



Towards data-driven control of general nonlinear systems with stability and performance guarantees

EPFL Automatic Control Lab Seminar, October 27th 2023

Chris Verhoek, Roland Tóth



Control Systems group, Dept. Electrical Engineering, Eindhoven University of Technology, The Netherlands

A few things about me

Eindhoven University of Technology (TU/e)

MSc in Systems & Control (TU/e)

PhD @ Control Systems group (EE) since Feb. '21

- Roland Tóth & Sofie Haesaert



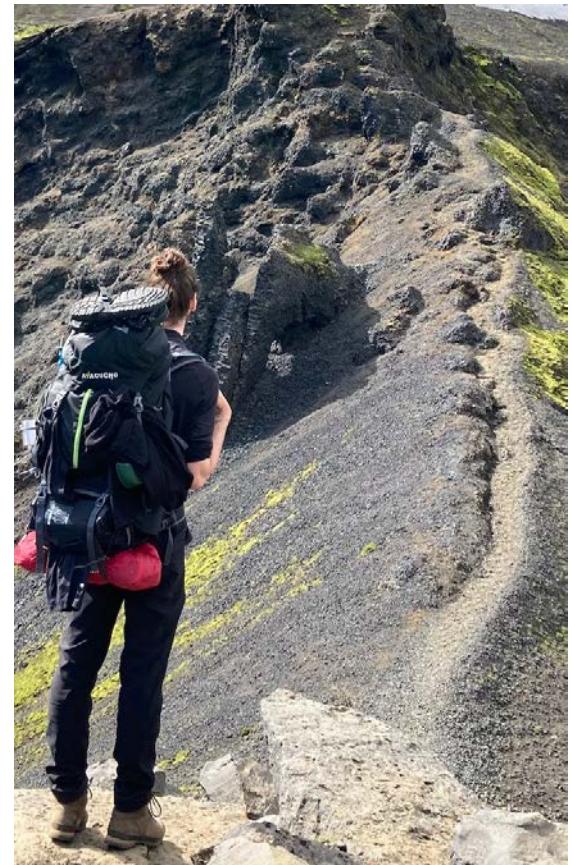
A few things about me



Hiking (multi-day trails)

Drumming (jazz)

Swimming



Motivation

Dynamic systems in engineering

- Increasing performance requirements
- Surge of system complexity
- Nonlinear ([NL](#)) behavior is becoming dominant

Industrial practice:

- Linear Time-Invariant ([LTI](#)) framework
 - Systematic tools for shaping performance
 - Small operating range
- Need for an [NL](#) framework
 - Stability guarantees, but (in general) no performance shaping
 - Non-convex, cumbersome tools

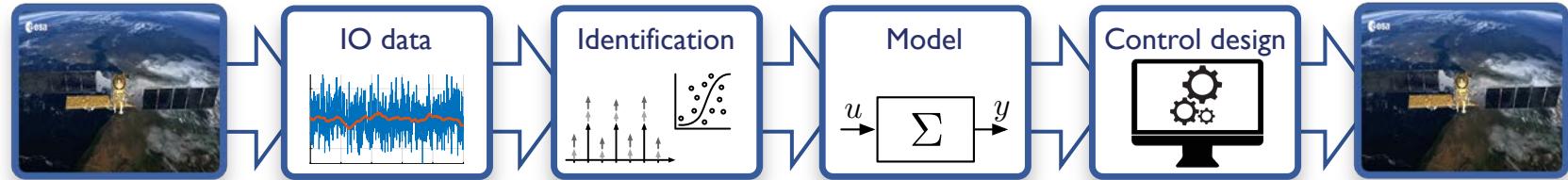


Toolchain based on models

First-principles modeling + model-based control

- Control design with stab. & perf. guarantees ([LTI](#), [LPV](#), etc.)
- Complex, inaccurate, costly modelling
- Effect of unmodelled dynamics on the design

Identify system model + model-based control



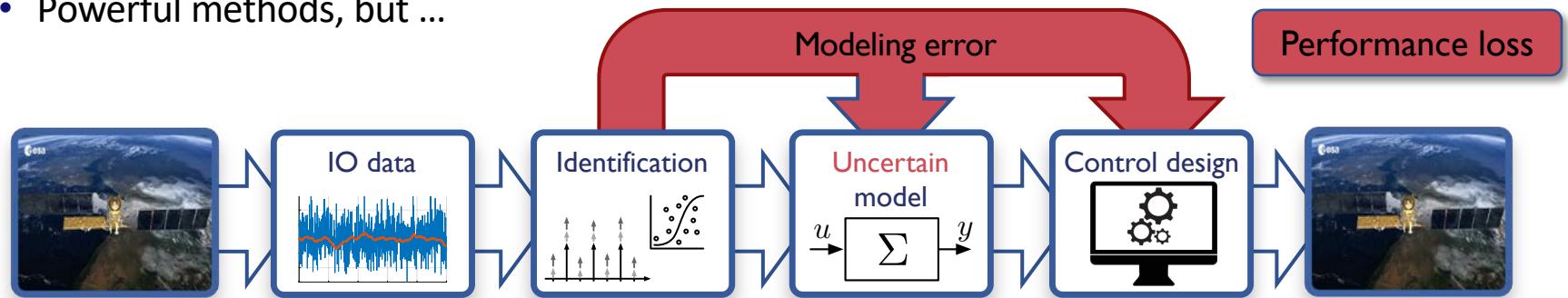
Toolchain based on models

First-principles modeling + model-based control

- Control design with stab. & perf. guarantees ([LTI](#), [LPV](#), etc.)
- Complex, inaccurate, costly modelling
- Effect of unmodelled dynamics on the design

Identify system model + model-based control

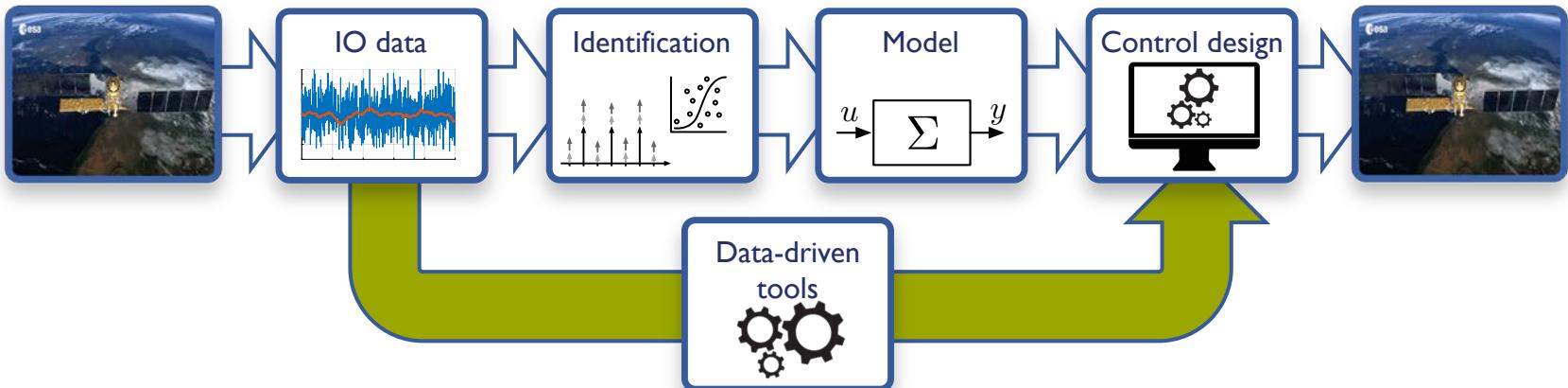
- Powerful methods, but ...



Direct data-driven control

Direct data-driven analysis and control design

- Joint design with guarantees
- Promising approaches



Direct data-driven control

LTI approaches

- Frequency-domain methods
 - PID tuning [1]
 - Nyquist stability (conservative) [2]
 - Nyquist stability (necessary & sufficient) [3]
 - MIMO stab. through approximation [4]
- Time-domain methods
 - Virtual-feedback reference tuning (VFRT) [5]
 - Non-iterative correlation-based tuning (CbT) [6]
 - Behavioral methods [7,8]
- Many more ...

LPV approaches

- Time-domain approaches
 - VFRT methods [9]
- Frequency-domain
 - Nyquist-based, conservative [10]
- Behavioral [12]

NL approaches

- Sector bounded static nonlinearities [13]
- Behavioral (LTI+, Wiener & Ham.) [14-16]

How to address NL systems systematically and give guarantees?

[1] K. Aström, et al., ECC, 2013

[2] S. Khadraoui, et al., Automatica, 2014

[3] A. Karimi et al., Int. J. Rob. Cont., 2018

[4] A. Karimi et al., Automatica 2017

[5] M. Campi, et al., Automatica, 2002

[6] van Heusden, et al. Int. J. ACDS, 2011

[7] Markovsky, Dörfler, Ann.R. Cont., 2022

[8] Coulson, et al., ECC, 2019

[9] Formentin et al. Automatica, 2016

[10] Kunze et. al, ECC , 2007

[11] Bloemers et. al., IEEE-LCSS, 2022

[12] Verhoek et.al., IEEE TAC 2022

[13] Nicoletti et al., J. Rob. Cont., 2018

[14] Alsalti ,et. al., IEEE TAC, 2023

[15] Mishra, et. al., ESPC, 2021

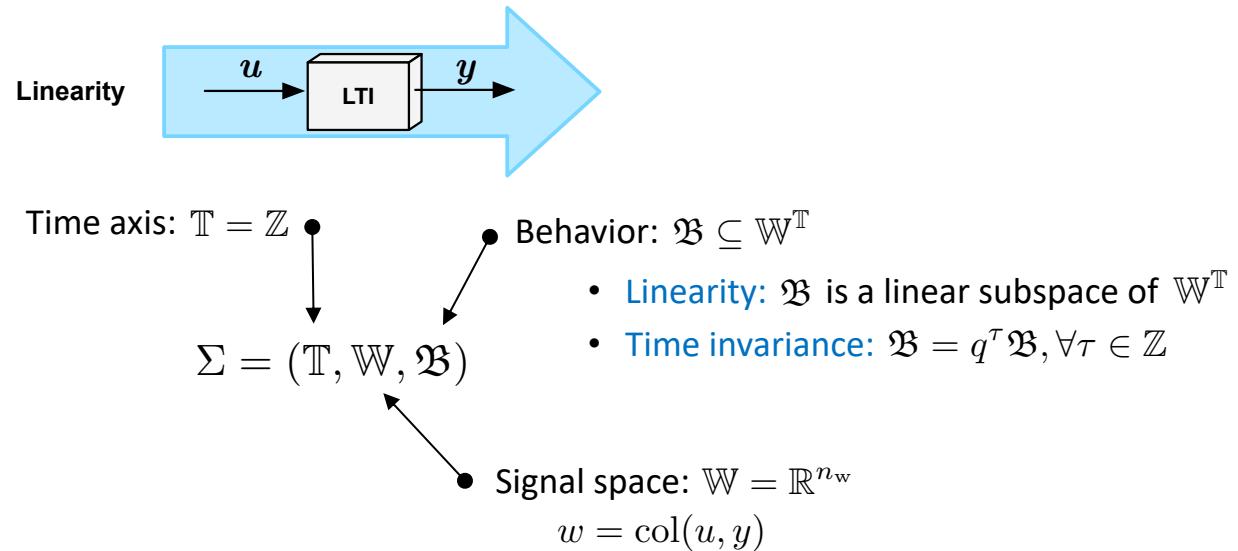
[16] Berberich, et. al., ECC, 2021

Table of contents

- Behavioral LTI data-driven control
 - LTI behavioral theory
 - Data-driven LTI behavioral representation
 - Data-driven LTI behavioral control
- Behavioral LPV data-driven control
- Behavioral NL data-driven control
- Conclusions

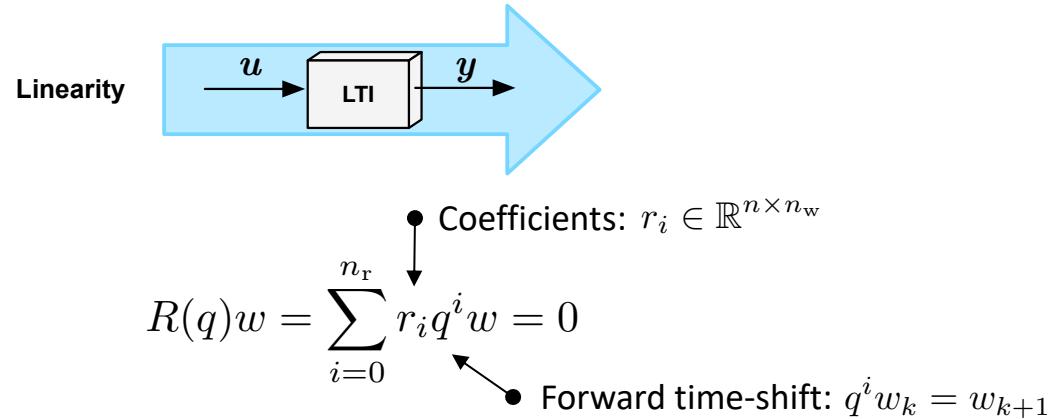
LTI behavioral theory

Behavioral concept (discrete time)



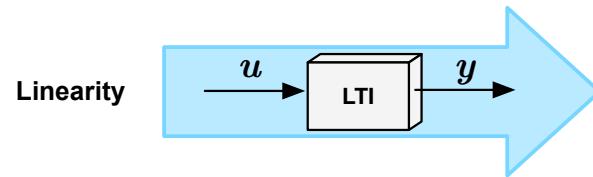
LTI behavioral theory

Kernel representation (discrete time)



LTI behavioral theory

Kernel representation (discrete time)



$$R(q)w = \sum_{i=0}^{n_r} r_i q^i w = 0$$

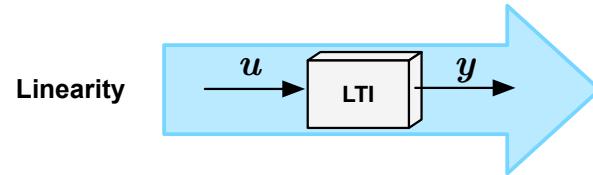
Existence of full row-rank kernel representation

is the representation of the LTI system $\Sigma = (\mathbb{T}, \mathbb{W}, \mathfrak{B})$ if

$$\mathfrak{B} = \{w \in (\mathbb{R}^{n_w})^{\mathbb{Z}} \mid R(q)w = 0\}$$

Data-driven LTI behavioral representation

Data-driven representation (discrete time)

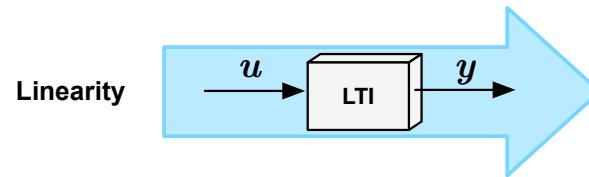


$$\mathcal{D}_N = \left\{ \underbrace{\underline{u}_k^d, \underline{y}_k^d}_{w_k^d} \right\}_{k=1}^N$$

(data dictionary)

Data-driven LTI behavioral representation

Data-driven representation (discrete time)



Hankel matrix (L-row):

$$\mathcal{H}_L(w^d) = \begin{bmatrix} w_1^d & w_2^d & \cdots & w_{N-L+1}^d \\ w_2^d & w_3^d & \cdots & w_{N-L+2}^d \\ \vdots & \vdots & \ddots & \vdots \\ w_L^d & w_{L+1}^d & \cdots & w_N^d \end{bmatrix}$$

(data dictionary)

Willems' Fundamental Lemma [17]:

$$\text{span}^{\text{col}}(\mathcal{H}_L(w^d)) = \mathfrak{B}|_{[1,L]}$$

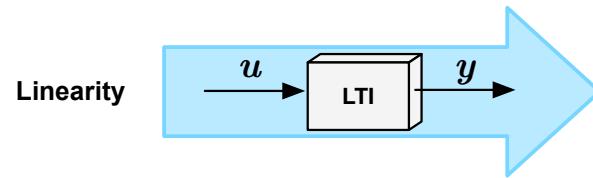
$$\text{if } \text{rank}(\mathcal{H}_L(u^d)) = n_u L$$

(Persistency of excitation)

$$N \geq (n_u + 1)L - 1$$

Data-driven LTI behavioral representation

Data-driven representation (discrete time)



Hankel matrix (L-row):

$$\mathcal{H}_L(w^d) = \begin{bmatrix} w_1^d & w_2^d & \cdots & w_{N-L+1}^d \\ w_2^d & w_3^d & \cdots & w_{N-L+2}^d \\ \vdots & \vdots & \ddots & \vdots \\ w_L^d & w_{L+1}^d & \cdots & w_N^d \end{bmatrix}$$

(data dictionary)

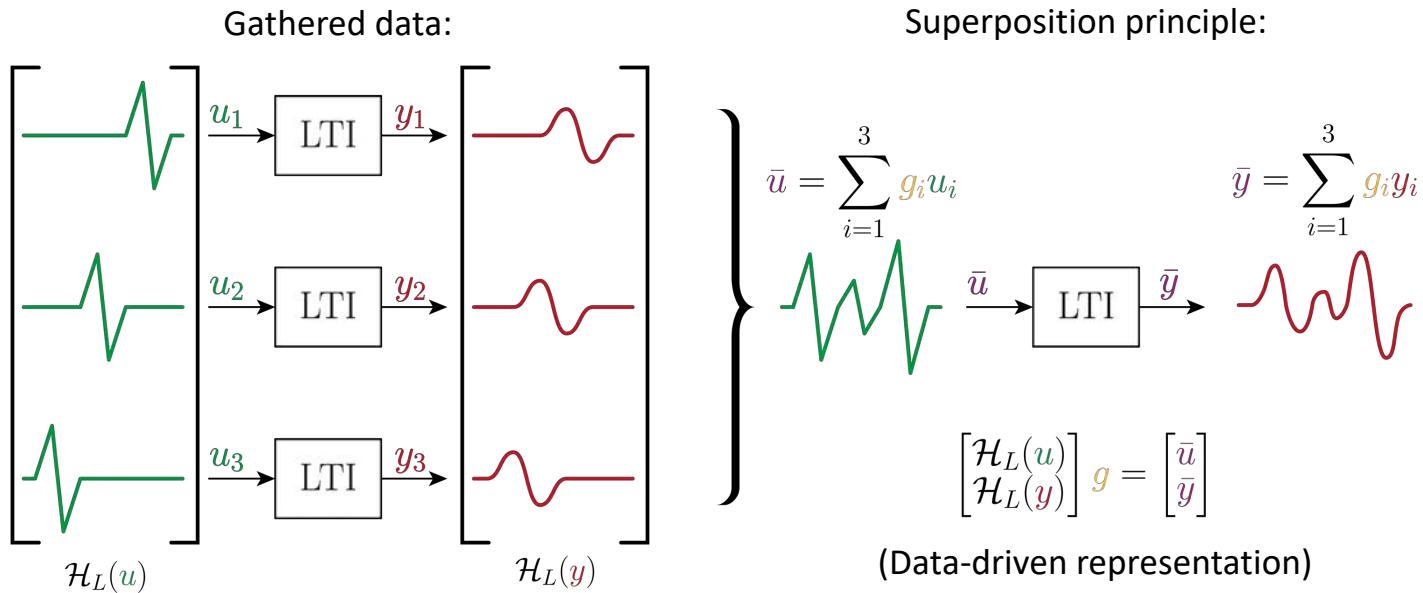


Data-driven representation:

$$\exists g \in \mathbb{R}^{N-L+1}$$
$$\begin{bmatrix} \mathcal{H}_L(u^d) \\ \mathcal{H}_L(y^d) \end{bmatrix} g = \begin{bmatrix} \text{col}(\bar{u}_{[1,L]}) \\ \text{col}(\bar{y}_{[1,L]}) \end{bmatrix}$$
$$\Downarrow$$
$$\text{col}(\bar{y}_{[1,L]}, \bar{u}_{[1,L]}) \in \mathfrak{B}_{[1,L]}$$

Data-driven LTI behavioral representation

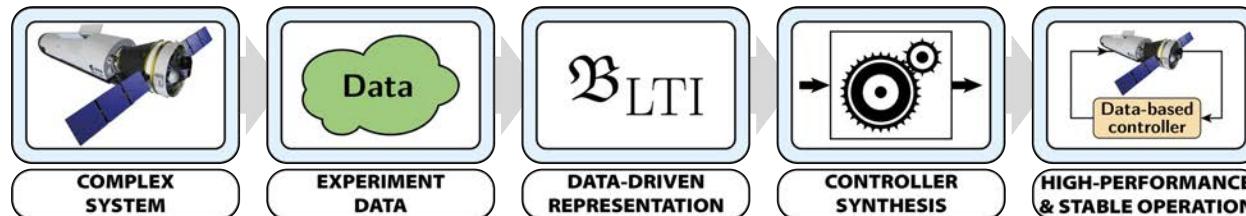
Data-driven representation (discrete time)



Data-driven LTI behavioral control

Direct data-driven analysis and control design

- Analysis
 - Simulation (Data spans the full behavior of length L) [7]
 - Stability & performance analysis (dissipativity, quadratic perf., etc.) [18]
- Control
 - Predictive control schemes (e.g., DeePC [8])
 - State-feedback control [7]
 - Noise handling & robustness guarantees [19]



[7] Markovsky, et.al.: Data-driven simulation and control, *Int. Journal of Control.*, (2008)

[8] Coulson, et.al.: Data-Enabled Predictive Control: In the Shallows of the DeePC, in *Proc. of the ECC*, (2019)

[18] Romer et al.: One-shot verification of dissipativity properties from input-output data, *Control Systems Letters*, (2019)

[19] Berberich et al.: Data-Driven Model Predictive Control With Stability and Robustness Guarantees, *IEEE TAC*, (2021)

Table of contents

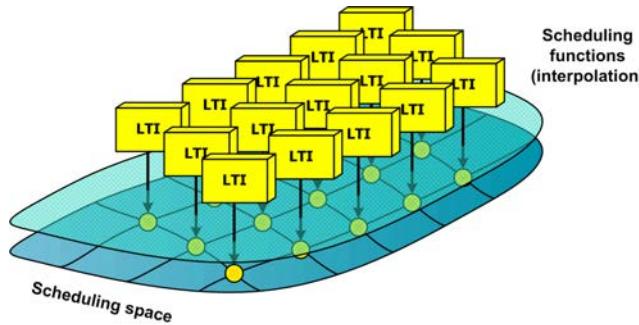
- Behavioral LTI data-driven control
- Behavioral LPV data-driven control
 - LPV behavioral theory
 - Data-driven LPV behavioral representation
 - Simplified LPV Fundamental Lemma
 - Data-driven LPV behavioral control
- Behavioral NL data-driven control
- Conclusions

Linear parameter-varying framework

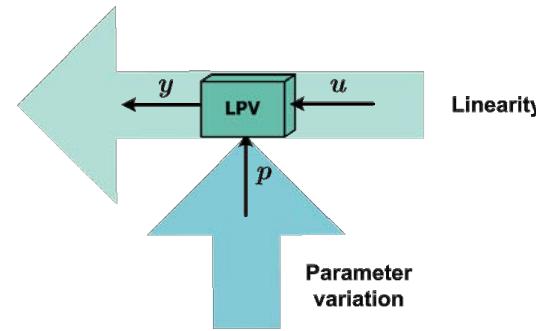
The Engineers' Dream:

How to use "simple" linear control for NL systems with performance guarantees?

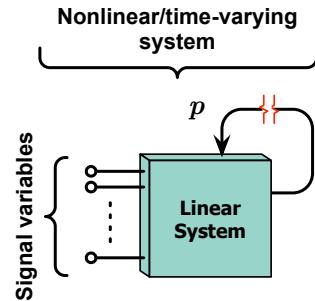
Linear parameter-varying framework



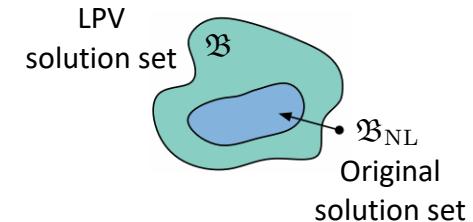
Local approximation principle



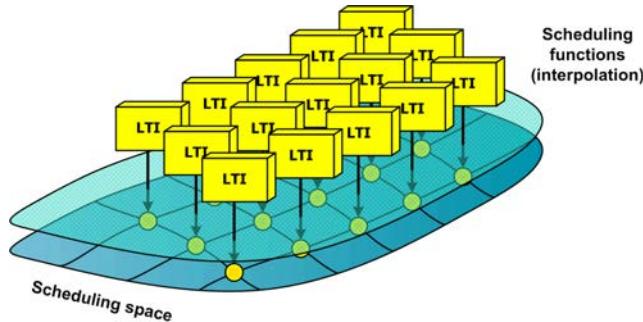
Linearity



Global embedding principle



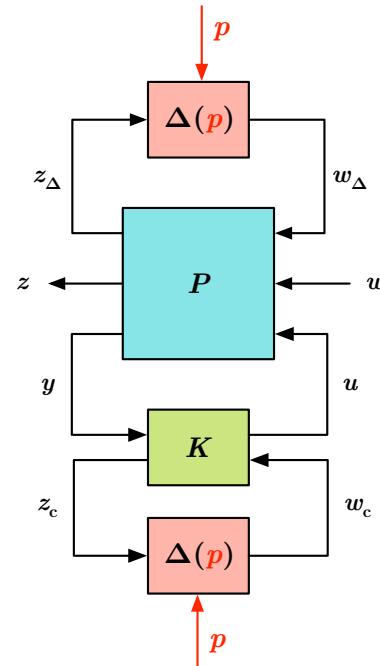
Linear parameter-varying framework



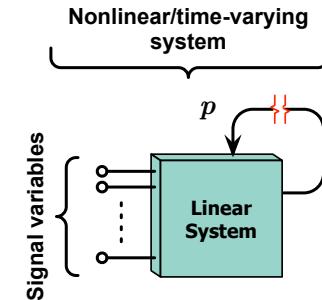
Local approximation principle



Local synthesis:
Gain scheduling
(interpolated LTI control)



Controller Synthesis



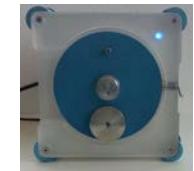
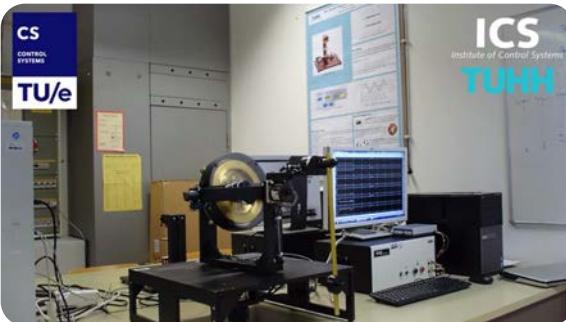
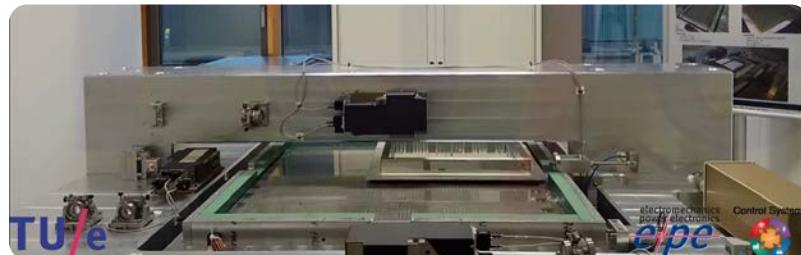
Global embedding principle



Global synthesis: Optimal LPV control (NL control)

Linear parameter-varying framework

A plethora of success stories via model-based control

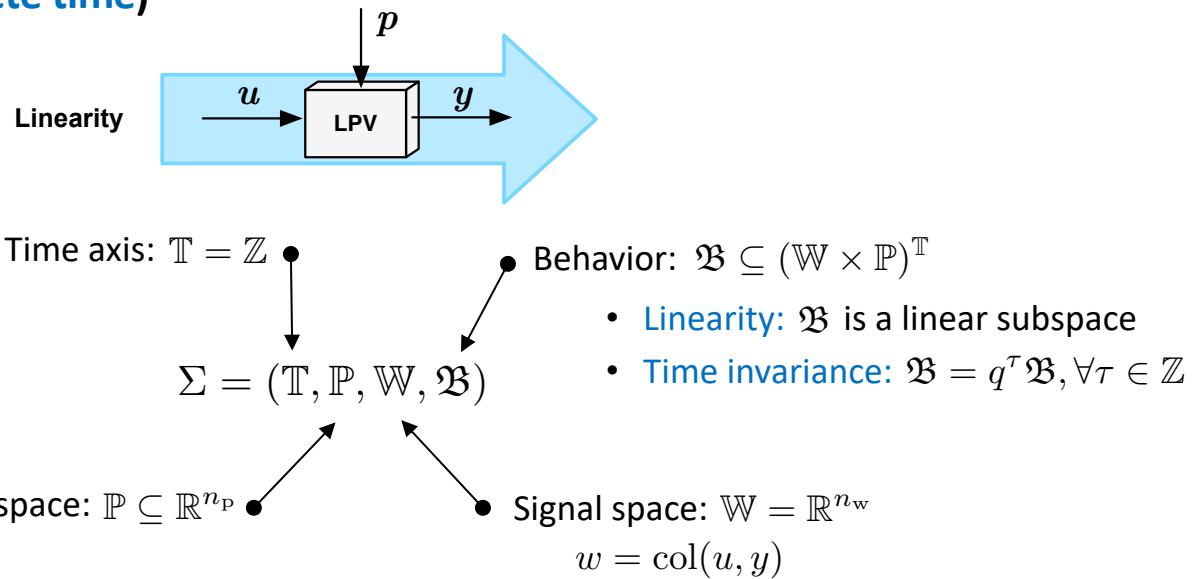


Pending question:

How to achieve data-driven control with guarantees?

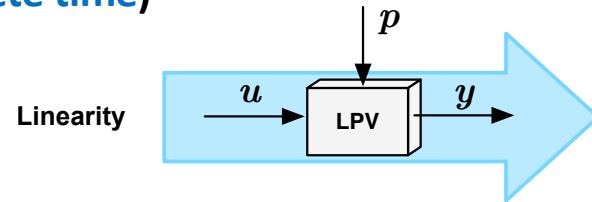
LPV behavioral theory

Behavioral concept (discrete time)



LPV behavioral theory

Behavioral concept (discrete time)



$$\Sigma = (\mathbb{T}, \mathbb{P}, \mathbb{W}, \mathfrak{B})$$

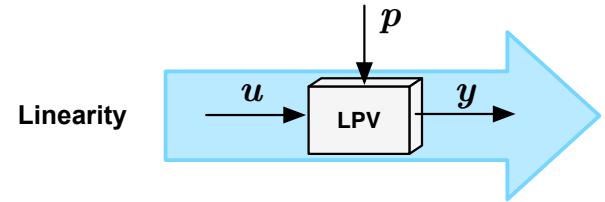
- Projected scheduling behavior:

$$\mathfrak{B}_{\mathbb{P}} = \pi_p \mathfrak{B} := \{p \in \mathbb{P}^{\mathbb{T}} \mid \exists w \in \mathbb{W}^{\mathbb{T}} \text{ s.t. } (w, p) \in \mathfrak{B}\}$$

- Projected behavior for a given: $p \in \mathfrak{B}_{\mathbb{P}}$

$$\mathfrak{B}_p = \{w \in \mathbb{W}^{\mathbb{T}} \mid (w, p) \in \mathfrak{B}\}$$

LPV behavioral theory



Kernel representation (discrete time)

Coefficient functions:

$$r_i : \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n \times n_w}$$

Types (static dep.):

- Affine/linear functions
- Polynomial functions
- Rational functions
- Meromorphic functions

$$r(\cdot) = \frac{g(\cdot)}{h(\cdot)}$$

holomorphic
 $h \neq 0$

Meromorphic field

$$r_i \in \mathcal{R}^{n \times n_w}$$

$$\sum_{i=0}^{n_r} r_i(p_k) q^i w_k = 0$$

Shift operator:
 $q^i w_k = w_{k+1}$

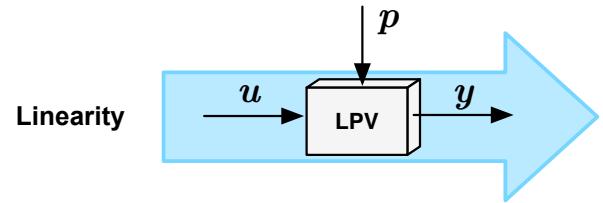
Confined in
 $\mathbb{P} \subseteq \mathbb{R}^{n_p}$

Signals:
 $w : \mathbb{Z} \rightarrow \mathbb{R}^{n_w}$

Commutation problem (dynamic dependence):

$$q^i r_i(p_k) w_k \neq r_i(p_k) w_{k+i}$$

LPV behavioral theory



Kernel representation (discrete time)

Coefficient functions:

$$r_i : \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{n \times n_w}$$

Types (static dep.):

- Affine/linear functions
- Polynomial functions
- Rational functions
- Meromorphic functions

$$r(\cdot) = \frac{g(\cdot)}{h(\cdot)}$$

holomorphic
 $h \neq 0$

Meromorphic field

$$r_i \in \mathcal{R}^{n \times n_w}$$

$$\sum_{i=0}^{n_r} r_i(p_k) q^i w_k = 0$$

Shift operator:
 $q^i w_k = w_{k+1}$

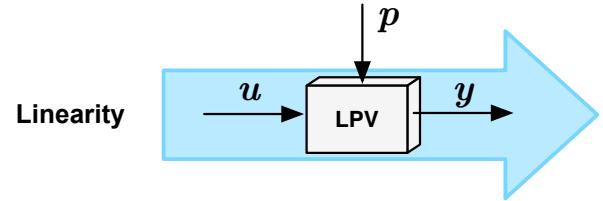
Confined in
 $\mathbb{P} \subseteq \mathbb{R}^{n_p}$

Signals:
 $w : \mathbb{Z} \rightarrow \mathbb{R}^{n_w}$

Commutation problem (dynamic dependence):

$$q^i r_i(p_k) w_k = r_i(p_{k+i}) w_{k+i}$$

LPV behavioral theory



Kernel representation (discrete time)

Coefficient functions with finite dynamic dependence

Features:

- Causal

$$r_i(p_k, p_{k-1}, p_{k-2}, \dots)$$

- Non-causal

$$r_i(\dots, p_{k+1}, p_k, p_{k-1}, \dots)$$

$$\sum_{i=0}^{n_r} (r_i \diamond p)_k q^i w_k = 0$$

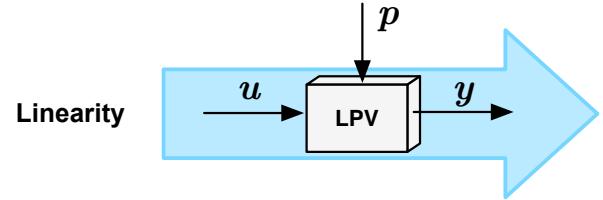
$\underbrace{}_{R(q) \diamond p}$

Shorthand for evaluation over dynamic dependence

Polynomials over \mathcal{R}

$$R \in \mathcal{R}[\xi]^{n \times n_w}$$

LPV behavioral theory



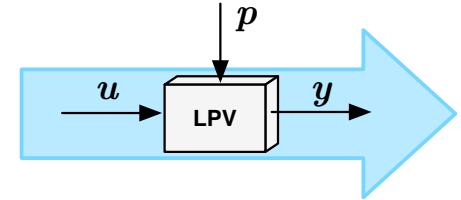
Kernel representation (discrete time)

$$\underbrace{\sum_{i=0}^{n_r} (r_i \diamond \textcolor{red}{p})_k q^i w_k}_{R(q) \diamond p} = 0$$

is the representation of the LPV system $\Sigma = (\mathbb{T}, \mathbb{P}, \mathbb{W}, \mathfrak{B})$ if

$$\mathfrak{B} = \{(w, p) \in (\mathbb{R}^{n_w} \times \mathbb{P})^{\mathbb{Z}} \mid (R(q) \diamond p)w = 0\}$$

Data-driven LPV behavioral representation



Data-driven representation (discrete time)

$$\underbrace{\sum_{i=0}^{n_r} (r_i \diamond p)_k q^i w_k}_{R(q) \diamond p} = 0$$

Data-dictionary:

$$\mathcal{D}_N = \{u_k^d, y_k^d, p_k^d\}_{k=1}^N$$



Complex condition

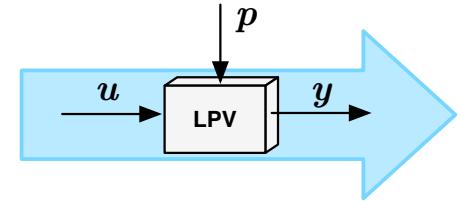
Can we simplify this to an easily computable form / representation?

LPV Fundamental Lemma:

$$\text{span}_{\mathcal{R}, p}^{\text{col}}(\mathcal{H}_L(w^d)) = \mathfrak{B}_p|_{[1, L]}$$

(Persistency of excitation)
existence of a “unique” R w.r.t \mathcal{D}_N .

Data-driven LPV behavioral representation



Data-driven representation (discrete time, simple case)

Consider the IO form (partitioned kernel rep.):

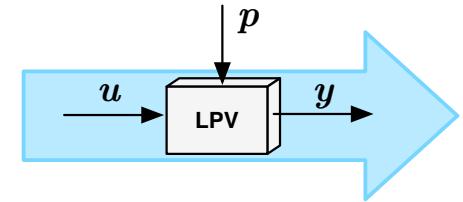
$$y_k + \sum_{i=1}^{n_a} a_i(p_{k-i})y_{k-i} = \sum_{i=1}^{n_b} b_i(p_{k-i})u_{k-i}$$

Restricted, but useful subclass of LPV systems

with shifted-affine scheduling dependence:

$$a_i(p_{k-i}) = a_{i,0} + \sum_{j=1}^{n_p} a_{i,j} p_{j,k-i}, \quad b_i(p_{k-i}) = b_{i,0} + \sum_{j=1}^{n_p} b_{i,j} p_{j,k-i}$$

Data-driven LPV behavioral representation



Data-driven representation (discrete time, simple case)

$$y_k + \sum_{i=1}^{n_a} \underbrace{a_i(p_{k-i}) y_{k-i}}_{p_{k-i} \otimes y_{k-i}} = \sum_{i=1}^{n_b} \underbrace{b_i(p_{k-i}) u_{k-i}}_{p_{k-i} \otimes u_{k-i}}$$

Data-dictionary:

$$\mathcal{D}_N = \{u_k^d, y_k^d, p_k^d\}_{k=1}^N$$

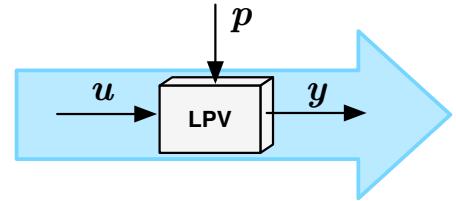


Data-driven representation:

$$\begin{bmatrix} \mathcal{H}_L(u^d) \\ \mathcal{H}_L(p^d \otimes u^d) - \bar{\mathcal{P}}^{n_u} \mathcal{H}_L(u^d) \\ \mathcal{H}_L(y^d) \\ \mathcal{H}_L(p^d \otimes y^d) - \bar{\mathcal{P}}^{n_y} \mathcal{H}_L(y^d) \end{bmatrix} g = \begin{bmatrix} \text{col}(\bar{u}_{[1,L]}) \\ 0 \\ \text{col}(\bar{y}_{[1,L]}) \\ 0 \end{bmatrix}$$

$$\text{col}(\bar{y}_{[1,L]}, \bar{u}_{[1,L]}, \bar{p}_{[1,L]}) \in \mathfrak{B}_{[1,L]}$$

Data-driven LPV behavioral representation



Simplified LPV Fundamental Lemma ([discrete time](#))

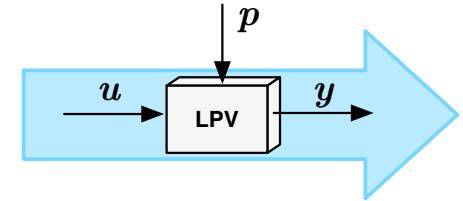
Given $\mathcal{D}_N = \{u_k^d, p_k^d, x_k^d\}_{k=1}^N$ and let

$$\mathcal{N}_{\bar{p}} := \text{nullspace} \left\{ \begin{bmatrix} \mathcal{H}_L(p^d \otimes u^d) - \bar{\mathcal{P}}^{n_u} \mathcal{H}_L(u^d) \\ \mathcal{H}_L(p^d \otimes y^d) - \bar{\mathcal{P}}^{n_y} \mathcal{H}_L(y^d) \end{bmatrix} \right\}, \quad \mathcal{S} := \text{span}^{\text{col}} \left\{ \begin{bmatrix} \mathcal{H}_L(u^d) \\ \mathcal{H}_L(y^d) \end{bmatrix} \right\}$$

Then, for all scheduling signals $\bar{p} \in \mathfrak{B}_{\mathbb{P}}$

$$\text{Proj}_{\mathcal{N}_{\bar{p}}}(\mathcal{S}) = \mathfrak{B}_{\bar{p}}|_{[1,L]} \iff \dim \left\{ \text{Proj}_{\mathcal{N}_{\bar{p}}}(\mathcal{S}) \right\} = n_x + n_u L$$

Data-driven LPV behavioral representation



Data-driven representation (**discrete time, state-feedback case**)

Consider the **SS** form:

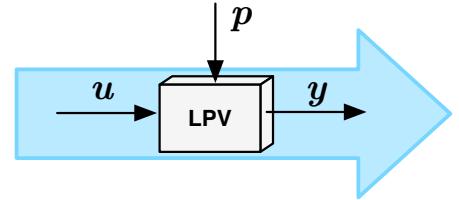
$$x_{k+1} = A(p_k)x_k + B(p_k)u_k$$

$$y_k = x_k$$

with **static-affine** scheduling dependence:

$$A(p_k) = A_0 + \sum_{i=1}^{n_p} A_i p_{i,k}, \quad B(p_k) = B_0 + \sum_{i=1}^{n_p} B_i p_{i,k}$$

Data-driven LPV behavioral representation



Data-driven representation (**discrete time, state-feedback case**)

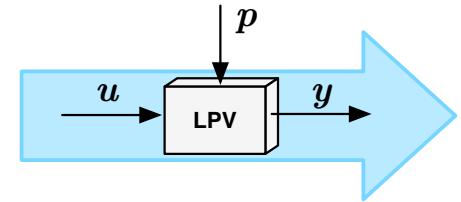
$$x_{k+1} = A(p_k)x_k + B(p_k)u_k,$$

$$y_k = x_k$$

Data-dictionary:

$$\mathcal{D}_N = \{u_k^d, x_k^d, p_k^d\}_{k=1}^N$$

Data-driven LPV behavioral representation



Data-driven representation (discrete time, state-feedback case)

$$\begin{aligned}x_{k+1} &= A(p_k)x_k + B(p_k)u_k, \\ y_k &= x_k\end{aligned}$$

Multi-step: Use direct realization
of the shifted-affine IO form

Data-dictionary:

$$\begin{aligned}U &= [u_1^d \quad \cdots \quad u_{N-1}^d] \\ X &= [x_1^d \quad \cdots \quad x_{N-1}^d] \\ \vec{X} &= [x_2^d \quad \cdots \quad x_N^d] \\ U^p &= [p_1^d \otimes u_1^d \quad \cdots \quad p_{N-1}^d \otimes u_{N-1}^d] \\ X^p &= [p_1^d \otimes x_1^d \quad \cdots \quad p_{N-1}^d \otimes x_{N-1}^d]\end{aligned}$$



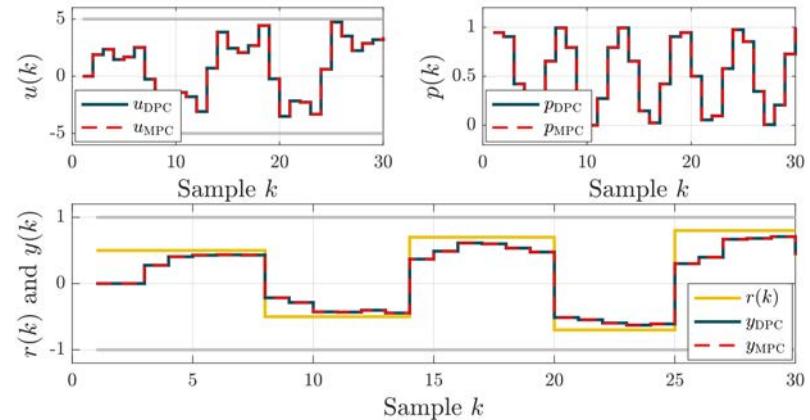
Data-driven representation:

$$\bar{x}_{k+1} = \vec{X}^\dagger \begin{bmatrix} X \\ X^p \\ U \\ U^p \end{bmatrix} \begin{bmatrix} \bar{x}_k \\ \bar{p}_k \otimes \bar{x}_k \\ \bar{u}_k \\ \bar{p}_k \otimes \bar{u}_k \end{bmatrix}$$

Data-driven LPV behavioral control

Direct data-driven analysis and control design

- Analysis
 - Simulation [24]
 - Stability & performance analysis [22] (dissipativity, quadratic perf., etc.)
- Control
 - Predictive control [24, 25]
 - State-feedback control [23]
 - Noise handling & robustness guarantees (coming soon, initial results in [25])

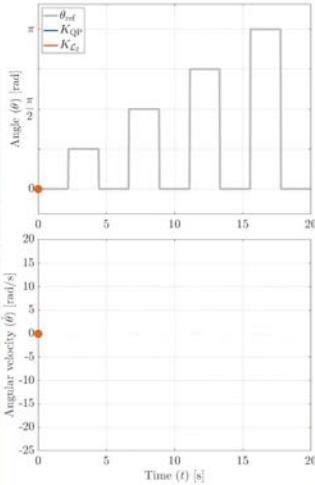


Data-driven vs. model based predictive control

Data-driven LPV behavioral control

Example (unbalanced disc):

K_{QP}



K_{L_2}



Optimal state-
feedback design

Data-driven advantage

LPV controller synthesized using **7** data-points (70 milliseconds)!

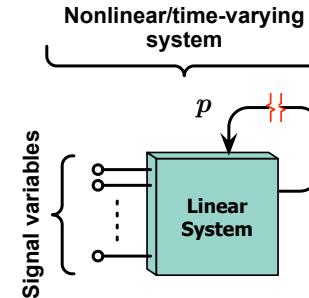
Towards nonlinear data-driven control?

Now developed a data-driven behavioral framework for LPV systems

- When underlying system is NL \rightarrow possible unexpected stability restrictions [27]

Data-driven control for nonlinear systems:

- Feedback/online linearizations [28, 29]
- Nonlinearity cancellation [30]
- Koopman-based [31]
- Polynomial systems [32]



Mostly rely on / go back to **LTI** Fundamental Lemma...

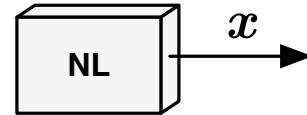
→ Local guarantees
Can we get a bit more general?

Table of contents

- Behavioral LTI data-driven control
- Behavioral LPV data-driven control
- Behavioral NL data-driven control
 - Shifted stability
 - Shifted dissipativity
 - Data-driven NL synthesis with the velocity form
 - Data-driven LPV behavioral control
- Conclusions

Concept of global stability

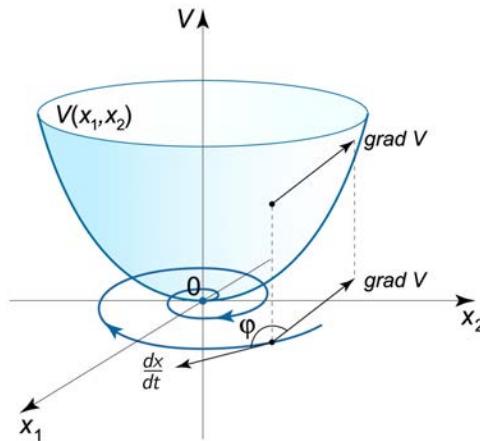
Nonlinear system (**autonomous, discrete time**)



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

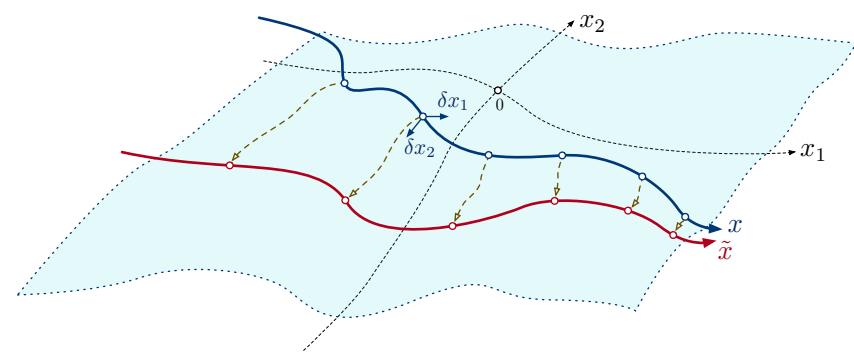
$$\begin{array}{c} \uparrow \\ f \in \mathcal{C}^1 \end{array}$$

Concept of global stability



Lyapunov Stability

Core stability concept in the NL/LPV context



Incremental Stability

Convergence of trajectories
(equilibrium free stability)



Can fail in case of tracking!

Closed-loop NL guarantees can be lost.

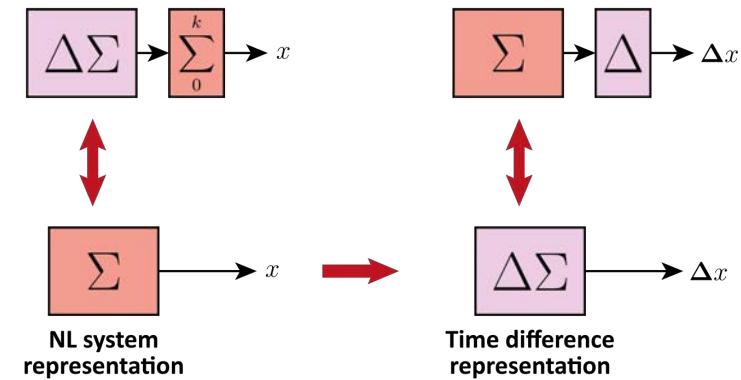
Concept of global stability

- Krasovskii type of condition
 - Consider the **time difference** form

$$\Delta x_{k+1} = f(x_k) - f(x_{k-1}) \quad \Delta x_0 \in \mathbb{R}^{n_x}$$

$$\Delta x_{k+1} = \int_0^1 \frac{\partial f}{\partial x}(\check{x}_k(\lambda)) d\lambda \cdot \Delta x_k \quad \check{x}_k(\lambda) = x_{k-1} + \lambda(x_k - x_{k-1})$$

$$\Delta x_{k+1} = \mathcal{A}(x_k, x_{k-1}) \Delta x_k$$



Concept of global stability

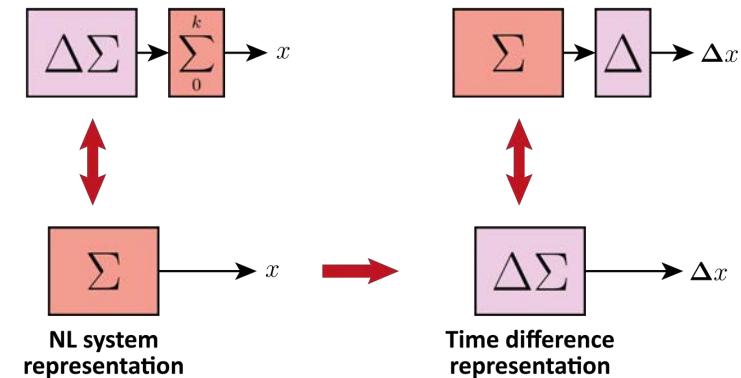
- Krasovskii type of condition
 - Consider the **time difference** form

$$\begin{aligned}\Delta x_{k+1} &= \mathcal{A}(x_k, x_{k-1})\Delta x_k \\ \Delta x_0 &\in \mathbb{R}^{n_x}\end{aligned}$$

Shifted Stability (Asymptotic)

There exists a **KL** function β such that for any $x_0 \in \mathbb{X}$, there is a $x_* \in \mathbb{X}$ s.t.:

$$\|x_k - x_*\|_2 \leq \underbrace{\beta(\|x_0 - x_*\|_2, k)}_{\kappa e^{-ct} \|x_0 - x_*\|_2}$$



Shifted Stability (Sufficiency condition)

If there exists a $\mathcal{X} \succ 0$ such that $\forall \bar{x}, \bar{\bar{x}} \in \mathbb{X}$

$$\mathcal{A}^\top(\bar{x}, \bar{\bar{x}})\mathcal{X}\mathcal{A}(\bar{x}, \bar{\bar{x}}) - \mathcal{X} \prec 0$$

Concept of global stability

- Krasovskii type of condition
 - Consider the **time difference** form

$$\begin{aligned}\Delta x_{k+1} &= \mathcal{A}(x_k, x_{k-1})\Delta x_k \\ \Delta x_0 &\in \mathbb{R}^{n_x}\end{aligned}$$

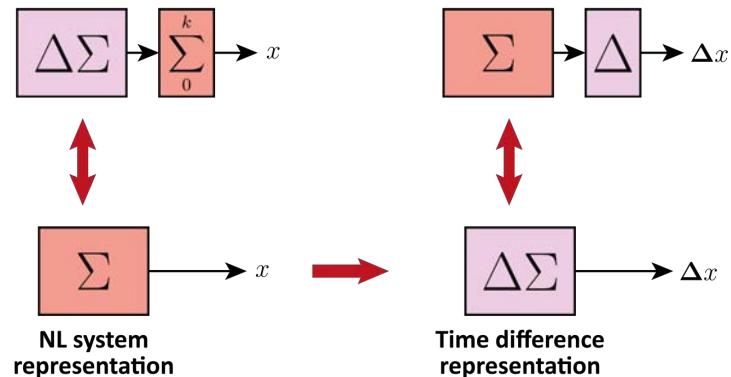
- Quadratic stability

$$V(x) = \underbrace{(f(x) - x)^\top}_{\Delta x} \mathcal{X} \underbrace{(f(x) - x)}_{\Delta x} \quad \mathcal{X} \succ 0$$

$$\Delta x^\top \mathcal{A}^\top(qx, x) \mathcal{X} \mathcal{A}(qx, x) \Delta x - \Delta x^\top \mathcal{X} \Delta x \prec 0$$



$$(f(x) - x)^\top \mathcal{A}^\top(f(x), x) \mathcal{X} \mathcal{A}(f(x), x) (f(x) - x) - (f(x) - x)^\top \mathcal{X} (f(x) - x) \prec 0$$



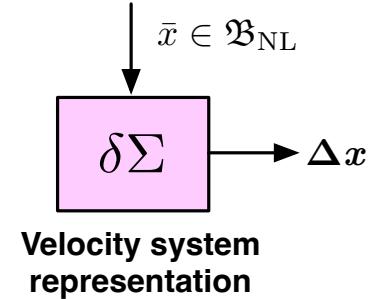
Shifted Stability (Sufficiency condition)

If there exists a $\mathcal{X} \succ 0$ such that $\forall \bar{x}, \bar{\bar{x}} \in \mathbb{X}$

$$\mathcal{A}^\top(\bar{x}, \bar{\bar{x}}) \mathcal{X} \mathcal{A}(\bar{x}, \bar{\bar{x}}) - \mathcal{X} \prec 0$$

Concept of global stability

- **Velocity stability**
 - Enough to consider the stability of the velocity form



$$\Delta x_{k+1} = \mathcal{A}(\bar{x}_k, \bar{x}_{k-1}) \cdot \Delta x_k$$

Velocity stability

$$\mathcal{X} \succ 0$$

$$\mathcal{A}^\top(\bar{x}, \bar{\bar{x}})\mathcal{X}\mathcal{A}(\bar{x}, \bar{\bar{x}}) - \mathcal{X} \prec 0$$

$$\forall \bar{x}, \bar{\bar{x}} \in \mathbb{X}$$



Shifted stability

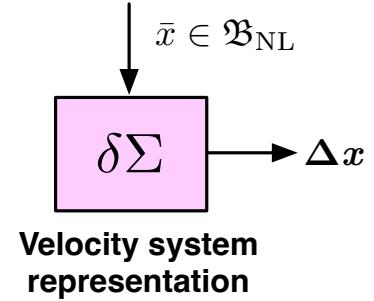
$$V(x) \succ 0$$

$$\Delta V(x) \prec 0$$

$$\forall x \in \mathbb{X}_{x_*} \subseteq \mathbb{X}$$

Concept of global stability

- Velocity stability
 - Enough to consider the stability of the velocity form

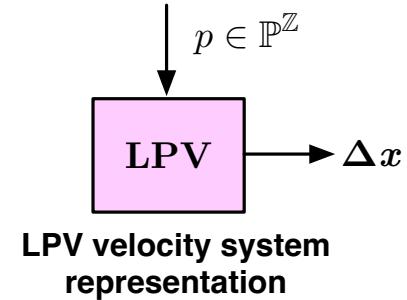


$$\Delta x_{k+1} = \mathcal{A}(\bar{x}_k, \bar{x}_{k-1}) \cdot \Delta x_k$$

Looks like an LPV form!

Concept of global stability

- **Velocity stability**
 - Enough to consider the stability of the velocity form



$$\Delta x_{k+1} = A(\textcolor{red}{p_k}) \cdot \Delta x_k$$

Looks like an LPV form!

Quadratic LPV stability

$$\mathcal{X} \succ 0$$

$$A^\top(\textcolor{red}{p})\mathcal{X}A(\textcolor{red}{p}) - \mathcal{X} \prec 0$$

$$\forall \textcolor{red}{p} \in \mathbb{P}$$



Shifted stability

$$V(x) \succ 0$$

$$\Delta V(x) \prec 0$$

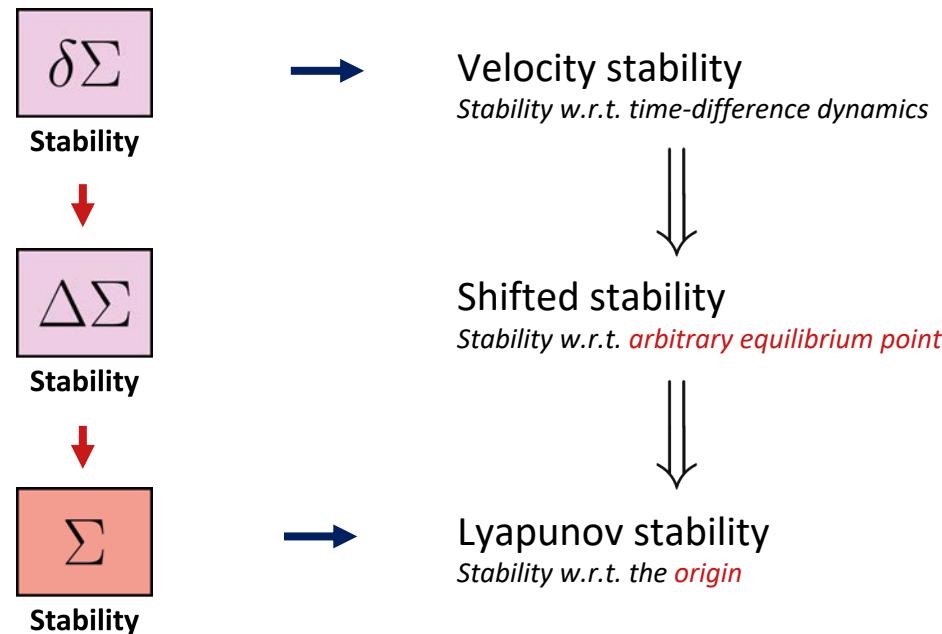
$$\forall x \in \mathbb{X}_{x_*} \subseteq \mathbb{X}$$

LPV embedding

We can guarantee stability via an LPV embedding of the velocity form

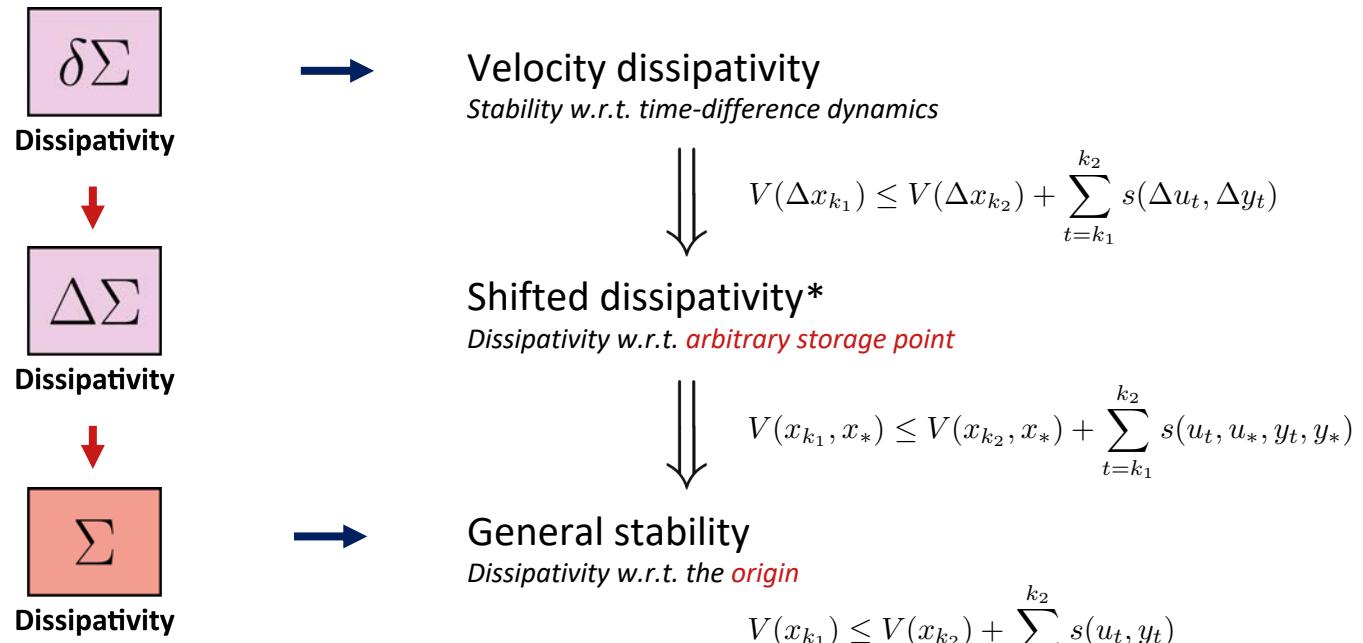
Concept of global stability

Theory in pictures

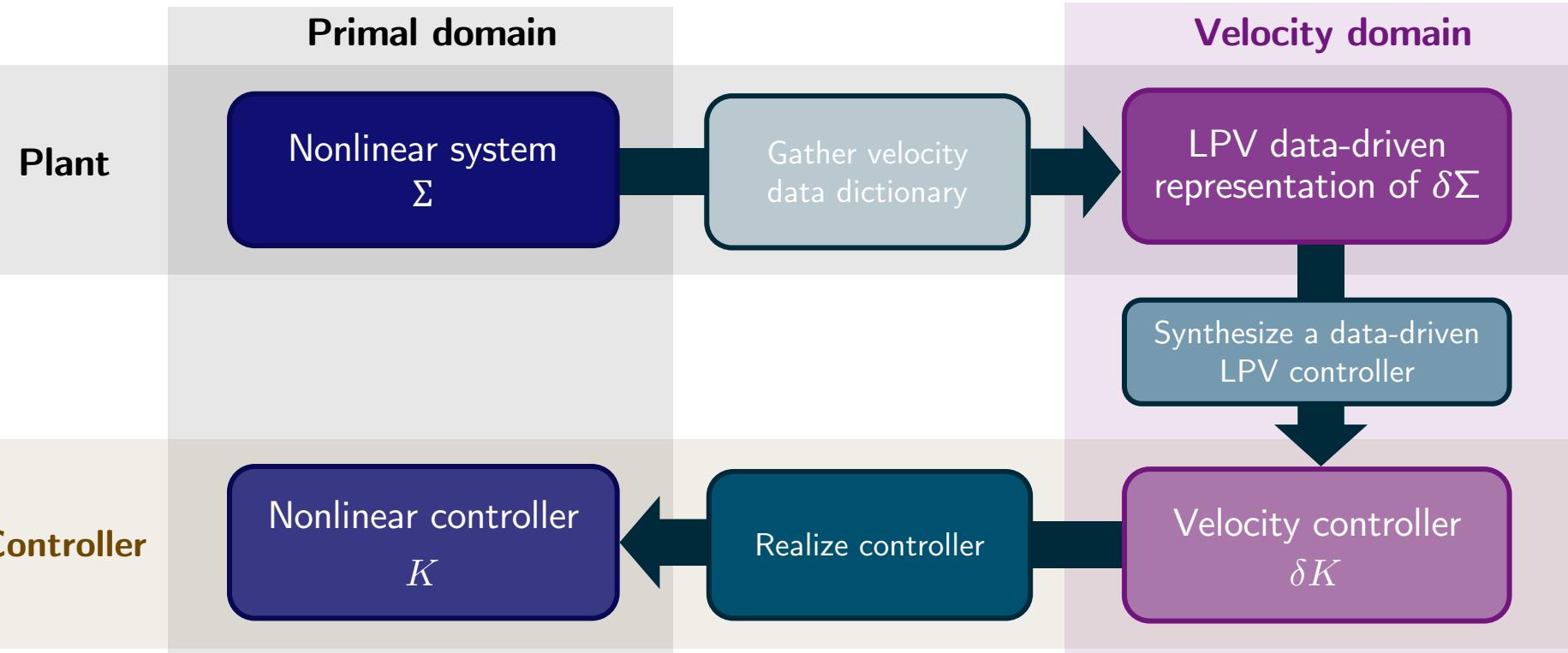


Concept of global performance

Theory in pictures



Data-driven NL controller synthesis



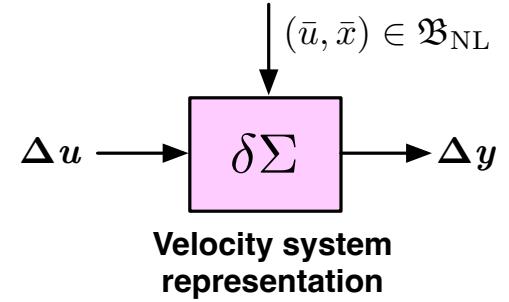
Data-driven NL controller synthesis

- State-feedback synthesis

- Consider the time difference form ($w_k = \text{col}(u_k, x_k)$)

$$\Delta x_{k+1} = \mathcal{A}(w_k, w_{k-1})\Delta x_k + \mathcal{B}(w_k, w_{k-1})\Delta u_k,$$

$$\Delta y_k = \Delta x_k$$



- Assume a given set of basis functions $\psi_1, \dots, \psi_{n_p}$, such that

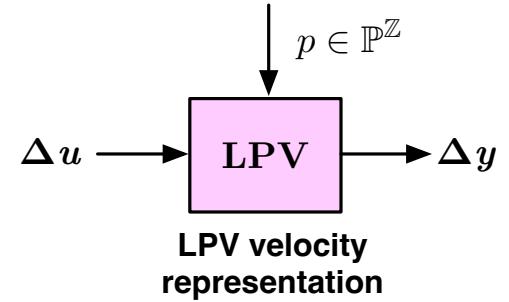
$$\mathcal{A}(w_k, w_{k-1}) = A_0 + \sum_{i=1}^{n_p} A_i \psi_i(w_k, w_{k-1})$$

$$\mathcal{B}(w_k, w_{k-1}) = B_0 + \sum_{i=1}^{n_p} B_i \psi_i(w_k, w_{k-1})$$

Data-driven NL controller synthesis

- State-feedback synthesis
 - LPV embedding

$$\begin{aligned}\Delta x_{k+1} &= A(p_k)\Delta x_k + B(p_k)\Delta u_k, \\ \Delta y_k &= \Delta x_k\end{aligned}$$



- Scheduling is defined as

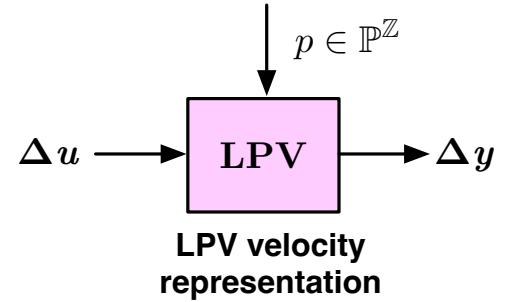
$$p_k := \psi(x_k, u_k, x_{k-1}, u_{k-1})$$

Data-driven NL controller synthesis

- State-feedback synthesis

$$\Delta x_{k+1} = A(p_k) \Delta x_k + B(p_k) \Delta u_k,$$

$$\Delta y_k = \Delta x_k$$



Data-dictionary:

$$\mathcal{D}_{N+1}^{\text{NL}} = \{u_k^{\text{d}}, x_k^{\text{d}}\}_{k=0}^N$$



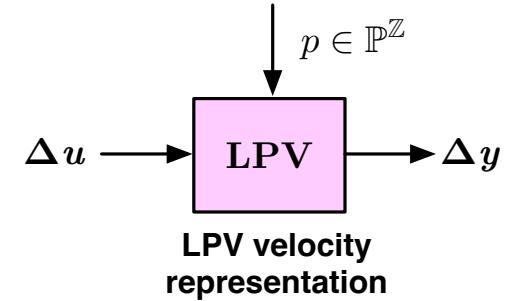
$$\mathcal{D}_N^{\Delta} = \{\Delta u_k^{\text{d}}, \Delta x_k^{\text{d}}, p_k^{\text{d}}\}_{k=1}^N$$

Data-driven NL controller synthesis

- State-feedback synthesis

$$\Delta x_{k+1} = A(p_k)\Delta x_k + B(p_k)\Delta u_k,$$

$$\Delta y_k = \Delta x_k$$



Apply LPV state feedback synthesis

Data-dictionary:

$$U_\Delta = [\Delta u_1^d \quad \cdots \quad \Delta u_{N-1}^d],$$

$$X_\Delta = [\Delta x_1^d \quad \cdots \quad \Delta x_{N-1}^d],$$

$$\vec{X}_\Delta = [\Delta x_2^d \quad \cdots \quad \Delta x_N^d],$$

$$U_\Delta^p = [p_1^d \otimes \Delta u_1^d \quad \cdots \quad p_{N-1}^d \otimes \Delta u_{N-1}^d],$$

$$X_\Delta^p = [p_1^d \otimes \Delta x_1^d \quad \cdots \quad p_{N-1}^d \otimes \Delta x_{N-1}^d].$$



Data-driven representation:

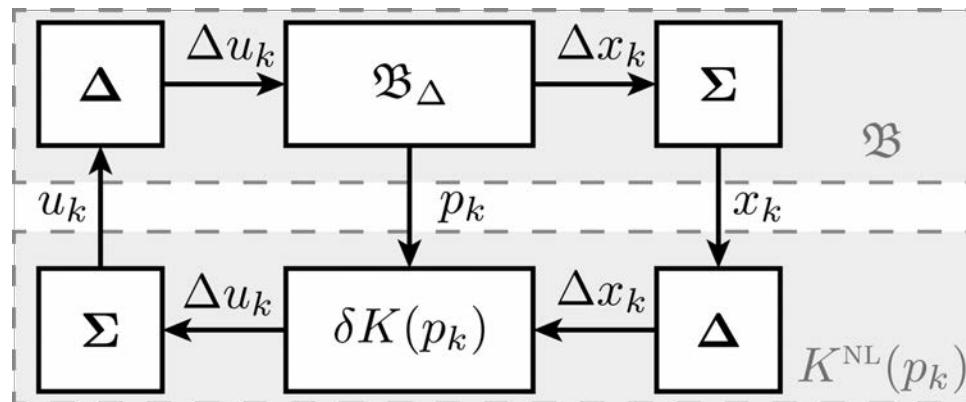
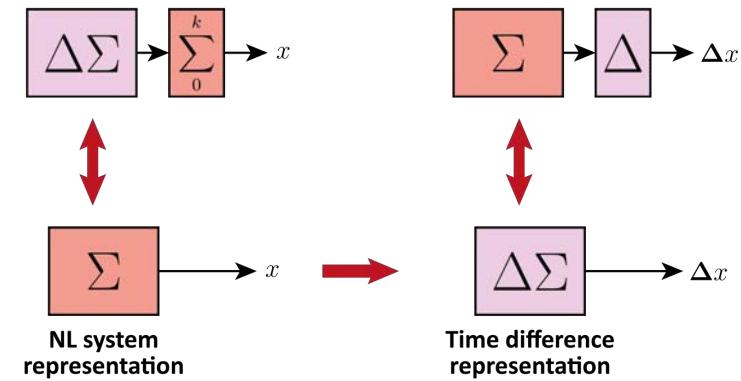
$$\Delta \bar{x}_{k+1} = \vec{X}_\Delta^\top \begin{bmatrix} \Delta \bar{x}_k \\ \bar{p}_k \otimes \Delta \bar{x}_k \\ \Delta \bar{u}_k \\ \bar{p}_k \otimes \Delta \bar{u}_k \end{bmatrix}$$

Controller realization

- Controller realization

- Velocity form (state-feedback case):

$$\begin{aligned}\Delta u_k &= \delta K(p_k) \Delta x_k \\ &= \delta K(\psi(x_k, u_k, x_{k-1}, u_{k-1})) \Delta x_k\end{aligned}$$



Controller realization

- Controller realization
 - Velocity form (state-feedback case):

$$\begin{aligned}\Delta u_k &= \delta K(p_k) \Delta x_k \\ &= \delta K(\psi(x_k, u_k, x_{k-1}, u_{k-1})) \Delta x_k\end{aligned}$$

- Primal form (realization):

$$K^{\text{NL}} \left\{ \begin{array}{l} \chi_{k+1} = \begin{bmatrix} 0 & 0 \\ -\delta K(p_k) & I \end{bmatrix} \chi_k + \begin{bmatrix} I \\ \delta K(p_k) \end{bmatrix} x_k \\ u_k = \begin{bmatrix} -\delta K(p_k) & I \end{bmatrix} \chi_k + \delta K(p_k) x_k \\ p_k = \psi(x_k, u_k, \chi_k) \\ \chi_k = [x_{k-1}^\top \quad u_{k-1}^\top]^\top \end{array} \right.$$

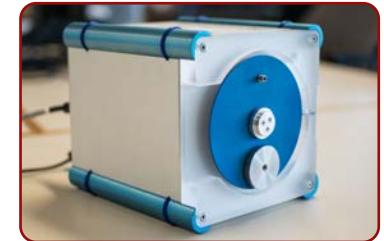
Preservation of guarantees

Realization preserves shifted stability & dissipativity!

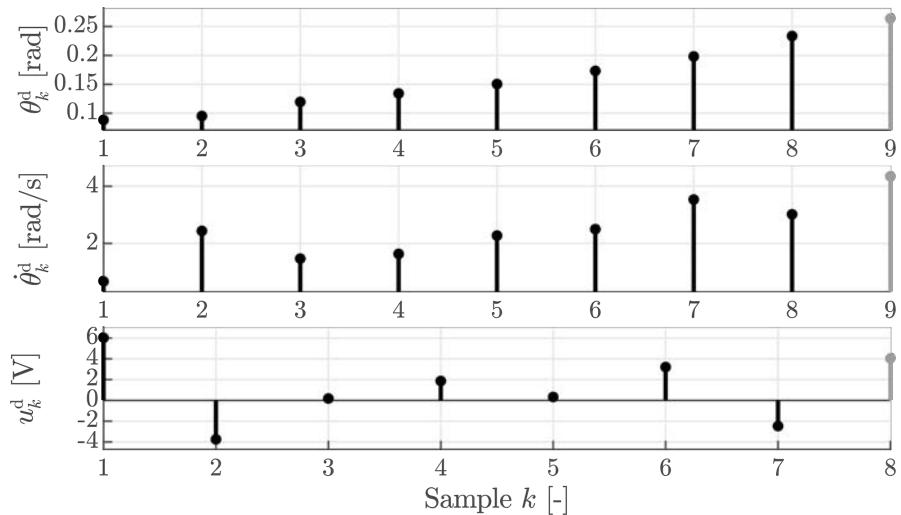
Achievement:

The LPV synthesis is used as a surrogate tool for designing an NL controller with perf. guarantees.

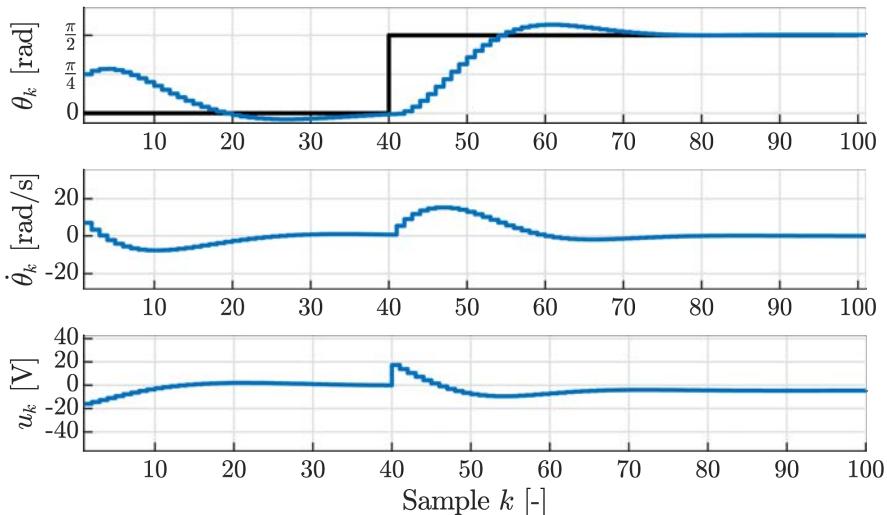
Controller realization



- **Unbalanced disc system (simulation):**
 - Basis functions chosen based on a priori knowledge



Data-dictionary

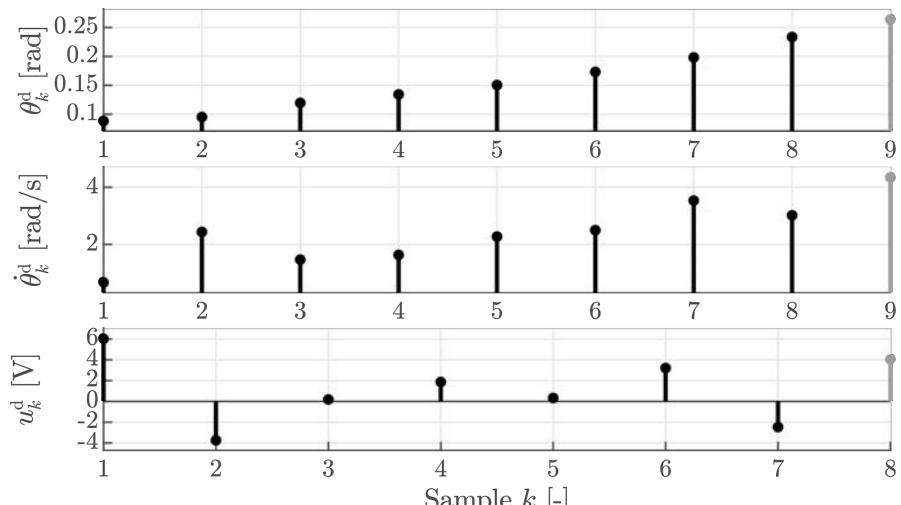


Data-driven **nonlinear**

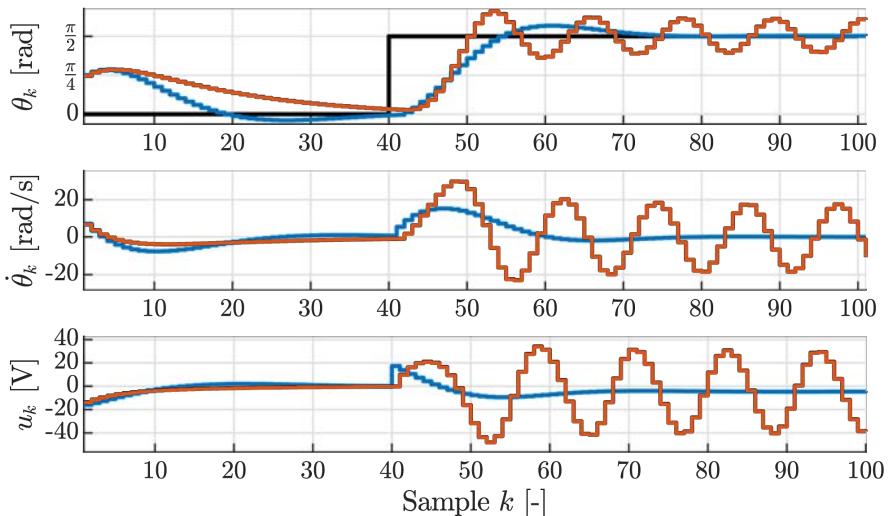
controller

Controller realization

- Unbalanced disc system (**simulation**):
 - Basis functions chosen based on a priori knowledge



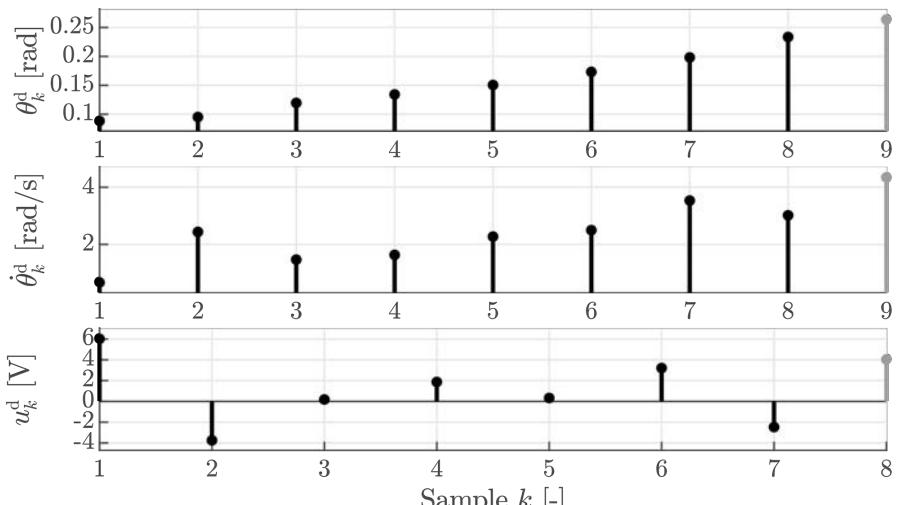
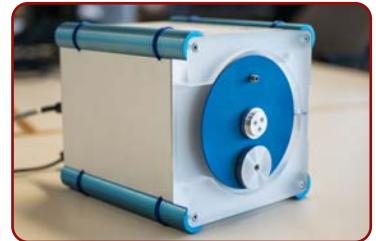
Data-dictionary



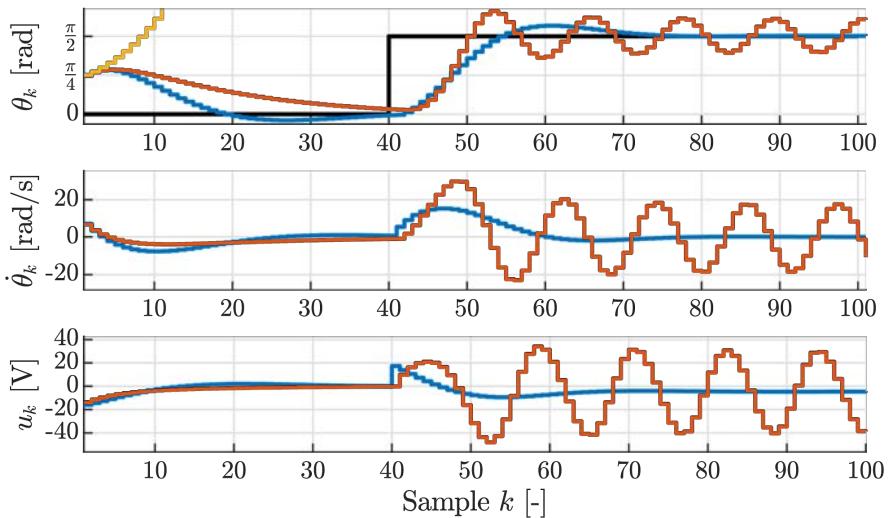
Data-driven **nonlinear**, **LPV** controller

Controller realization

- **Unbalanced disc system (simulation):**
 - Basis functions chosen based on a priori knowledge



Data-dictionary



Data-driven **nonlinear**, **LPV** and **LTI** controller

Table of contents

- Behavioral LTI data-driven control
- Behavioral LPV data-driven control
- Behavioral NL data-driven control
- Conclusions

Conclusions

Effective tools for direct synthesis [NL](#) controllers from time-domain data

- Using tools in behavioral data-driven LPV framework
- Generalization to output-feedback and predictive control case
- General performance objectives (passivity, \mathcal{L}_2 , generalized \mathcal{H}_2 , etc.)

Outlooks

- Data-driven learning of the basis functions
- Scaling up to incremental stability and performance ([reference tracking](#))
- Handling noise and stochastic aspects
- Integration into [LPVcore](#) (off-the-shelf software solution)

