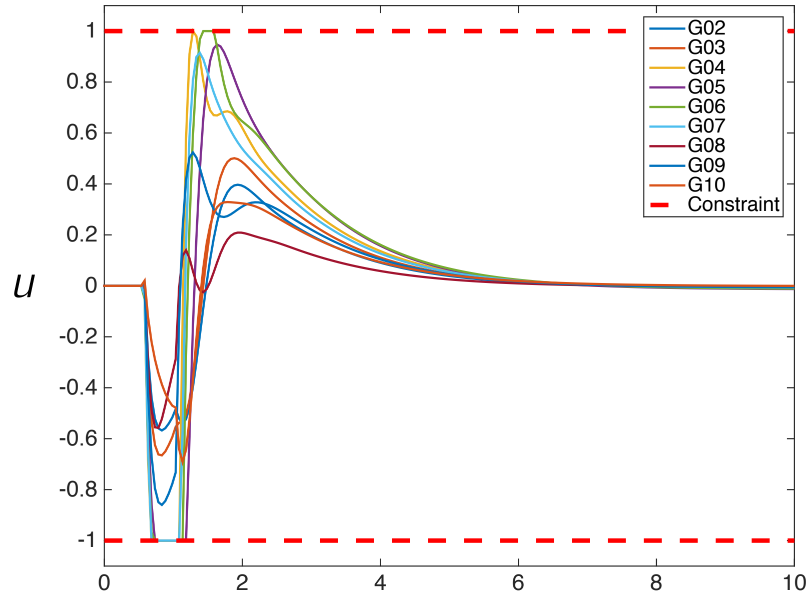
Numerical examples

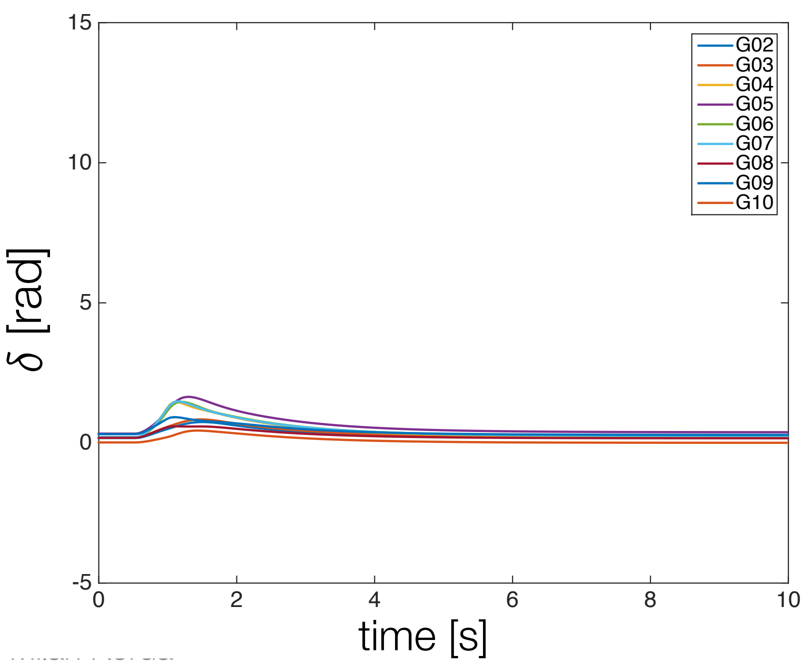
New England power grid

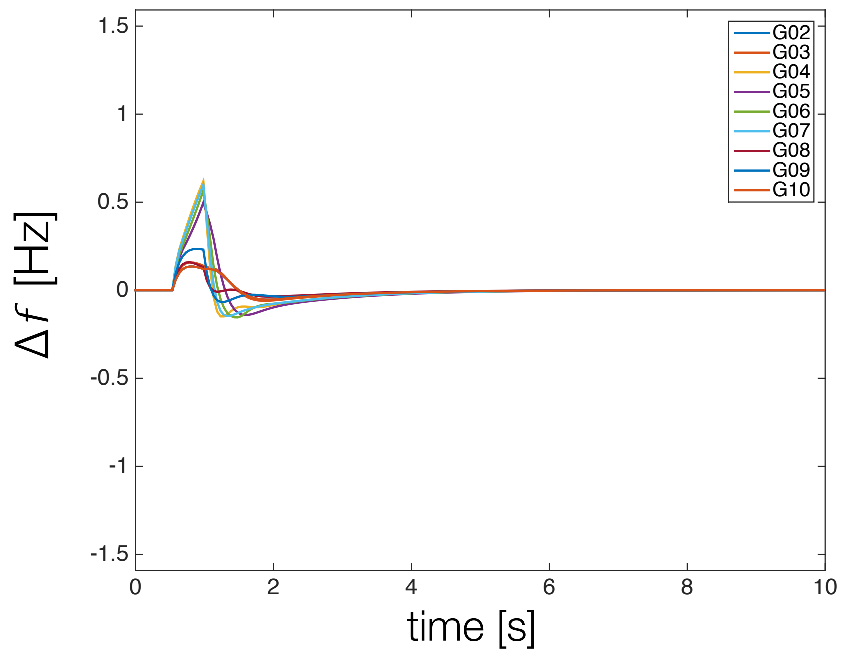
$$\dot{\delta}_i = \omega_i$$

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

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







Numerical examples – NE cascade

7 NE grid cascade

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

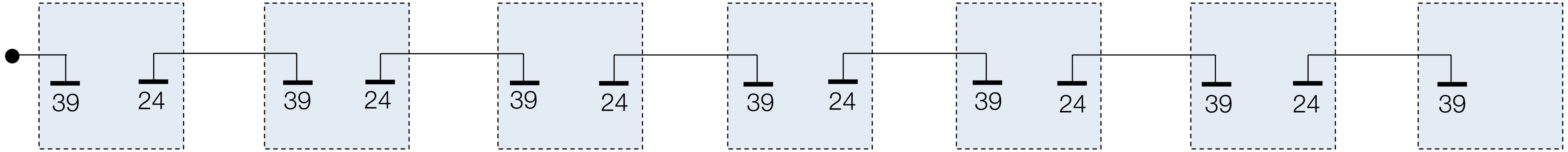
$$\frac{H_i}{\pi f_\mathrm{b}}\dot{\omega}_i = -D_i\omega_i + P_{\mathrm{m}_i}$$

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

Setup from *[Susuki et al, 2012]*

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

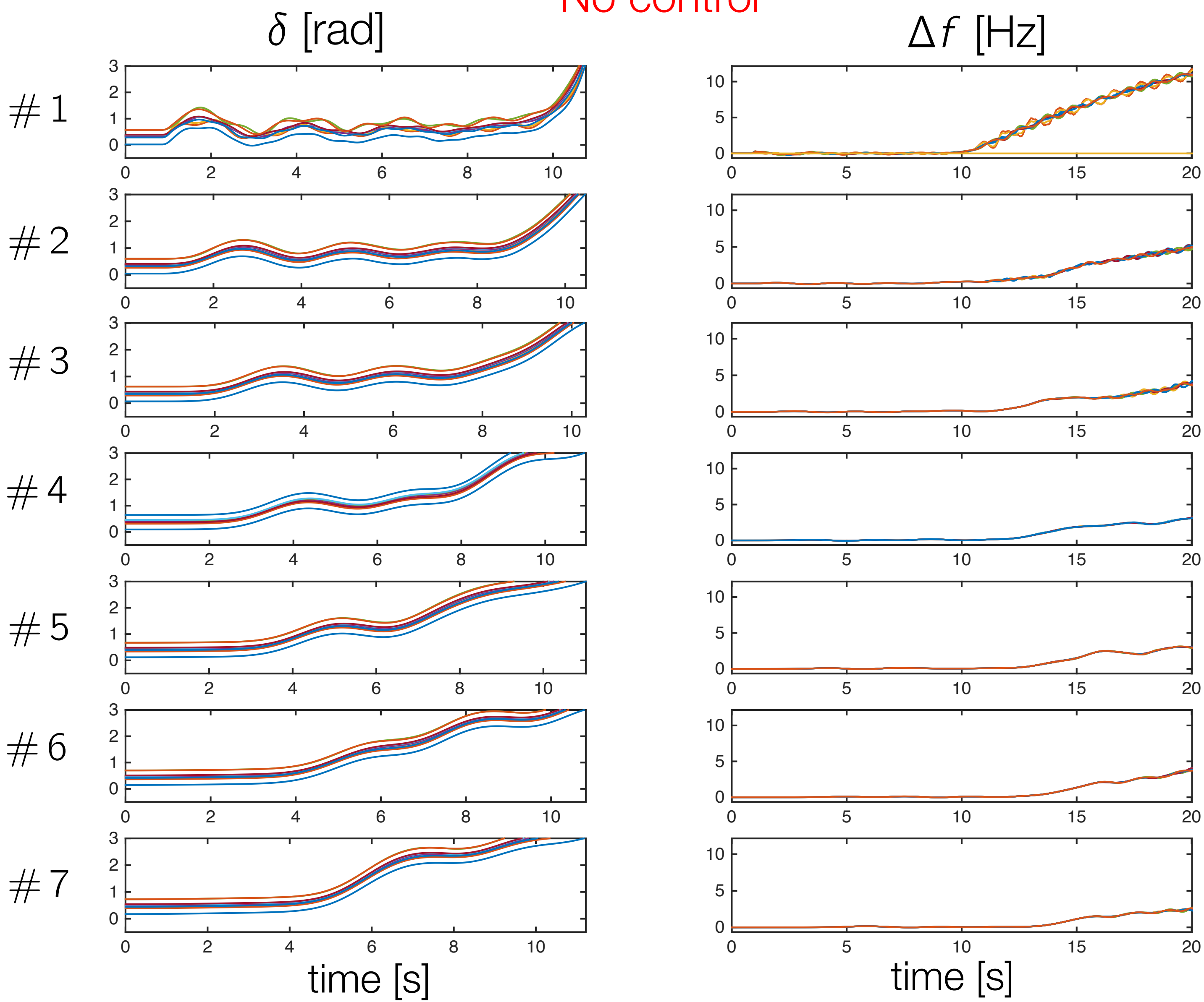
$t = 1\,\mathrm{s}$ – faulted line removed



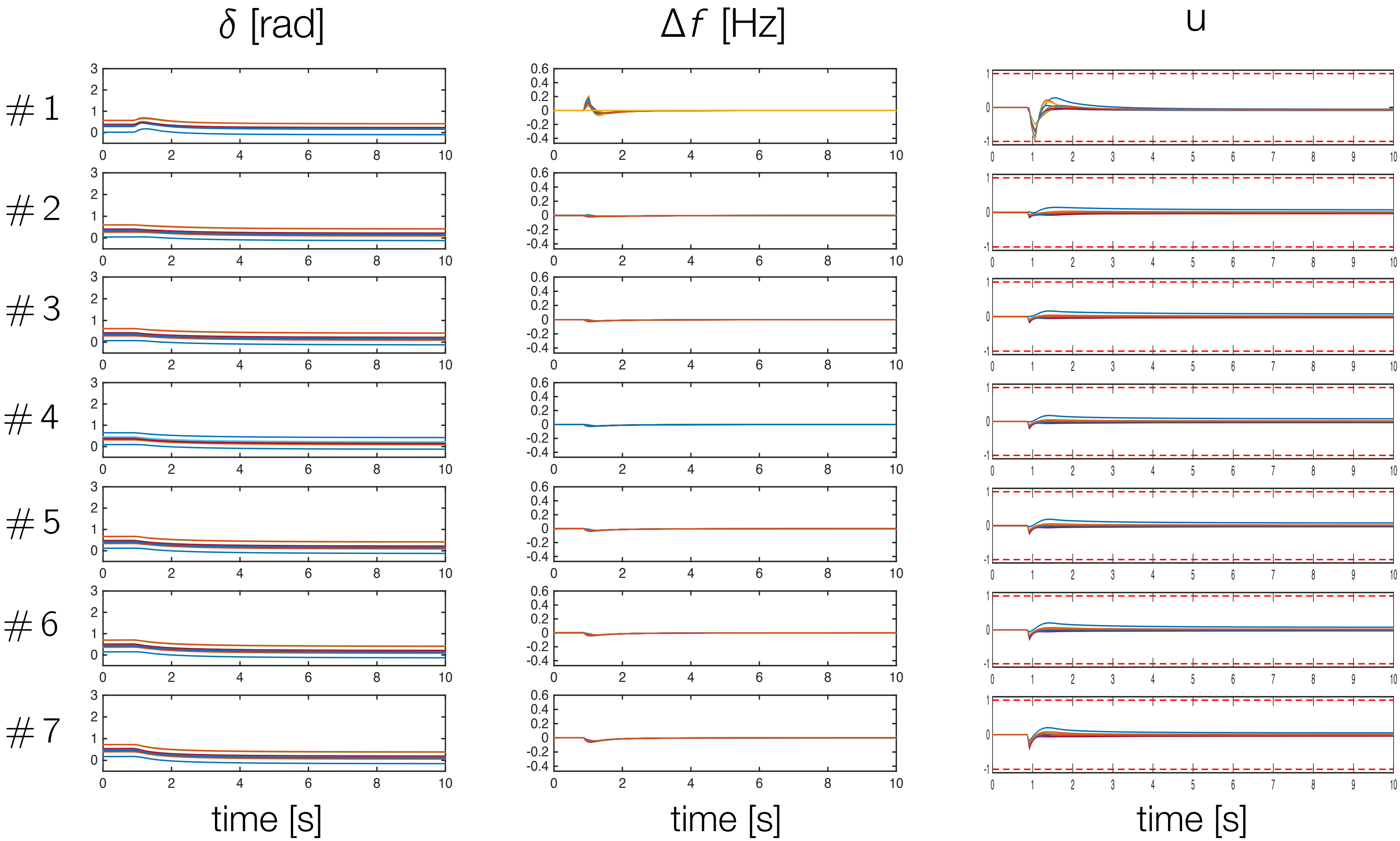
Each grid controlled separately

Numerical examples - NE cascade

No control



Numerical examples - NE cascade



Numerical examples – NE cascade

7 NE grid cascade

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

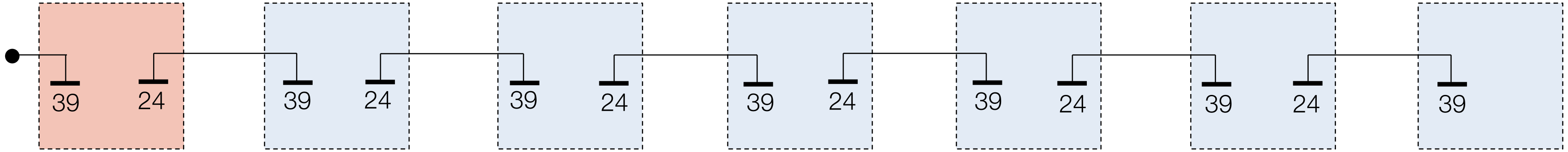
$$\frac{H_i}{\pi f_\mathrm{b}}\dot{\omega}_i = -D_i\omega_i + P_{\mathrm{m}_i}$$

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

Setup from *[Susuki et al, 2012]*

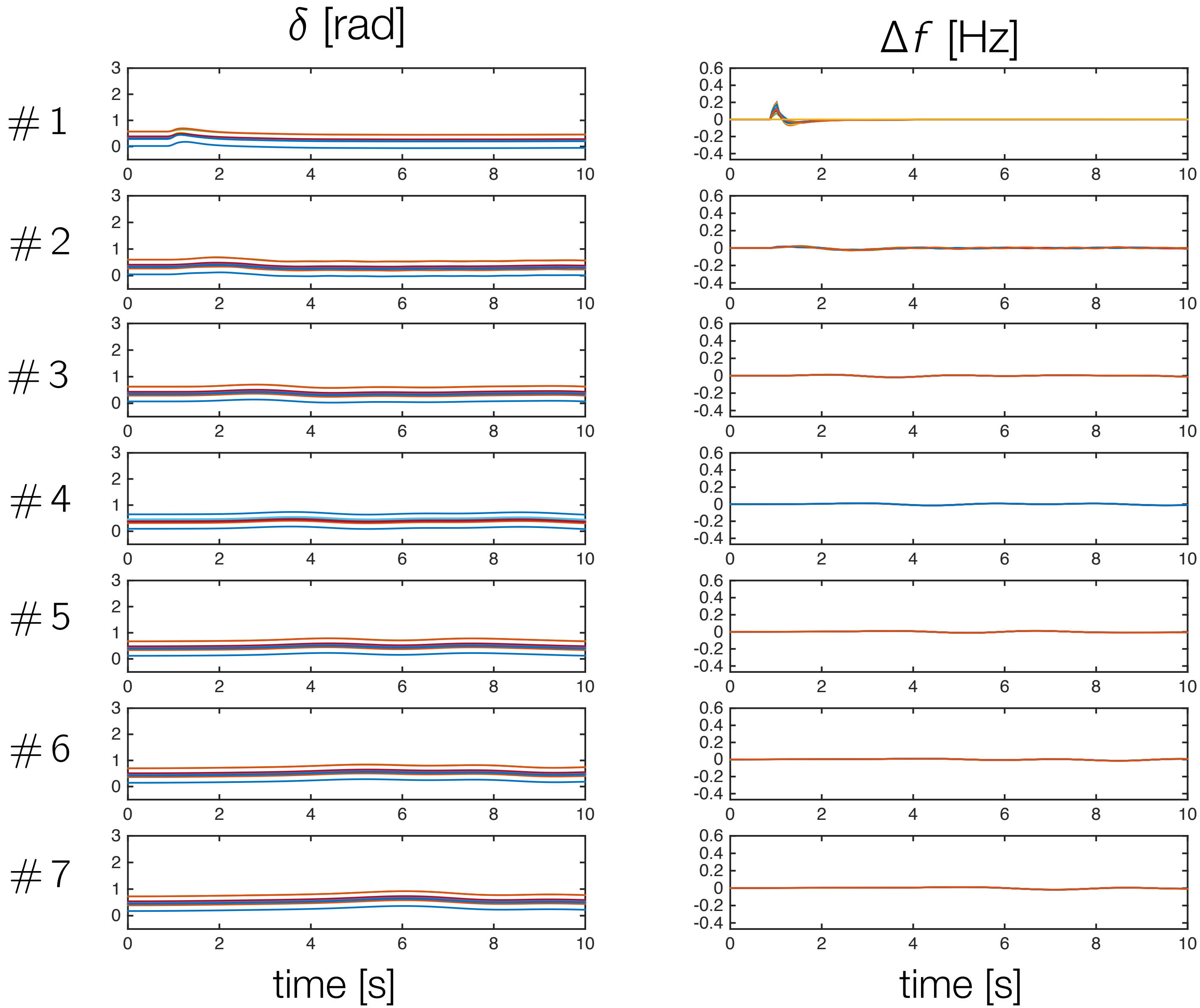
$t = 0.87\,\mathrm{s}$ – fault occurs in grid $\#1$

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

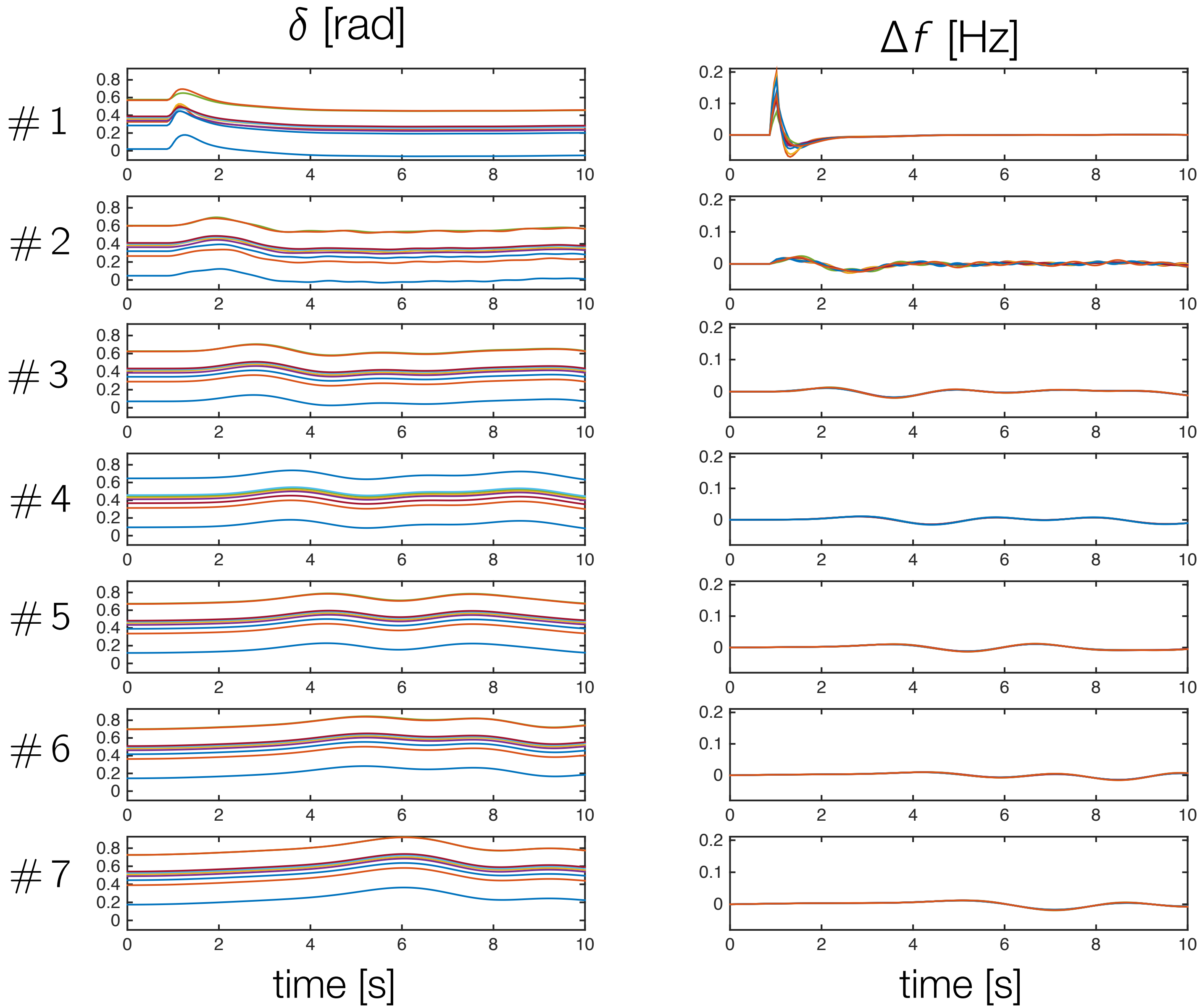


Only the grid where the fault occurred controlled

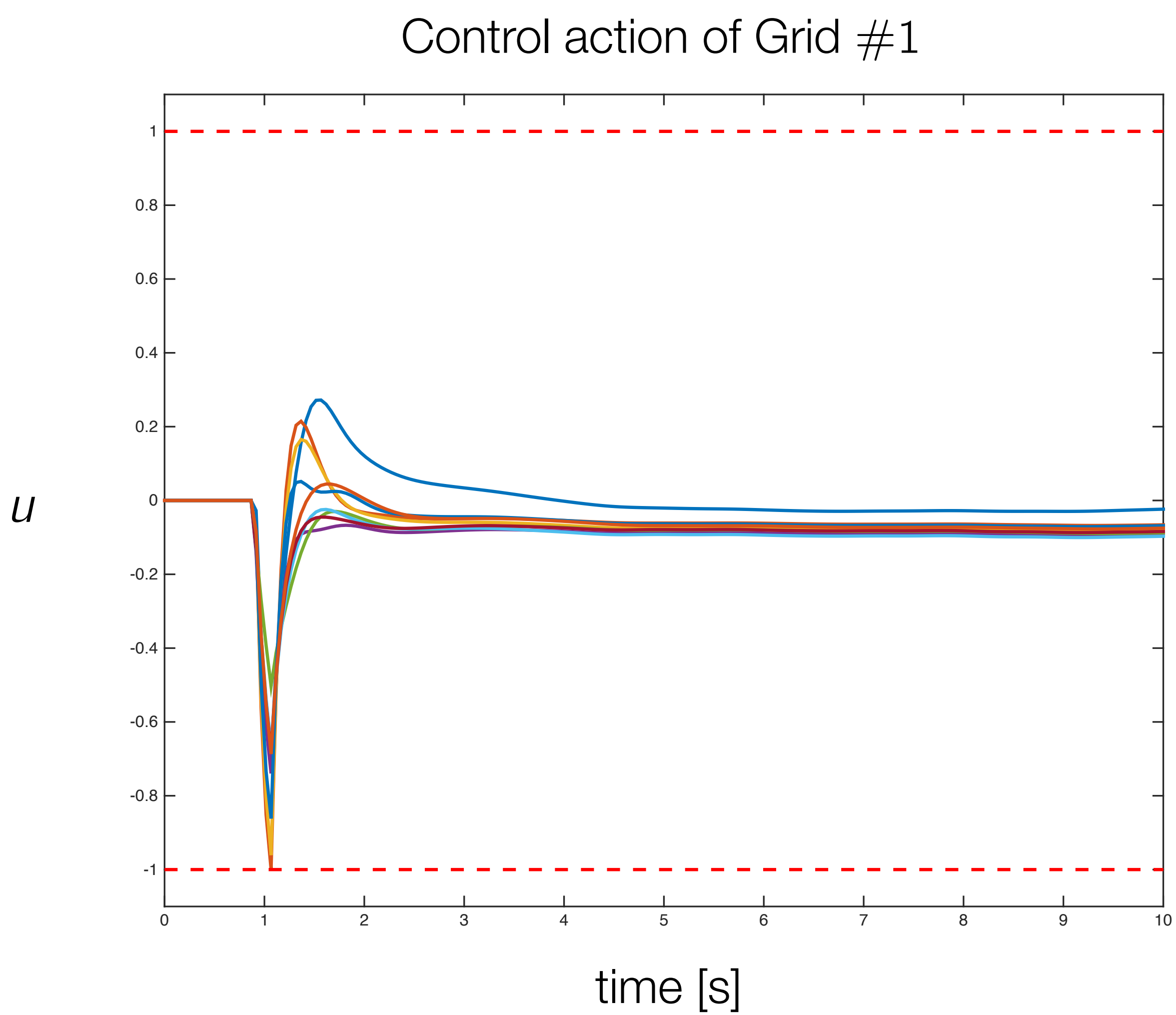
Numerical examples - NE cascade



Numerical examples - NE cascade



Numerical examples - NE cascade



Numerical examples - NE cascade

Computation time $\approx 10\text{ms}$ per grid

(Matlab + qpOASES, 2GHz i7)

PDE control

(Joint work with Hassan Arbabi)

Numerical examples

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]*

Numerical examples

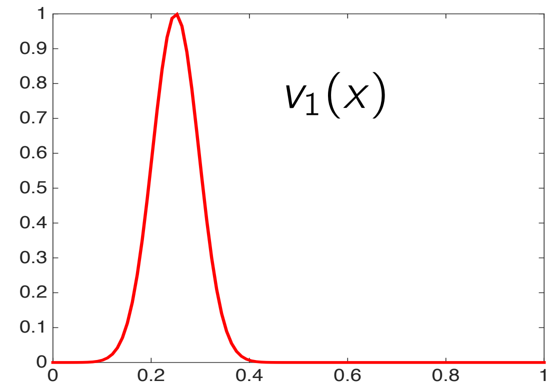
Burgers' equation

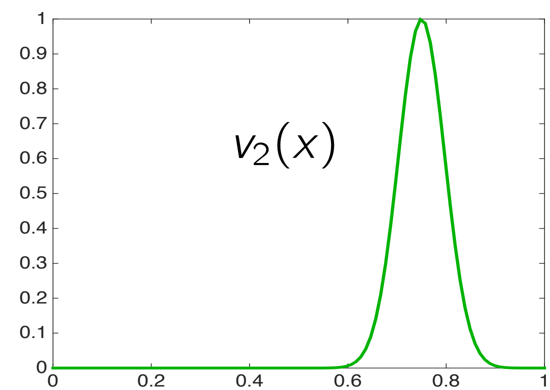
$$\frac{\partial y}{\partial t} - \nu \frac{\partial^2 y}{\partial x^2} + y \frac{\partial y}{\partial x} = u(t,x)$$

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

Setup from *[Peitz, Klus 2017]*

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





Numerical examples

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

Numerical examples

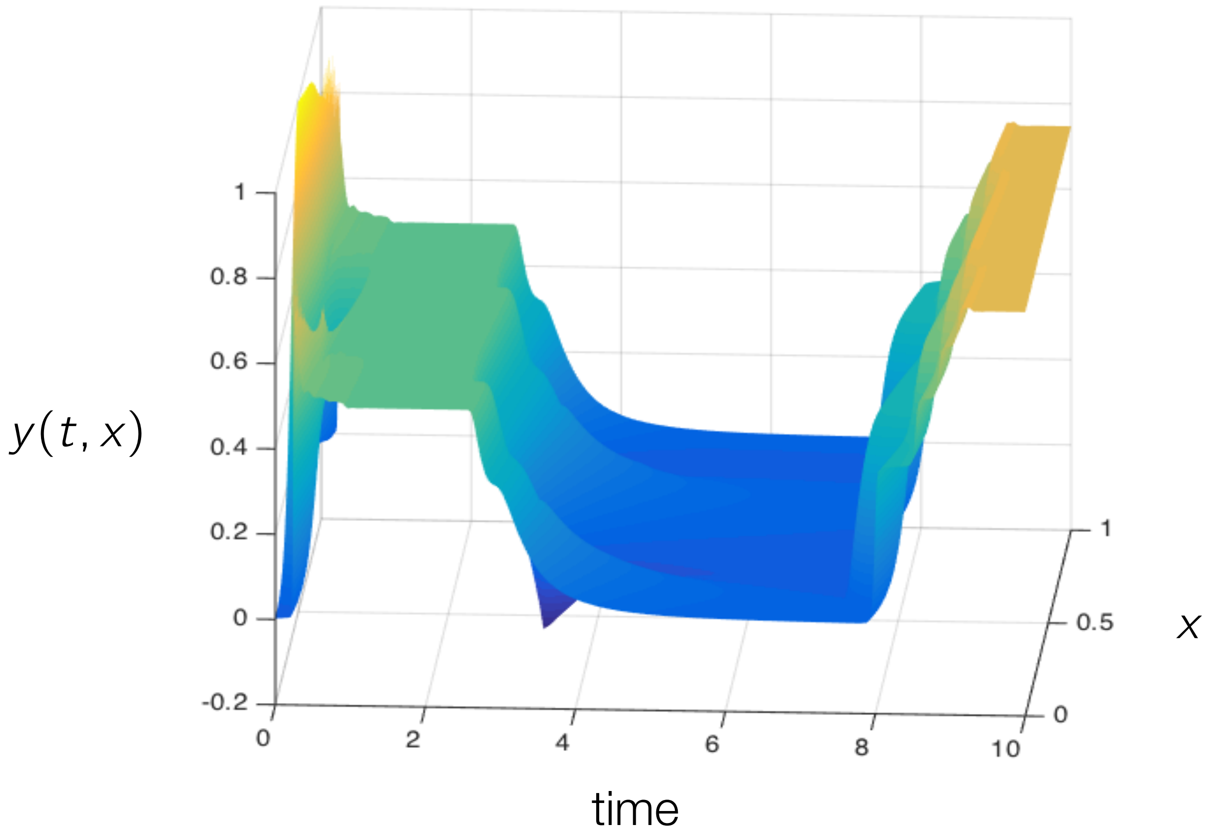
Burgers' equation

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

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

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

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



Numerical examples

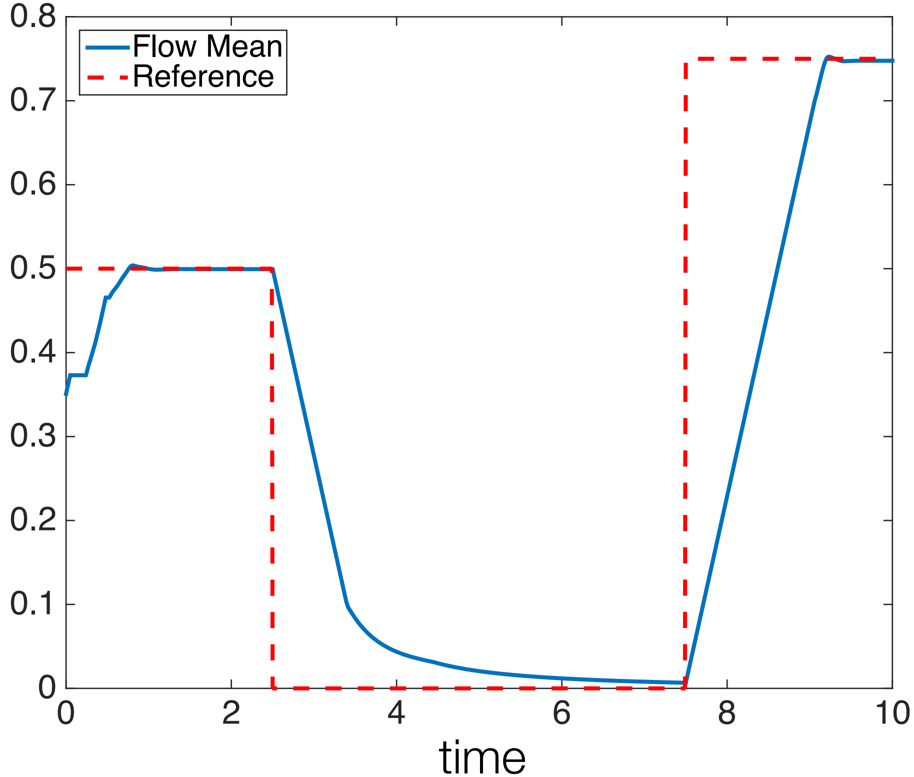
Burgers' equation

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

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

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

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



Numerical examples

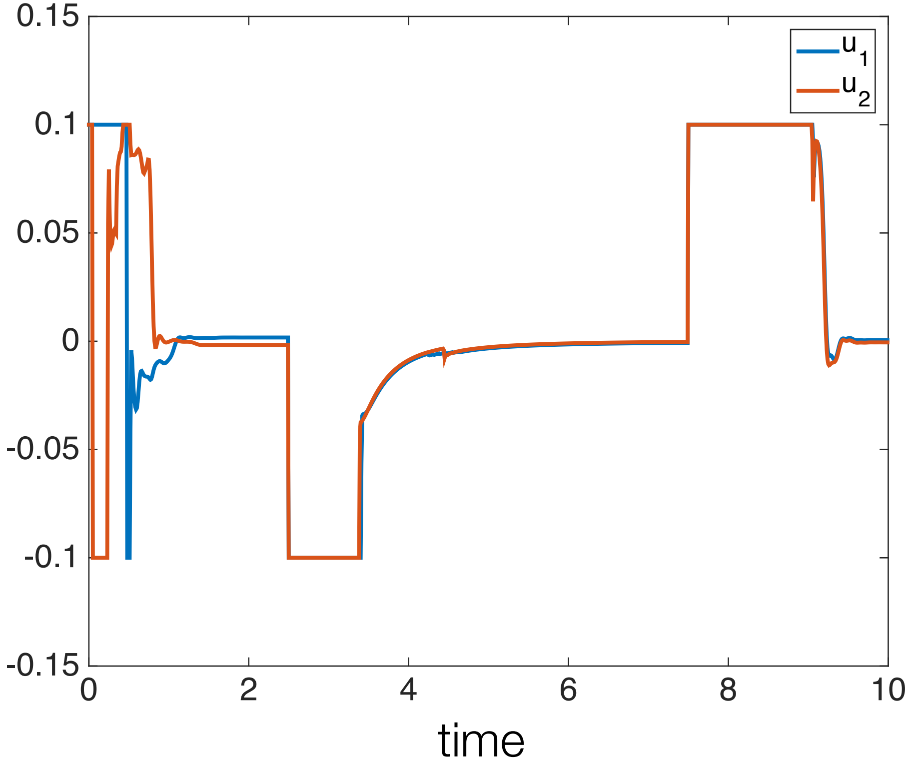
Burgers' equation

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

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

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

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



Numerical examples

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

Tracking time-varying reference

Numerical examples

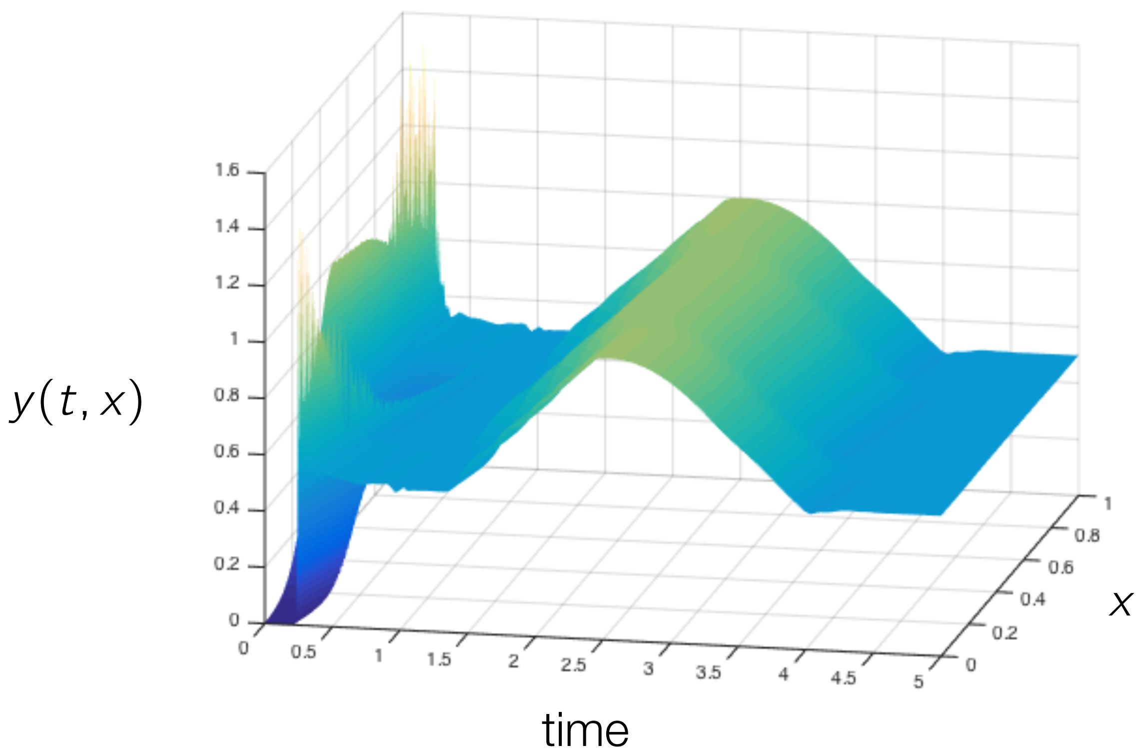
Burgers' equation

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

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

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

$$|u_1(t)| \le 1\,, \quad |u_2(t)| \le 1$$



Numerical examples

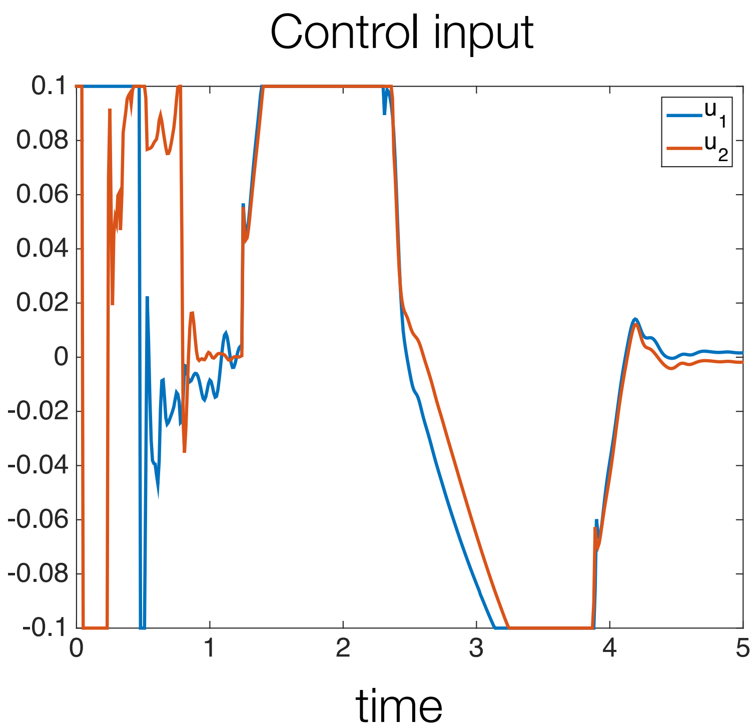
Burgers' equation

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

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

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

$$|u_1(t)| \leq 1\ , \quad |u_2(t)| \leq 1$$



Numerical examples

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

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

$$|u_1(t)| \le 1\,, \quad |u_2(t)| \le 1$$

Computation time $\approx$ 2ms

(Matlab + qpOASES, 2GHz i7)

Open problems

- Accuracy of the predictors for finite N
- Choice of observables
- Guarantees on the controllers (stability, optimality)
- Control for other classes of predictors (bilinear)

arXiv preprint:

Korda M. and Mezić I. *Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control*, arXiv, 2017

Question time

Thank you