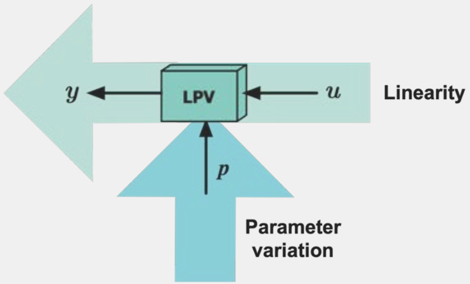
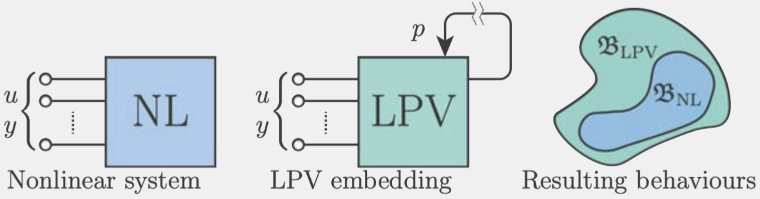


$$\dot{x} = A(p)x + B(p)u$$
$$y = C(p)x + D(p)u$$



LPV Modeling and Control

Tutorial on the Linear Parameter-Varying framework

CSCG Group Meeting, October 17th, 2023

Chris Verhoek

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

Table of contents

- A motivating example
- The linear parameter-varying concept
- LPV representations
- LPV modeling via the unbalanced disc
- LPV controller synthesis
- LPV control of the unbalanced disc
- Summary and final comments

Motivating example

ESA Space Rider

- Reusable space craft
- Multi-million euro vehicle
- Return to Earth autonomously
- Landing-precision requirement: ≤ 1 meter

Heavily nonlinear system, subject to harsh disturbances!

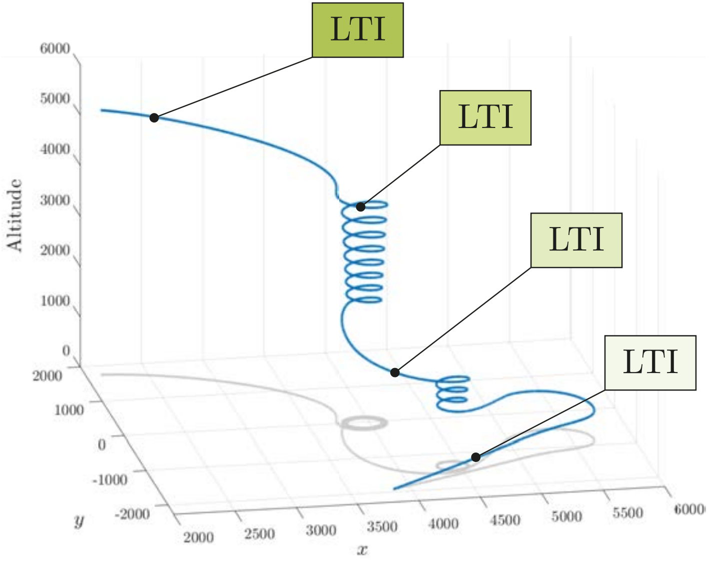
Need for accurate control with:

- Wide operating range
- Guaranteed stability & performance

Motivating example

Simplified aerodynamic model already rather complex...

➢ How to control this system?



Flight controller design:

- Hierarchal control structure (GNC)
- Needs to work for all operating conditions!

Our control options?

- Nonlinear control? → Performance shaping? Guarantees?
- Robust control? → Why sacrifice performance if we know the altitude?

Motivating example

Engineers' dream:

Design controllers for nonlinear systems with *linear control synthesis and shaping* concepts.

- **Idea:** Apply *robust control* by embedding variations as uncertainty.
- **Result:** Controller can only stabilize a narrow operating range

Robust control systematically trades performance for stability and the size of the uncertainty a *single* LTI controller can stabilize is limited...

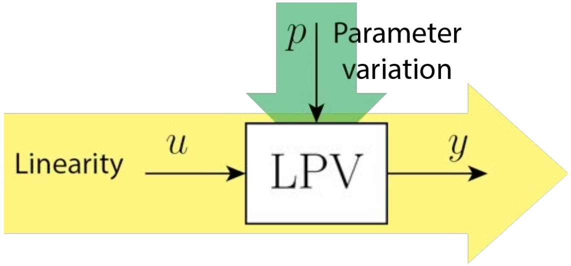
- Overcome limitations requires going **beyond LTI systems**

Table of contents

Linear parameter-varying systems

Core aspects of LPV systems

- Linear dynamic relationship w.r.t. input, output, (state) signals
- Relationship varies along a *measurable* scheduling signal $p(t)$
- Scheduling signal is assumed to vary *independently* in a set $\mathbb{P}$
- LPV behavior is **linear** and **time-invariant** along $p(t)$



- ➢ 30+ years of development
- ➢ Strong theoretical framework (modeling, identification, control)
- ➢ Many successful industrial applications

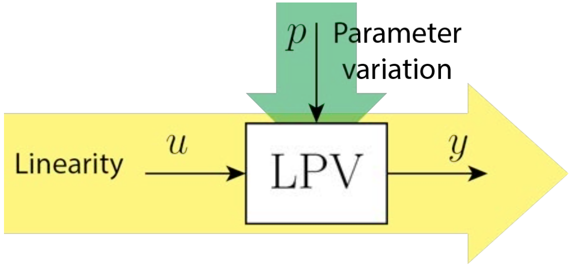
Linear parameter-varying systems

Obtaining LPV models:

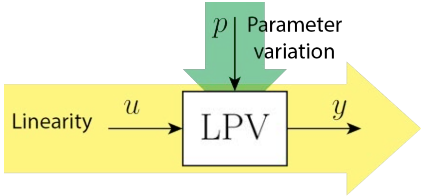
- ‘True’ LPV models
- From nonlinear systems

From nonlinear systems:

- **Local** approaches
- **Global** approaches

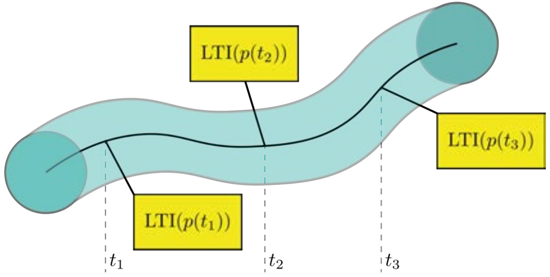


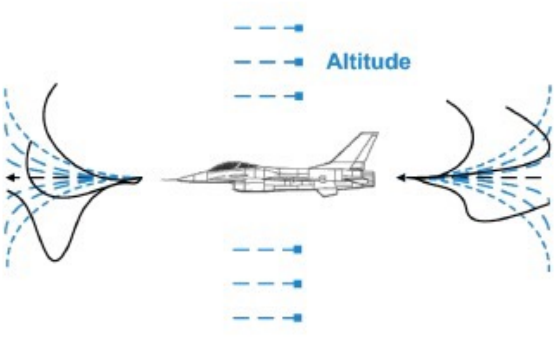
The LPV concept: Principles & formulation

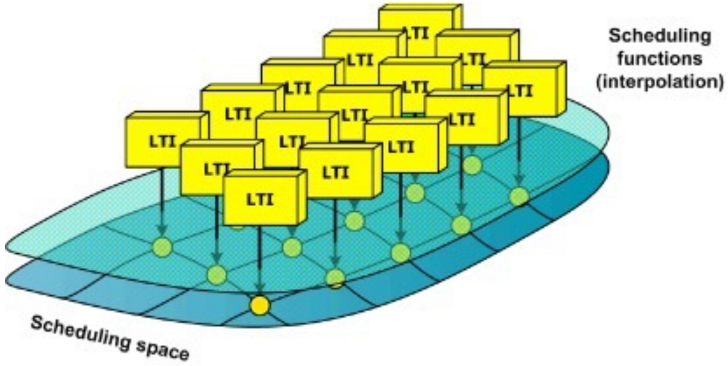


The **local** approach:

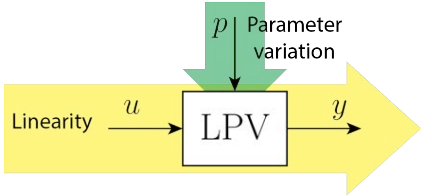
- Schedule **local linearizations** of the system
- Measurable scheduling signal $p(t)$ becomes exogenous!





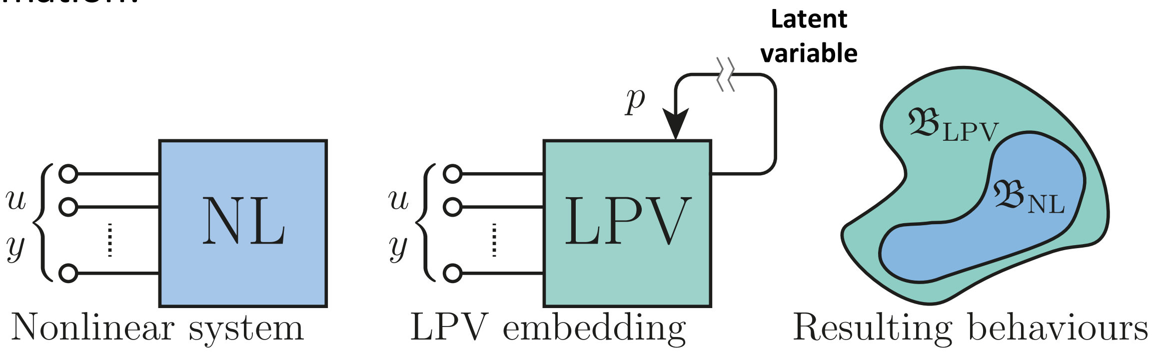


The LPV concept: Principles & formulation



The **global** approach:

- Introduce $p(t)$ as **latent variable** s.t. remaining relations are linear
- We consider $p(t)$ to be exogenous and measurable
- Embedding of NL behavior in LPV behavior
- No approximation!

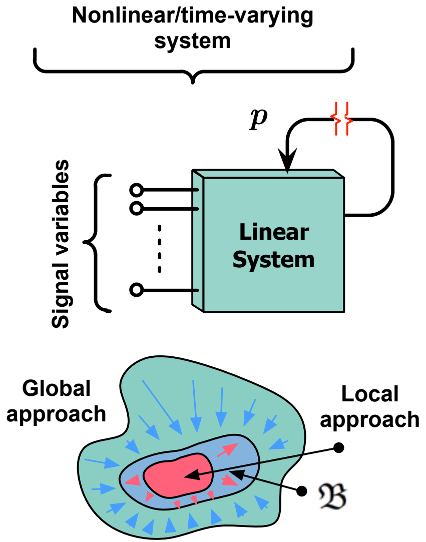


The LPV concept: Principles & formulation

Local and global approaches characterize the spectrum of LPV *embedding principles*.

- Local LPV modeling (inner approx.):
  1. Choose operating conditions
  2. Linearize system at chosen points
  3. Interpolate local models

- Global LPV modeling (outer approx.):
  1. Choose scheduling signal
  2. Transform system



Primary objective: reducing approximation error and/or conservatism

The LPV concept: Applications & outlooks

Many promising applications:

- Aerospace control
- Robotics and high-tech
- Process control
- Magnetic bearings & gyro control
- Automotive systems
- Energy management (batteries, inverter)
- Biomechanics
- Environmental (rain flow, canal models)

Table of contents

- A motivating example
- The linear parameter-varying concept
- LPV representations
- LPV modeling via the unbalanced disc
- LPV controller synthesis
- LPV control of the unbalanced disc
- Summary and final comments

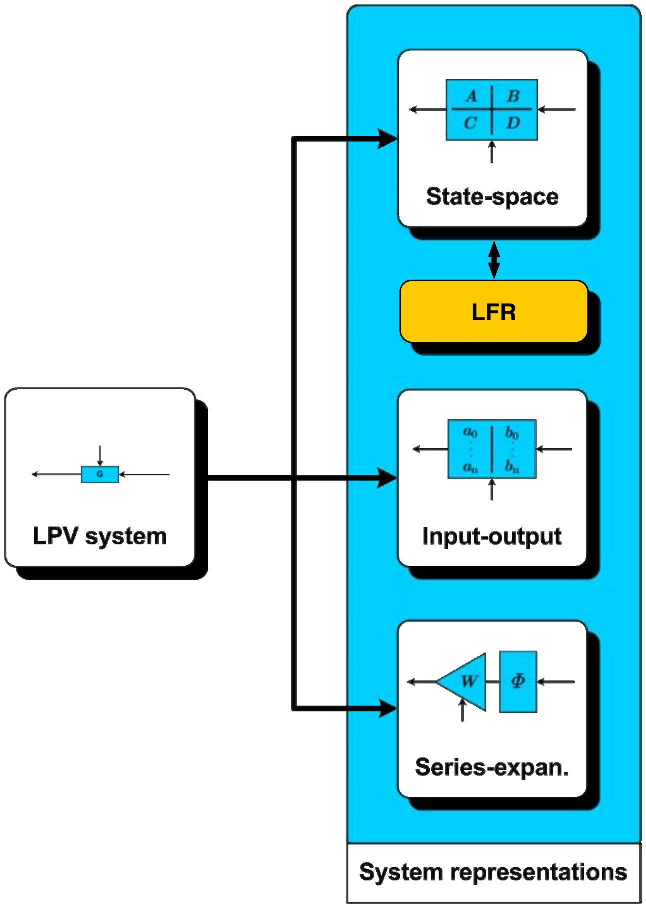
LPV representations

Coefficient functions of representations characterized by:

- Functional dependence
- Static/dynamic dependence

Many different representations:

- State-space (LFR)
- Input-Output
- Kernel
- Infinite impulse response



Tóth: Modeling and Identification of Linear Parameter-Varying Systems, *Springer*, (2010)

LPV representations

Coefficient functions of representations characterized by:

- Functional dependence
- Static/dynamic dependence

State-space representations (**static** dependence)

$$\dot{x}(t) = A(p(t))x(t) + B(p(t))u(t)$$
$$y(t) = C(p(t))x(t) + D(p(t))u(t)$$

with coefficient functions $A : \mathbb{P} \to \mathbb{R}^{n_\mathrm{x} \times n_\mathrm{x}}$, etc.

Coefficient functions:

$$A : \mathbb{P} \to \mathbb{R}^{n_\mathrm{x} \times n_\mathrm{x}}, \ldots$$

Functional dependence:

- Affine/linear
- Polynomial
- Rational
- Meromorphic

Tóth: Modeling and Identification of Linear Parameter-Varying Systems, *Springer*, (2010)

LPV representations

Coefficient functions of representations characterized by:

- Functional dependence
- Static/dynamic dependence

State-space representations (**dynamic** dependence)

$$\dot{x}(t) = (A \diamond p)(t)x(t) + (B \diamond p)(t)u(t)$$
$$y(t) = (C \diamond p)(t)x(t) + (D \diamond p)(t)u(t)$$

Shorthand for evaluation over dynamic dependence

Coefficient functions with finite **dynamic** dependence

$$A(p(t), \tfrac{\mathrm{d}}{\mathrm{d}t}p(t), \tfrac{\mathrm{d}^2}{\mathrm{d}t^2}p(t), \ldots)$$

Same functional dep. options!

Discrete-time equivalent:

$$A(p_k, p_{k-1}, p_{k-2}, \ldots)$$

Tóth: Modeling and Identification of Linear Parameter-Varying Systems, *Springer*, (2010)

LPV representations

Kernel representations (**dynamic** dependence)

$$\underbrace{\sum_{i=0}^{n_\xi}(r_i \diamond p)(t)\frac{\mathrm{d}^i}{\mathrm{d}t^i}}_{R(\frac{\mathrm{d}}{\mathrm{d}t})\diamond p} w(t) = 0$$

Polynomials over $\mathcal{R}$
$R \in \mathcal{R}[\xi]^{n \times n_\mathrm{w}}$
with $\mathcal{R}$ field of meromorphic functions

Behavior is defined as:

$$\mathfrak{B} = \{(w,p) \in (\mathbb{R}^{n_\mathrm{w}} \times \mathbb{P})^{\mathbb{R}} \mid (R(\tfrac{\mathrm{d}}{\mathrm{d}t}) \diamond p)w = 0\}$$

Tóth: Modeling and Identification of Linear Parameter-Varying Systems, *Springer*, (2010)

LPV representations

Similarly for input-output representations:

$$\underbrace{\sum_{i=0}^{n_\mathrm{a}} (a_i \diamond p)(t) \frac{\mathrm{d}^i}{\mathrm{d}t^i}}_{R_\mathrm{y}(\frac{\mathrm{d}}{\mathrm{d}t}) \diamond p} y(t) = \underbrace{\sum_{j=0}^{n_\mathrm{b}} (b_j \diamond p)(t) \frac{\mathrm{d}^j}{\mathrm{d}t^j}}_{R_\mathrm{u}(\frac{\mathrm{d}}{\mathrm{d}t}) \diamond p} u(t)$$

Where:

- $n_\mathrm{a} \geq n_\mathrm{b}$
- u is a free signal
- y doesn't contain any free components

Representations all fit in LPV behavioral framework (complete LPV systems theory)

- Associated notions of minimality, ’uniqueness’, controllability, observabilities, etc.
- Realization theory for equivalence transformations

Tóth: Modeling and Identification of Linear Parameter-Varying Systems, *Springer*, (2010)

LPV modeling

For the sake of the tutorial, focus on static scheduling dependence

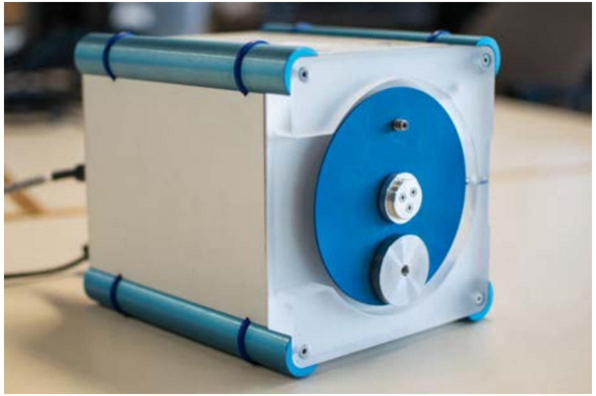
How to obtain such an LPV representation?

- First-principles based
- LPV system identification
  - Local and global methods
  - ARX, ARMAX, OE, Subspace methods, Frequency-domain
- Learning-based
- Direct data-driven (see IfA Coffee Talk)

Demonstrate with Unbalanced Disc

Table of contents

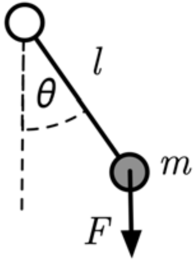
LPV modeling of the unbalanced disc



First-principles based modeling

- Input voltage: u
- Armature current: i
- Angular position: θ
- Angular velocity: ω

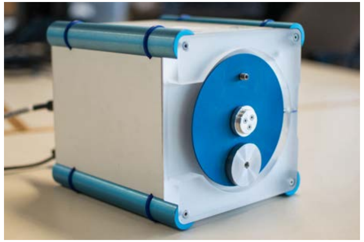
Nonlinear model with lumped electrical dynamics:



$$\begin{pmatrix} \dot{\omega} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} -\frac{1}{\tau} & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \omega \\ \theta \end{pmatrix} + \begin{pmatrix} \frac{\kappa_{\mathrm{m}}}{\tau} \\ 0 \end{pmatrix} u - \begin{pmatrix} \frac{mgl}{J} \sin(\theta) \\ 0 \end{pmatrix}$$

$$y = \theta$$

Local LPV modeling of the unbalanced disc

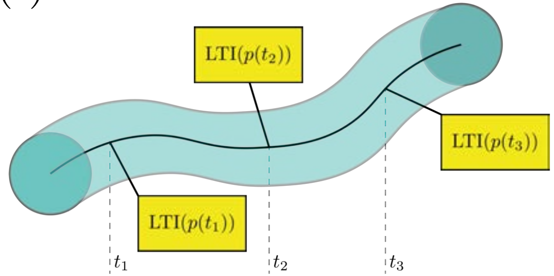


Linearization at $x_* = \begin{pmatrix}\omega_* & \theta_*\end{pmatrix}^\top$ and u_*:

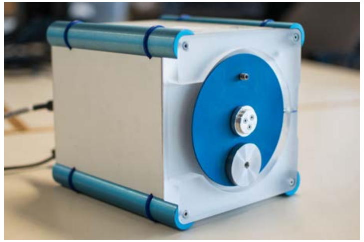
$$\frac{\partial f}{\partial x} = \begin{pmatrix} -\frac{1}{\tau} & -\frac{mgl}{J}\cos(\theta_*) \\ 1 & 0 \end{pmatrix}, \quad \frac{\partial f}{\partial u} = \begin{pmatrix} \frac{\kappa_{\mathrm{m}}}{\tau} \\ 0 \end{pmatrix}, \quad \frac{\partial h}{\partial x} = \begin{pmatrix} 0 & 1 \end{pmatrix}, \quad \frac{\partial h}{\partial u} = 0$$

And **interpolate** the linearized LTI aspects as an LPV model:

- Choose the **scheduling map** ψ, describing local variations with $p(t)$
  - $p = \psi(x, u) := \cos(\theta)$ with clearly $\mathbb{P} = [-1,\ 1]$
- Static affine scheduling dependence



Local LPV modeling of the unbalanced disc

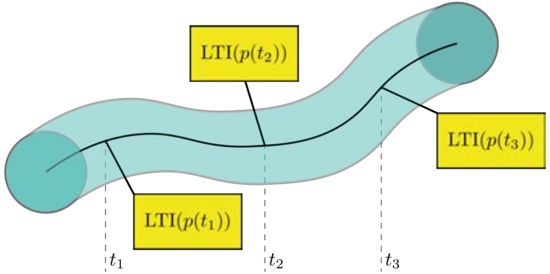


For the *equilibrium* manifold $(\omega_*, \theta_*, u_*) = (0, \theta_*, \frac{mgl\tau}{J\kappa_\mathrm{m}}\sin(\theta_*))$, the LPV model is:

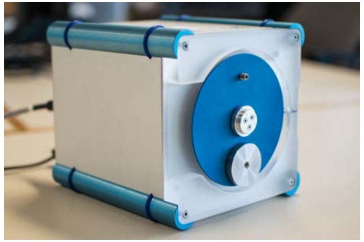
$$\dot{\tilde{x}}(t) = \begin{pmatrix} -\frac{1}{\tau} & -\frac{mgl}{J}p(t) \\ 1 & 0 \end{pmatrix} \tilde{x}(t) + \begin{pmatrix} \frac{\kappa_\mathrm{m}}{\tau} \\ 0 \end{pmatrix} \tilde{u}(t), \quad \tilde{y}(t) = \begin{pmatrix} 0 & 1 \end{pmatrix} \tilde{x}(t)$$

with $\tilde{x} = x - x_*$, $\tilde{u} = u - u_*$, $\tilde{y} = y - y_*$ called trimming.

- If (x_*, u_*) is not an equilibrium point, $\tilde{w} = f(x_*, u_*) \neq 0$ must be added
  - Can be absorbed by trimming or treated as disturbance
- If linearization is accomplished on a set of points, then $A(p), \ldots, D(p)$ can be obtained via interpolation or fitting



Global LPV modeling of the unbalanced disc



Given the nonlinear dynamical equations:

$$\dot{\omega} = -\frac{mgl}{J}\sin(\theta) - \frac{1}{\tau}\omega + \frac{\kappa_{\mathrm{m}}}{\tau}u$$

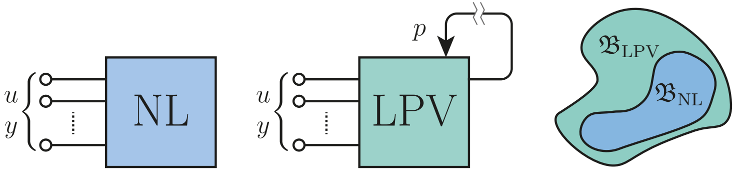
$$\dot{\theta} = \omega$$

Now **factorize** the nonlinearities for linear dependence on θ, ω, u, y

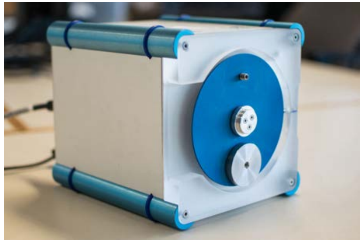
$$\dot{\omega} = -\frac{mgl}{J}\,\mathrm{sinc}(\theta)\,\theta - \frac{1}{\tau}\,\omega + \frac{\kappa_{\mathrm{m}}}{\tau}\,u$$

$$\dot{\theta} = \omega$$

and define $p = \dfrac{\sin(\theta)}{\theta} = \mathrm{sinc}(\theta),\ p(t) \in \mathbb{P} = [-0.22, 1]$



Global LPV modeling of the unbalanced disc



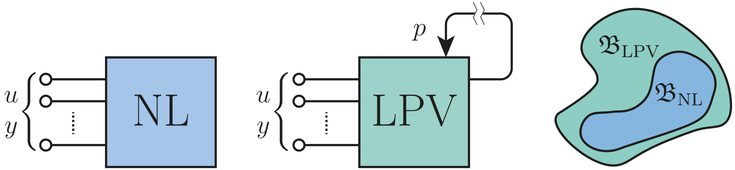
Given the nonlinear dynamical equations:

$$\dot{\omega} = -\frac{mgl}{J}\sin(\theta) - \frac{1}{\tau}\omega + \frac{\kappa_{\mathrm{m}}}{\tau}u$$
$$\dot{\theta} = \omega$$

Now **factorize** the nonlinearities for linear dependence on θ, ω, u, y

$$\dot{x}(t) = \begin{pmatrix} -\frac{1}{\tau} & -\frac{mgl}{J}p(t) \\ 1 & 0 \end{pmatrix} x(t) + \begin{pmatrix} \frac{\kappa_{\mathrm{m}}}{\tau} \\ 0 \end{pmatrix} u(t), \quad y(t) = \begin{pmatrix} 0 & 1 \end{pmatrix} x(t)$$

and define $p = \dfrac{\sin(\theta)}{\theta} = \mathrm{sinc}(\theta),\ p(t) \in \mathbb{P} = [-0.22, 1]$

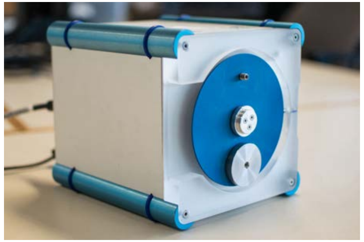


Direct conversion! No approximation & trimming!

Note: factorization generally not unique, but always possible under mild conditions

Koelewijn & Tóth: Automatic Grid-based LPV Embedding of Nonlinear Systems. *Tech. Rep. TU/e* (2021).

LPV modeling of the unbalanced disc



How to do this in MATLAB? **LPVcore**

- Open-source MATLAB toolbox for modeling, identification & control

https://lpvcore.net

```
% define scheduling
p = preal('sinc(x1)','ct','Range',[-0.22, 1]);
% for local case: p = preal('cos(x1)','ct','Range',[-1, 1]);

% coefficient matrices of LPV-SS rep.
A = [-1/tau, -(m*g*l/J)*p; 1, 0];
B = [0; Km/tau];
C = [0, 1];
D = 0;

% create LPV model
UnbalancedDisk = LPVcore.lpvss(A,B,C,D);
```

Usage analogous to MATLABs Robust Control and System Identification toolbox

Table of contents

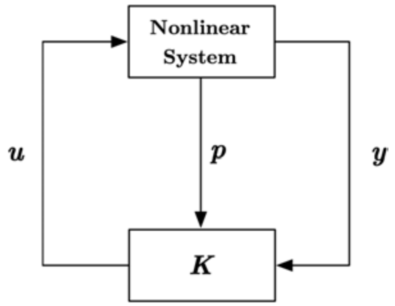
LPV control

Now we can model LPV systems... use them for analysis & control!

➢ This talk: Focus on control

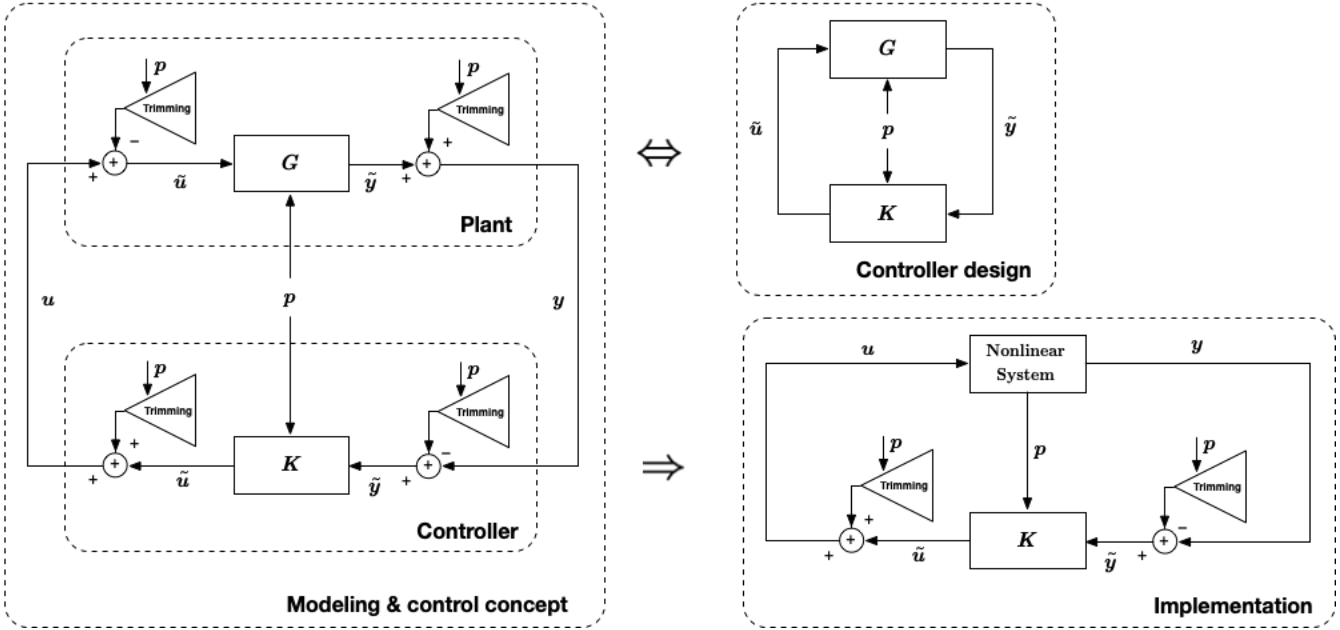
In a nutshell:

- Inspired by robust control (one controller stabilizing all of $\mathbb{P}$)
  - Sacrifices performance for robustness
- Make **LPV controller** dependent on $p(t)$
- $K(p)$ designed for LPV system and implemented for NL system
  - With $p(t)$ measured from the plant or exogenous signals



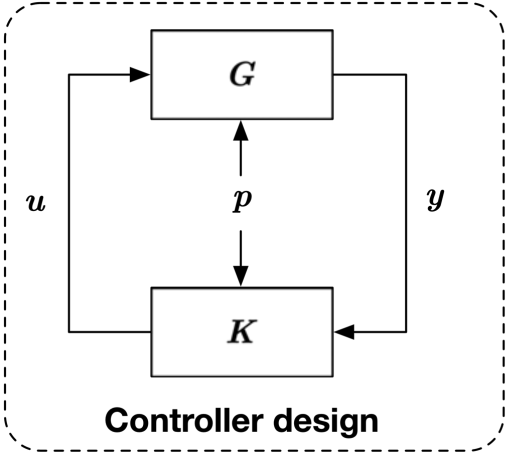
LPV control

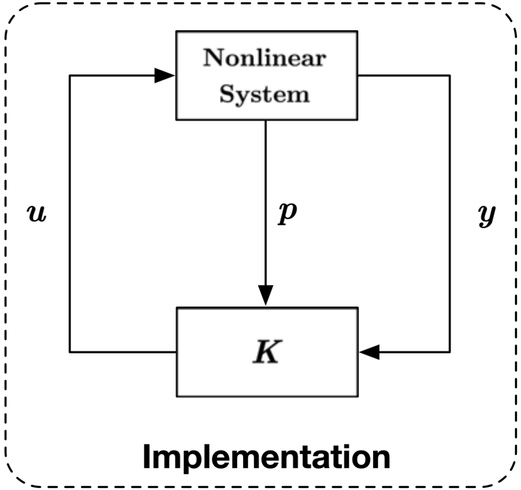
Local and global methods:



LPV control

Local and **global** methods:





LPV control

Many available methods available:

- State-feedback synthesis
- Output feedback synthesis
- Model predictive control

➢ Different strategies for different functional dependencies

➢ All fit in a systematic framework (LFRs)

Coefficient functions:

$$A : \mathbb{P} \to \mathbb{R}^{n_\mathrm{x} \times n_\mathrm{x}}, \ldots$$

Functional dependence:

- Affine/linear
- Polynomial
- Rational
- Meromorphic

LPV control

Many available methods available:

- State-feedback synthesis
- **Output feedback synthesis**
- Model predictive control

➢ Different strategies for different functional dependencies

➢ All fit in a systematic framework (LFRs)

Polytopic approach for global LPV controller synthesis

Coefficient functions:

$$A : \mathbb{P} \to \mathbb{R}^{n_\mathrm{x} \times n_\mathrm{x}}, \ldots$$

Functional dependence:

- **Affine/linear**
- Polynomial
- Rational
- Meromorphic

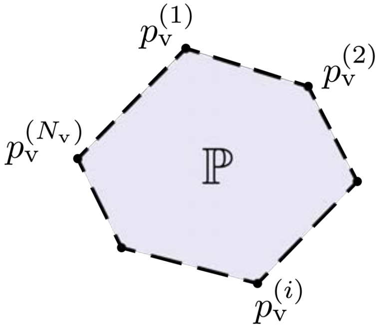
LPV control

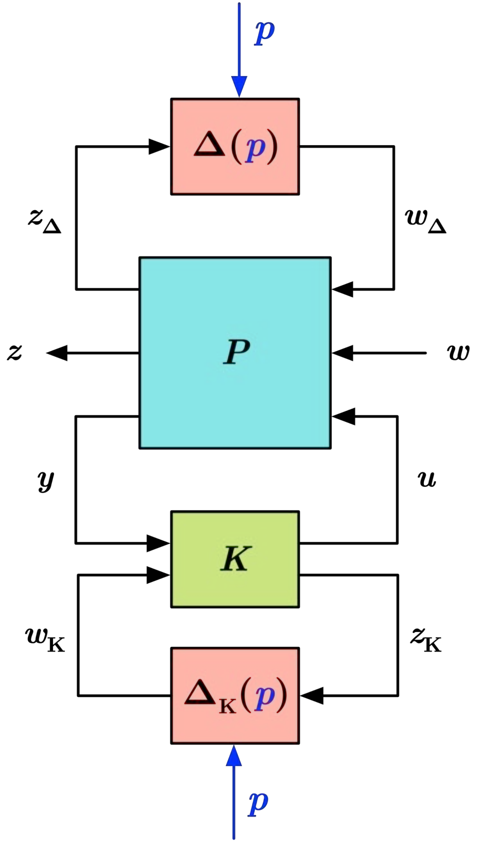
Designing a scheduling-dependent controller guaranteeing:

- Quadratic internal stability (Lyapunov-based)
- $\mathcal{L}_2$-gain based performance (extending $\mathcal{H}_\infty$-control)

For polytopic synthesis, assume:

$$p(t) \in \mathbb{P} = \text{cohull}\{p_\text{v}^{(1)}, \ldots, p_\text{v}^{(N_\text{v})}\}$$

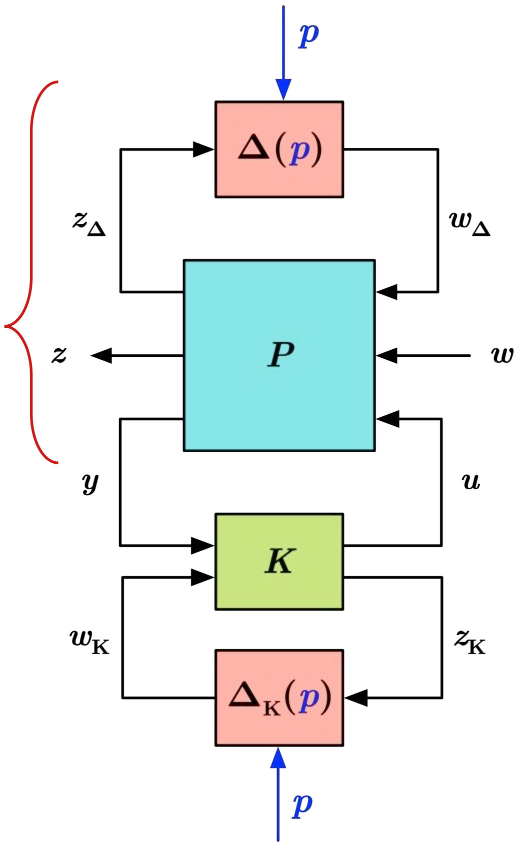




Configuration for LPV synthesis

Open-loop system:

$$\begin{pmatrix} \dot{x}(t) \\ \hline z(t) \\ y(t) \end{pmatrix} = \left(\begin{array}{c|cc} A(p(t)) & B_{\mathrm{w}}(p(t)) & B_{\mathrm{u}}(p(t)) \\ \hline C_{\mathrm{z}}(p(t)) & D_{\mathrm{zw}}(p(t)) & D_{\mathrm{zu}}(p(t)) \\ C_{\mathrm{y}}(p(t)) & D_{\mathrm{yw}}(p(t)) & 0 \end{array}\right) \begin{pmatrix} x(t) \\ \hline w(t) \\ u(t) \end{pmatrix}$$



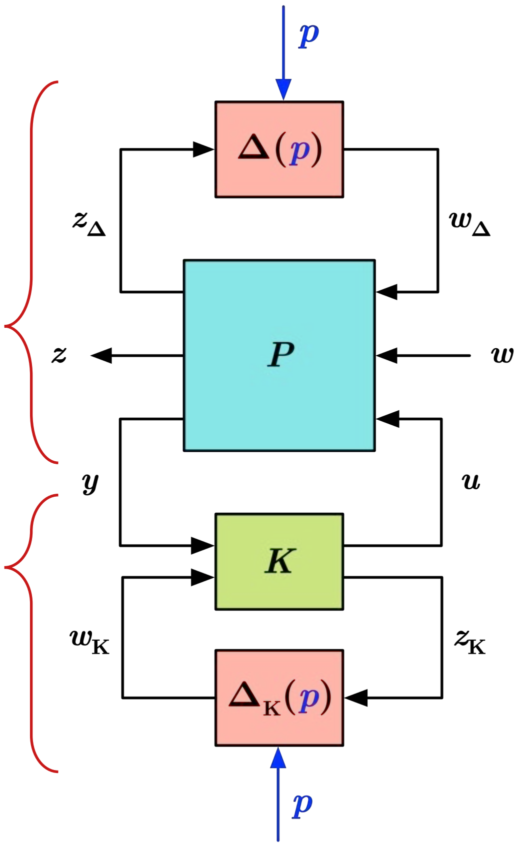
Configuration for LPV synthesis

Open-loop system:

$$\begin{pmatrix} \dot{x}(t) \\ \hline z(t) \\ y(t) \end{pmatrix} = \left(\begin{array}{c|cc} A(p(t)) & B_{\mathrm{w}}(p(t)) & B_{\mathrm{u}}(p(t)) \\ \hline C_{\mathrm{z}}(p(t)) & D_{\mathrm{zw}}(p(t)) & D_{\mathrm{zu}}(p(t)) \\ C_{\mathrm{y}}(p(t)) & D_{\mathrm{yw}}(p(t)) & 0 \end{array}\right) \begin{pmatrix} x(t) \\ \hline w(t) \\ u(t) \end{pmatrix}$$

Controller:

$$\begin{pmatrix} \dot{x}_{\mathrm{K}}(t) \\ \hline u(t) \end{pmatrix} = \left(\begin{array}{c|c} A_{\mathrm{K}}(p(t)) & B_{\mathrm{K}}(p(t)) \\ \hline C_{\mathrm{K}}(p(t)) & D_{\mathrm{K}}(p(t)) \end{array}\right) \begin{pmatrix} x_{\mathrm{K}}(t) \\ \hline y(t) \end{pmatrix}$$



Configuration for LPV synthesis

Open-loop system:

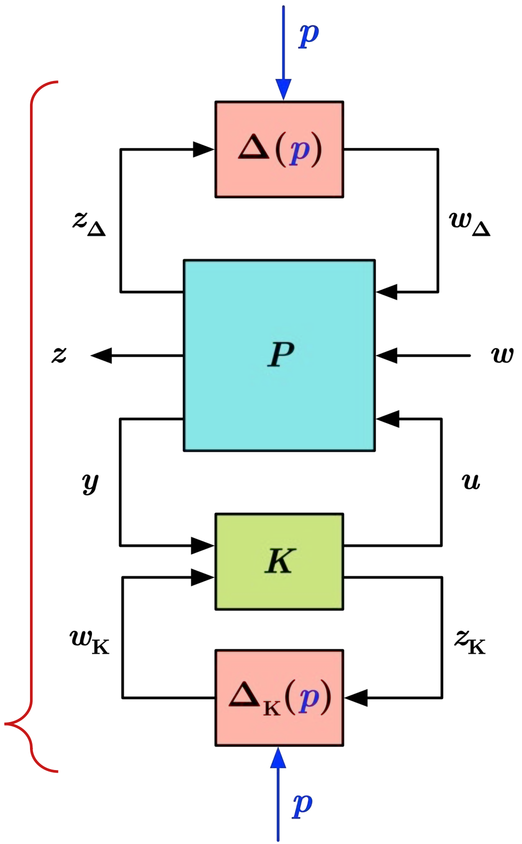
$$\begin{pmatrix} \dot{x}(t) \\ \hline z(t) \\ y(t) \end{pmatrix} = \left(\begin{array}{c|cc} A(p(t)) & B_{\mathrm{w}}(p(t)) & B_{\mathrm{u}}(p(t)) \\ \hline C_{\mathrm{z}}(p(t)) & D_{\mathrm{zw}}(p(t)) & D_{\mathrm{zu}}(p(t)) \\ C_{\mathrm{y}}(p(t)) & D_{\mathrm{yw}}(p(t)) & 0 \end{array}\right) \begin{pmatrix} x(t) \\ \hline w(t) \\ u(t) \end{pmatrix}$$

Controller:

$$\begin{pmatrix} \dot{x}_{\mathrm{K}}(t) \\ \hline u(t) \end{pmatrix} = \left(\begin{array}{c|c} A_{\mathrm{K}}(p(t)) & B_{\mathrm{K}}(p(t)) \\ \hline C_{\mathrm{K}}(p(t)) & D_{\mathrm{K}}(p(t)) \end{array}\right) \begin{pmatrix} x_{\mathrm{K}}(t) \\ \hline y(t) \end{pmatrix}$$

Closed-loop system:

$$\begin{pmatrix} \dot{\xi}(t) \\ \hline z(t) \end{pmatrix} = \left(\begin{array}{c|c} \mathcal{A}(p(t)) & \mathcal{B}(p(t)) \\ \hline \mathcal{C}(p(t)) & \mathcal{C}(p(t)) \end{array}\right) \begin{pmatrix} \xi(t) \\ \hline w(t) \end{pmatrix}$$



Polytopic synthesis concept

Remember the **Bounded Real Lemma**?

If there exists a $\mathcal{X} \succ 0$ such that

$$(*)^\top \left(\begin{array}{cc|cc} 0 & \mathcal{X} & 0 & 0 \\ \mathcal{X} & 0 & 0 & 0 \\ \hline 0 & 0 & Q_\mathrm{p} & S_\mathrm{p} \\ 0 & 0 & S_\mathrm{p}^\top & R_\mathrm{p} \end{array}\right) \left(\begin{array}{cc} I & 0 \\ \mathcal{A}(p) & \mathcal{B}(p) \\ \hline 0 & I \\ \mathcal{C}(p) & \mathcal{D}(p) \end{array}\right) \prec 0 \qquad \text{for all } p \in \mathbb{P}$$

then quadratic performance is achieved for the controlled system!

Infinite set of LMIs... How to make this computable?

Polytopic synthesis concept

With affine scheduling dependence of

$$\left(\begin{array}{c:c} \mathcal{A}(p(t)) & \mathcal{B}(p(t)) \\ \hdashline \mathcal{C}(p(t)) & \mathcal{C}(p(t)) \end{array}\right)$$

infinite set of LMIs of prev. slide reduces to set of LMIs in vertices $p_\mathrm{v}^{(1)}, \ldots, p_\mathrm{v}^{(N_\mathrm{v})}$...

How to guarantee this?

1. $$\left(\begin{array}{c:c:c} A(p) & B_\mathrm{w}(p) & B_\mathrm{u} \\ \hdashline C_\mathrm{z}(p) & D_\mathrm{zw}(p) & D_\mathrm{zu} \\ C_\mathrm{y} & D_\mathrm{yw} & 0 \end{array}\right)$$ is affine in p

2. $$\begin{pmatrix} A_\mathrm{K}(p) & B_\mathrm{K}(p) \\ C_\mathrm{K}(p) & D_\mathrm{K}(p) \end{pmatrix}$$ is affine in p

Then closed-loop is affine in $p(t)$

Polytopic synthesis

By means of a well-known parameter-transformation and elimination, we arrive at:

We achieve quadratic performance for the controlled system if there exists a $\mathcal{X} \succ 0$ such that for all $k = 1, \ldots, N_\mathrm{v}$

$$(*)^\top \left(\begin{array}{cc|cc} 0 & \mathcal{X} & 0 & 0 \\ \mathcal{X} & 0 & 0 & 0 \\ \hline 0 & 0 & Q_\mathrm{p} & S_\mathrm{p} \\ 0 & 0 & S_\mathrm{p}^\top & R_\mathrm{p} \end{array}\right) \left(\begin{array}{cc} I & 0 \\ \mathcal{A}(p^{(k)}) & \mathcal{B}(p^{(k)}) \\ \hline 0 & I \\ \mathcal{C}(p^{(k)}) & \mathcal{D}(p^{(k)}) \end{array}\right) \prec 0.$$

- Concept behind this: Convex-hull relaxation
- For $\mathcal{L}_2$-gain based performance, i.e., $\|G\|_{\mathcal{L}_2} < \gamma$, choose $(Q_\mathrm{p}, S_\mathrm{p}, R_\mathrm{p}) = (-\gamma I, 0, I)$

Scherer & Weiland: Linear Matrix Inequalities in Control (Chapter 9), *Lecture notes,* (2022)

Polytopic synthesis – LPV Controller Construction

Solving the synthesis problem gives:

$$\mathcal{X}, \qquad \begin{pmatrix} A_{\mathrm{K},k} & B_{\mathrm{K},k} \\ C_{\mathrm{K},k} & D_{\mathrm{K},k} \end{pmatrix}, \quad k = 1, \ldots, N_{\mathrm{v}}$$

For implementation, represent $p(t) \in \mathbb{P}$ as

$$p(t) = \sum_{k=1}^{N_{\mathrm{v}}} \lambda_k(t)\, p_{\mathrm{v}}^{(k)} \quad \text{with} \quad \lambda_k(t) \geq 0, \; \sum_{k=1}^{N_{\mathrm{v}}} \lambda_k(t) = 1$$

Then, the analysis inequalities are satisfied with

$$\begin{pmatrix} A_{\mathrm{K}}(p(t)) & B_{\mathrm{K}}(p(t)) \\ C_{\mathrm{K}}(p(t)) & D_{\mathrm{K}}(p(t)) \end{pmatrix} = \sum_{k=1}^{N_{\mathrm{v}}} \lambda_k(t) \begin{pmatrix} A_{\mathrm{K},k} & B_{\mathrm{K},k} \\ C_{\mathrm{K},k} & D_{\mathrm{K},k} \end{pmatrix}$$

LPV Controller Construction – Comments

- For simulation and implementation, proceed as follows:
  At time t, find convex combination coefficients in
  $$p(t) = \sum_{k=1}^{N_\mathrm{v}} \lambda_k(t)\, p_\mathrm{v}^{(k)} \quad \text{and use} \quad \sum_{k=1}^{N_\mathrm{v}} \lambda_k(t) \begin{pmatrix} A_{\mathrm{K},k} & B_{\mathrm{K},k} \\ C_{\mathrm{K},k} & D_{\mathrm{K},k} \end{pmatrix}$$
  to define the dynamics of the LPV controller.
  - This requires the solution of an LP → Uniqueness: e.g., $\min \|\lambda\|_2^2$
- If original system affine, transform back to affine possible
- Generalizations exist for parameter-dependent storage $\mathcal{X}$

How “**easy**” is this?

Table of contents

LPV control in MATLAB

1. We have our plant:

   `system = LPVcore.lpvss(A,B,C,D);`

2. Make generalized plant via

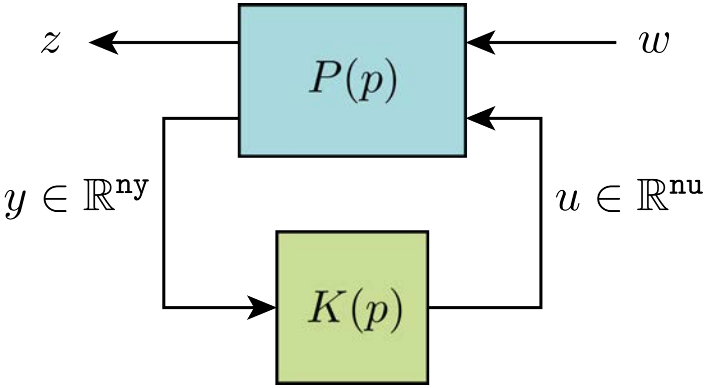
   P = `sysic` or P = `connect(...)`

3. Simply call the `lpvsyn` command with

   `[K, gam, Xcl] = lpvsyn(P, ny, nu);`

   Synthesizes an $\mathcal{L}_2$-gain optimal LPV controller

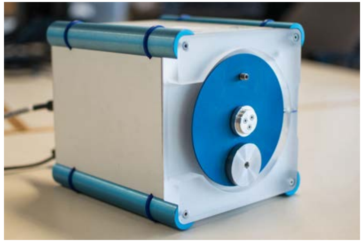
4. Simulate with our new controller!



Comments on LPVcore

- Analysis & synthesis tools available for:
  - $\mathcal{L}_2$-gain
  - Generalized $\mathcal{H}_2$-norm
  - Passivity
  - $\mathcal{L}_\infty$-gain
- Build with the ROLMIP and YALMIP open-source toolboxes (flexibility with solvers)
- Many available options: Control over scheduling dependence controller, pole constraints, numerical conditioning hyperparameters, etc.
- For continuous-time and discrete-time analysis & synthesis
- Simulink blocks available

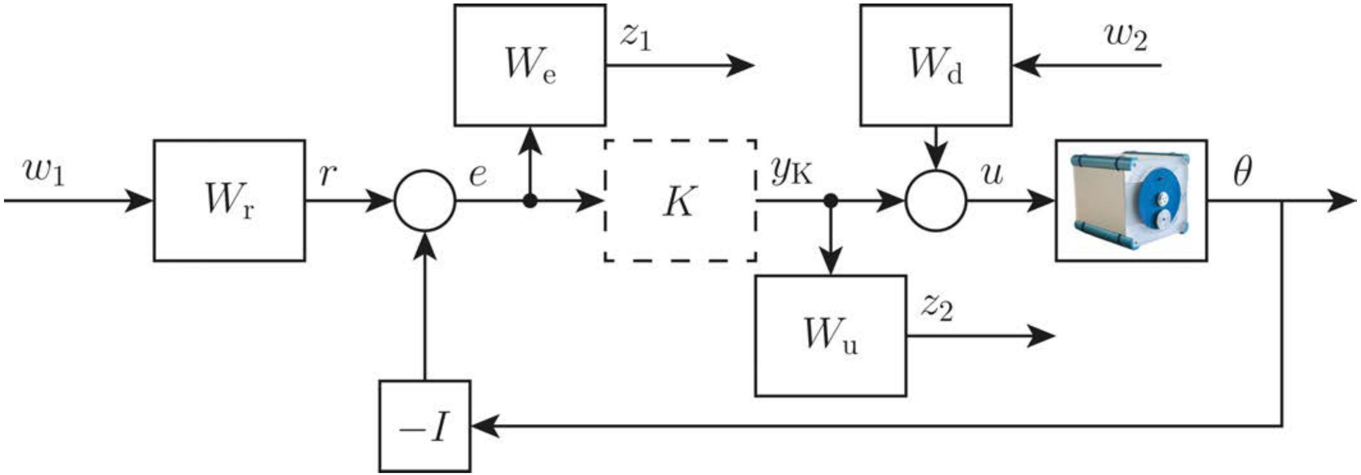
LPV control of the unbalanced disc



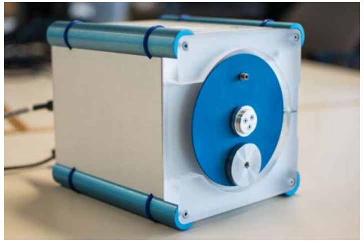
https://lpvcore.net

Really that easy? Yes, (with **LPVcore**)

Controller design:



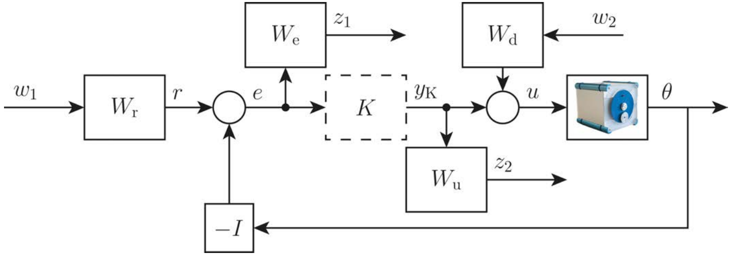
LPV control of the unbalanced disc



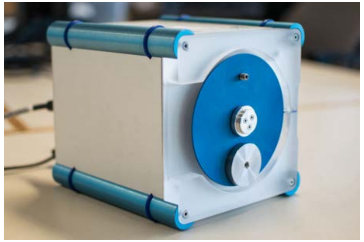
Really that easy? Yes, (with **LPVcore**)

https://lpvcore.net

```
% Interconnection structure
P = connect(UnbalancedDisk, sumblk('e = r - theta'), ...
                            sumblk('u = d + yk'), ...
                            {'r','d','yk'},{'e','yk','e'});
% Make generalized (weighted) plant
Pw = blkdiag(Wz,eye(ny)) * P * blkdiag(Ww,eye(nu));

% Synthesize!
[K, gamma, X] = lpvsyn(Pw, ny, nu);

% gamma = 1.41
```



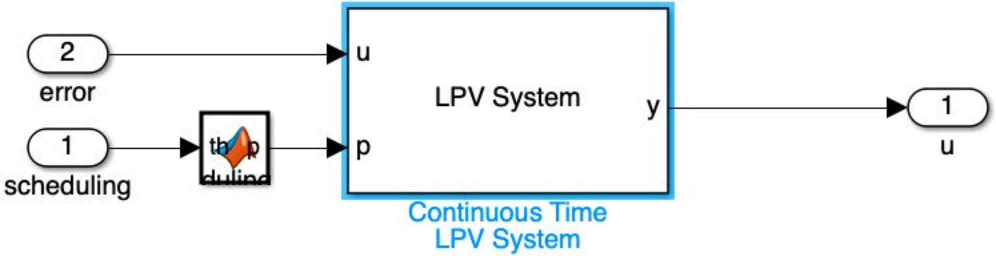
LPV control of the unbalanced disc



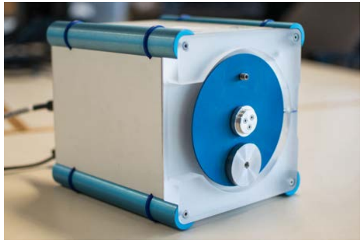
Really that easy? Yes, (with **LPVcore**)

https://lpvcore.net

```
% Interconnection structure
P = connect(UnbalancedDisk, sumblk('e = r - theta'), ...
                            sumblk('u = d + yk'), ...
                            {'r','d','yk'},{'e','yk','e'});
% Make generalized (weighted) plant
Pw = blkdiag(Wz,eye(ny)) * P * blkdiag(Ww,eye(nu));

% Synthesize!
[K, gamma, X] = lpvsyn(Pw, ny, nu);

% gamma = 1.41
```



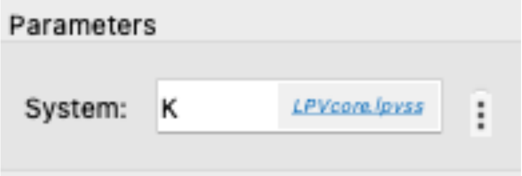
LPV control of the unbalanced disc

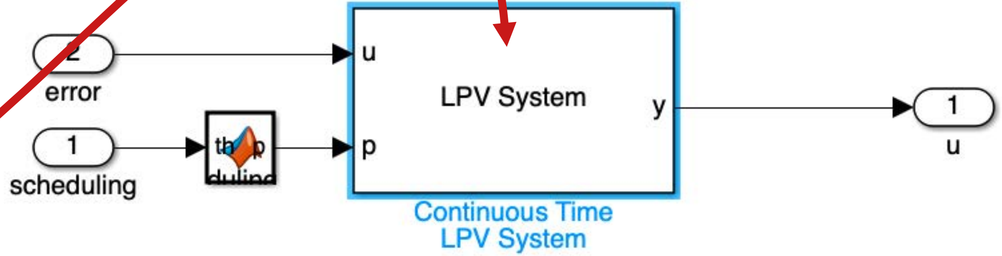


Really that easy? Yes, (with **LPVcore**)

https://lpvcore.net

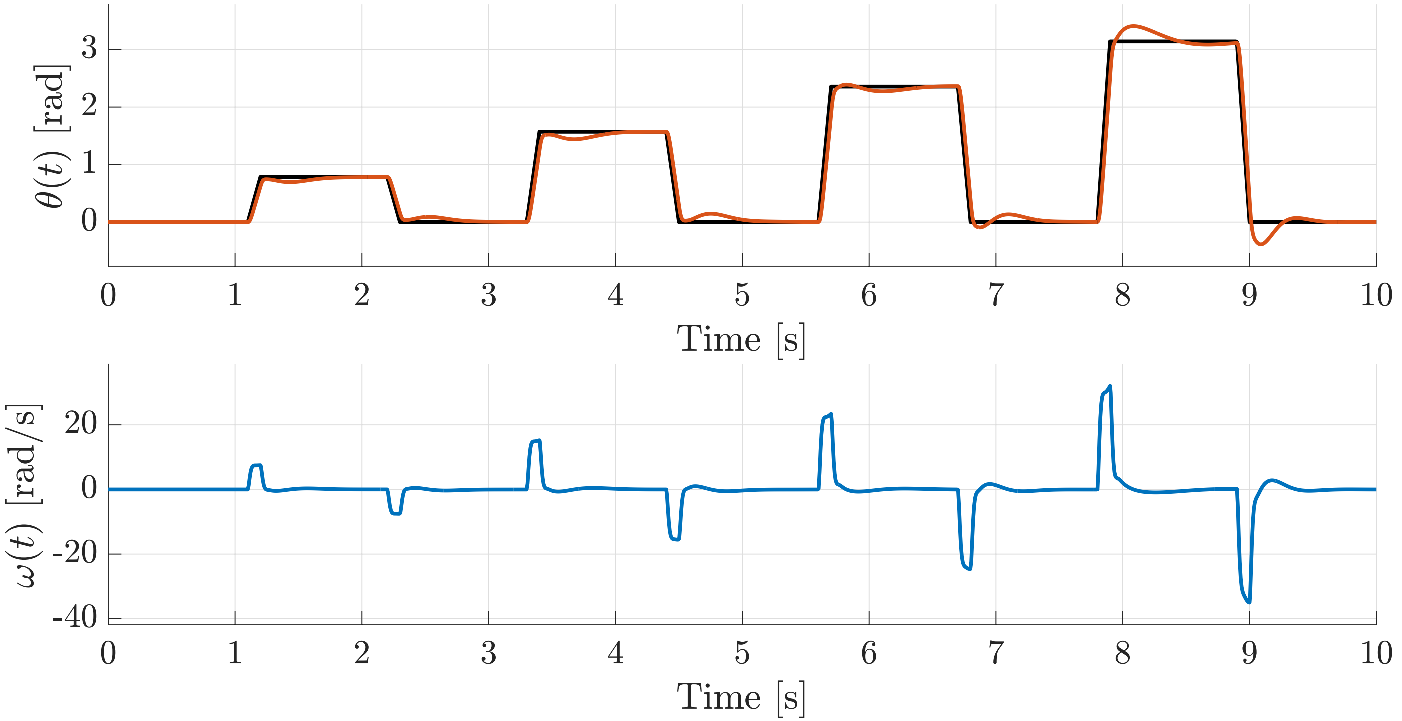
```
% Interconnection structure
P = connect(UnbalancedDisk, sumblk('e = r - theta'), ...
                            sumblk('u = d + yk'), ...
                            {'r','d','yk'},{'e','yk','e'});
% Make generalized (weighted) plant
Pw = blkdiag(Wz,eye(ny)) * P * blkdiag(Ww,eye(nu));

% Synthesize!
[K, gamma, X] = lpvsyn(Pw, ny, nu);

% gamma = 1.41
```





LPV control of the unbalanced disc – Simulation

Performance and stability over full operating range!



LPV synthesis comments

- Similar procedures exists for polynomial/rational dependencies, with variety of methods (S-proc., IQC's, full-block multipliers)
- Gain-scheduling methods (gridding)

1. Grid the scheduling space
2. Synthesize controller for every grid-point
3. Interpolate controllers using linear, behavioral, spline-based interpolation

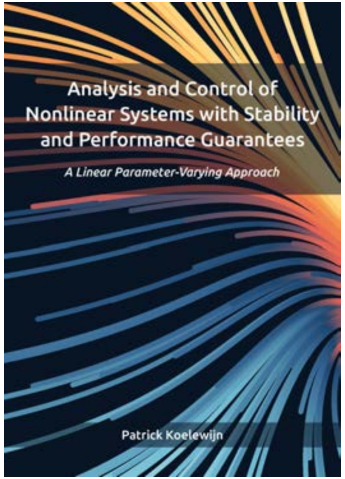
➢ Most standard use of LPV in the industry (available in Matlab)

- Currently working with Mathworks to push this further

Table of contents

Summary and final comments

- LPV modeling enables **linear** analysis and controller synthesis for **nonlinear** and **time-varying** plants.
- Capable to go beyond limitations of LTI controllers (nominal, robust, etc.) by exploiting **measurable** information on $p(t)$
- Compared to NL control, LPV control enables **performance shaping**
  - Recent results use LPV control to go beyond Lyapunov
- Active field of research:
  - Automation & complexity/conservatism reduction of LPV embeddings
  - Machine-learning assisted methods
  - Data-based control (my focus of research 🤓)



List of interesting references:

LPV modeling and identification:

- Tóth (2010). Modeling and Identification of Linear Parameter-Varying Systems, *Springer*.
- Lopes dos Santos, et al. (2011). Linear Parameter-Varying System Identification: New Developments and Trends. *World Scientific.*
- Briat (2015). Linear Parameter-Varying and Time- Delay Systems: Analysis, Observation, Filtering & Control. *Springer.*
- Lovera et al. (2013). LPV modelling and identification: An overview. *In Robust Control and Linear Parameter Varying Approaches: Application to Vehicle Dynamics*, 3-24.
- Van Wingerden & Verhaegen (2009). Subspace identification of bilinear and LPV systems for open-and closed-loop data. *Automatica*, 45(2), 372-381.
- Abbas, et al. (2021). LPV modeling of nonlinear systems: A multi-path feedback linearization approach. *International Journal of Robust and Nonlinear Control.* 31(18), 9436-9465.
- Verhoek, et al. (2022). Deep-learning-based identification of LPV models for nonlinear systems. *In Proc. of the 61st Conference on Decision and Control.*

LPV control:

- Mohammadpour & Scherer (2012). Control of Linear Parameter Varying Systems with Applications. *Springer.*
- Hoffmann & Werner (2015). A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations. *IEEE Transactions on Control Systems Technology*, 23(2), 416–433.
- Scherer (2001). LPV control and full block multipliers. *Automatica*, 37(3), 361-375.
- Steinbuch, et al. (2003). Experimental modelling and LPV control of a motion system. *In Proc. of the 2003 ACC.* Vol. 2 (pp. 1374-1379).
- Koelewijn (2023). Analysis and Control of Nonlinear Systems with Stability and Performance Guarantees: A Linear Parameter-Varying Approach, PhD thesis.

Able to achieve marvelous designs!

https://www.youtube.com/watch?v=vytjdqNpGUM