Internal Model Control of Partial Differential Equations

Lassi Paunonen

including joint work with
Jukka-Pekka Humaloja, Konsta Huhtala, and Duy Phan

Tampere University, Finland

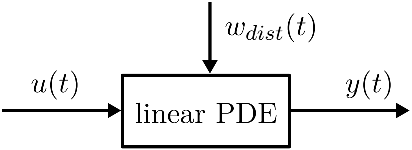
November 2023

Funded by Research Council of Finland grant 349002 (2022–2026)

Introduction

Problem

*Study **robust output regulation** of linear PDE models.*



Output Regulation = Tracking + Disturbance Rejection:
Design a controller such that the output $y(t)$ of the system converges to a reference signal despite the disturbance $w_{dist}(t)$, i.e.,

$$\|y(t) - y_{ref}(t)\| \to 0, \qquad \text{as} \quad t \to \infty$$

Robustness: The controller is required to tolerate uncertainty in the parameters of the system.

Applications

Applications of regulation for PDEs:

- Temperature tracking control, e.g., in manufacturing processes
- Tracking control of flexible robotic manipulators
- Rejection of unwanted periodic noises or vibrations

Robustness:

- Tolerance to the unavoidable **uncertainty** in models.
- Allows reliable use of **approximate** controller parameters.

Goal of the Talk

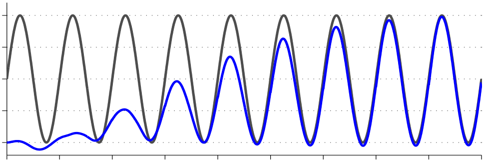
- Highlight differences between internal model control for (linear) ODE and PDE systems
- Discuss selected approaches to controller design
- Examples in tutorial style

The Reference and Disturbance Signals

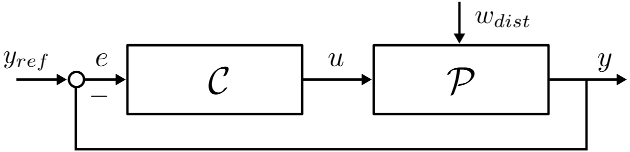
The reference and disturbance signals are of the form

$$y_{ref}(t) = \sum_{k=0}^{q} a_k \cos(\omega_k t + \theta_k)$$
$$w_{dist}(t) = \sum_{k=0}^{q} b_k \cos(\omega_k t + \varphi_k)$$

with **known frequencies** $0 = \omega_0 < \omega_1 < \cdots < \omega_q$ and unknown amplitudes and phases.



The Dynamic Error Feedback Controller



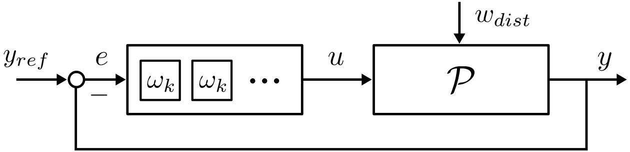
We consider a dynamic error feedback controller which is another linear system.

Theorem

The Robust Output Regulation Problem is solvable if the system

- *is stabilizable and detectable*
- *does not have transmission zeros at the frequencies* $\pm i\omega_k$ *of* $y_{\mathit{ref}}(t)$ *and* $w_{\mathit{dist}}(t)$.

The Internal Model Principle



Theorem (Francis–Wonham, Davison 1970's, . . .)

The following are equivalent:

- *The controller solves the robust output regulation problem.*
- *Closed-loop system is stable and the controller has* ***an internal model*** *of the frequencies* $\{\omega_k\}_k$ *of* $w_{\textsf{dist}}(t)$ *and* $y_{\textsf{ref}}(t)$.

*"****Internal Model****": For every* k*, the complex frequencies* $\pm i\omega_k$ *must be eigenmodes of the controller dynamics with at least* $p = \dim Y$ *independent eigenvectors.*

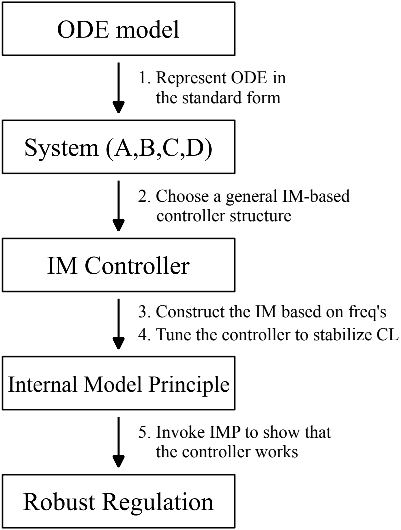
Internal Model Based Controller Design

The robust output regulation problem can be solved in two parts:

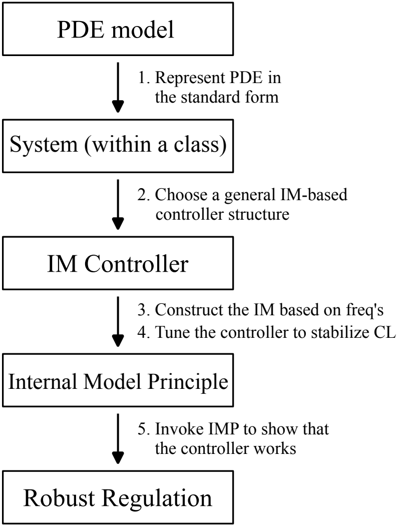
- Step 1° Include a suitable internal model into the controller
- Step 2° Use the rest of the controller's parameters to stabilize the closed-loop system.

Internal model has fixed structure (easy), the closed-loop stability can be achieved in several ways (often the main challenge).

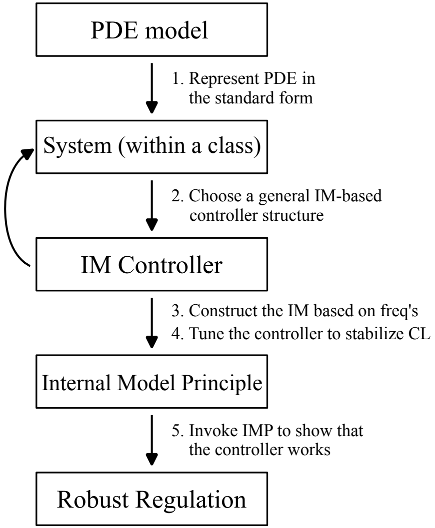
Robust Regulation of Linear ODE Systems



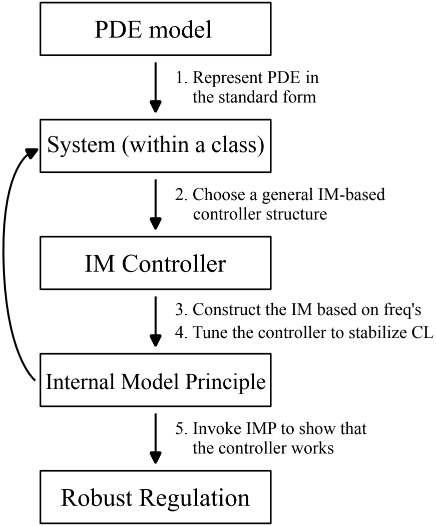
Robust Regulation of linear PDE systems



Robust Regulation of linear PDE systems



Robust Regulation of linear PDE systems



The Possible Types of "Standard Representations"

1. "**A class of PDEs**"
   - A parametrised collection of PDEs of the same type (parabolic, hyperbolic, coupled)
   - Allows various input/output configurations
   - Examples: Distributed port-Hamiltonian systems, $n \times m$ hyperbolic systems, reaction-convection-diffusion equations

The Possible Types of "Standard Representations"

1. "**A class of PDEs**"
    - A parametrised collection of PDEs of the same type (parabolic, hyperbolic, coupled)
    - Allows various input/output configurations
    - Examples: Distributed port-Hamiltonian systems, $n \times m$ hyperbolic systems, reaction-convection-diffusion equations

2. "**Abstract linear systems**" (or "DPS")
    - Represents the PDE as a linear system (A, B, C, D) but on an *infinite-dimensional* space X
    - Different subclasses based on properties of the *operators*
    - Examples: Regular Linear Systems, Well-Posed Systems

Comparison of Internal Model Controllers

1. **A class of PDEs:**
   - Currently there's no actual "IMP" for PDE classes
   - **but** IM-based controllers (with tuning recipes) exist!

2. **Abstract linear systems:**
   - Internal Model Principle exists for several system classes.
   - Several different types of IM controllers

Comparison of Internal Model Controllers

1. **A class of PDEs:**
   - Currently there's no actual "IMP" for PDE classes
   - **but** IM-based controllers (with tuning recipes) exist!
   - **Benefit:** Usually straightforward design!
2. **Abstract linear systems:**
   - Internal Model Principle exists for several system classes.
   - Several different types of IM controllers
   - **Trade-off:** Sometimes technically demanding design.

Part II: Controller Design for PDEs

- Illustrate how controller design works for concrete PDEs
- Keep the PDE models simple
- Swipe a lot of technical terms and details under the rug

The "Simplest Example"

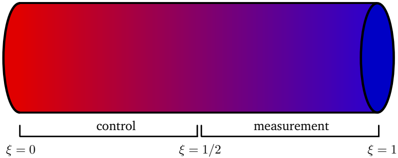
Consider a one-dimensional heat equation

$$\partial_t v(\xi,t) = \partial_{\xi\xi} v(\xi,t) + b(\xi)u(t) + b_d(\xi)w_{\mathit{dist}}(t), \qquad \xi \in (0,1)$$

$$\partial_\xi v(0,t) = \partial_\xi v(1,t) = 0, \qquad v(\xi,0) = v_0(\xi)$$

$$y(t) = \int_0^1 v(\xi,t)c(\xi)d\xi$$

with piecewise continuous functions $b, b_d, c : [0,1] \to \mathbb{R}$.



The "Simplest Example"

Consider a one-dimensional heat equation

$$\partial_t v(\xi,t) = \partial_{\xi\xi} v(\xi,t) + b(\xi)u(t) + b_d(\xi)w_{\mathit{dist}}(t), \qquad \xi \in (0,1)$$

$$\partial_\xi v(0,t) = \partial_\xi v(1,t) = 0, \qquad v(\xi,0) = v_0(\xi)$$

$$y(t) = \int_0^1 v(\xi,t)c(\xi)d\xi$$

with piecewise continuous functions $b, b_d, c : [0,1] \to \mathbb{R}$.

Choose $x(t) = v(\cdot,t)$ and look for a representation

$$\dot{x}(t) = Ax(t) + Bu(t) + B_d w_{\mathit{dist}}(t), \qquad x(0) = x_0 \in X$$

$$y(t) = Cx(t)$$

Abstract Representation

The heat equation can be rewritten as a linear system

$$\dot{x}(t) = Ax(t) + Bu(t) + B_d w_{\textit{dist}}(t), \qquad x(0) = x_0 \in X$$
$$y(t) = Cx(t)$$

if we choose

- State $x(t) = v(\cdot, t)$ (heat profile at $t \geq 0$),
- State space $X = L^2(0,1)$ (Hilbert), $U = Y = U_d = \mathbb{C}$
- Operators $A : \mathcal{D}(A) \subset X \to X$, B, B_d, and C

$$Af = \frac{d^2 f}{d\xi^2}, \quad \mathcal{D}(A) = \{ f \in H^2(0,1) \mid f'(0) = f'(1) = 0 \}$$

$$Bu = b(\cdot)u, \qquad B_d u = b_d(\cdot)u \qquad Cf = \int_0^1 f(\xi)c(\xi)d\xi$$

Internal Model Controller design

Simplifying assumptions:

- $y_{ref}(t) = a\sin(\omega t + \theta)$, single freq $\omega > 0$ and $w_{dist}(t) \equiv 0$
- SISO systems only (generalisation to MIMO easy)
- Use one particular “observer-based” controller structure (others exist too!)

Controller Design for the Heat System

Theorem (Immonen 2007, Hämäläinen–Pohjolainen 2010)

The Robust Output Regulation Problem can be solved with the internal model controller

$$\begin{aligned}
\dot{z}_1(t) &= G_1 z_1(t) + G_2(y(t) - y_{ref}(t)) \\
\dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + L(\hat{y}(t) - y(t) + y_{ref}(t)) \\
\hat{y}(t) &= C\hat{x}(t) \\
u(t) &= K_1 z_1(t) + K_2 \hat{x}(t)
\end{aligned}$$

with matrices G_1, G_2, K_1 and bounded operators L and K_2.

The matrices (G_1, G_2) contain the internal model,

$$G_1 = \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix}, \qquad G_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Controller Parameters L, K_1 and K_2

The operators L, K_1 and K_2: Chosen so that the systems

$$\dot{x}(t) = (A + LC)x(t) \qquad \rightsquigarrow \quad \text{output injection}$$

and

$$\dot{x}(t) = \left(\begin{bmatrix} G_1 & G_2 C \\ 0 & A \end{bmatrix} + \begin{bmatrix} 0 \\ B \end{bmatrix} \begin{bmatrix} K_1, K_2 \end{bmatrix} \right) x(t)$$

are exponentially stable.

Controller Parameters L, K_1 and K_2

Stabilize

$$\dot{x}(t) = \left(\begin{bmatrix} G_1 & G_2 C \\ 0 & A \end{bmatrix} + \begin{bmatrix} 0 \\ B \end{bmatrix} \begin{bmatrix} K_1, K_2 \end{bmatrix} \right) x(t)$$

Can be rewritten as feedback stabilization of a PDE-ODE cascade:

$$\begin{aligned}
\dot{z}_1(t) &= G_1 z_1(t) + G_2 \int_0^1 v(\xi,t) c(\xi) d\xi \\
\partial_t v(\xi,t) &= \partial_{\xi\xi} v(\xi,t) + b(\xi) K_1 z_1(t) + b(\xi) \int_0^1 v(\xi,t) k_2(\xi) d\xi \\
\partial_\xi v(0,t) &= \partial_\xi v(1,t) = 0,
\end{aligned}$$

Possible approaches:

- Numerical approximations and LQR (Banks–Kunisch '84)
- "**Forwarding**" (+ numerical approx)

The PDE Controller

Finally, rewrite the "abstract controller" as a PDE-ODE system:

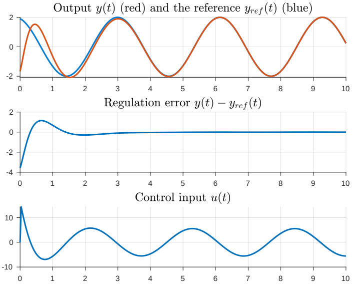
$$\dot{z}_1(t) = \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix} z_1(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} (y(t) - y_{ref}(t))$$

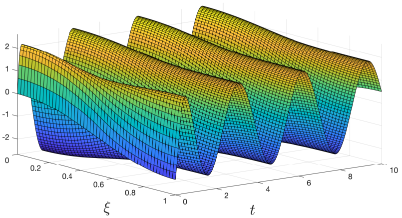
$$\partial_t \hat{v}(\xi,t) = \partial_{\xi\xi} \hat{v}(\xi,t) + b(\xi) K_1 z_1(t) + b(\xi) \int_0^1 k_2(\xi) \hat{v}(\xi,t) d\xi + \ell(\xi) \Big(\int_0^1 c(\xi) \hat{v}(\xi,t) d\xi - y(t) + y_{ref}(t) \Big)$$

$$u(t) = K_1 z_1(t) + \int_0^1 k_2(\xi) \hat{v}(\xi,t) d\xi, \quad \partial_\xi \hat{v}(\xi,t) = \partial_\xi \hat{v}(\xi,t) = 0.$$

(Here we use the knowledge of X, A, B, C etc.)

Simulation Results





RORPack – Matlab/Python libraries for Robust Output Regulation
Available at `https://github.com/lassipau/rorpack-matlab/`

Example: What did we learn?

1. Representation of the PDE as an abstract system was "easy"
2. Choices of L, K_1 and K_2 could be completed by solving stabilization problems for a PDE and a PDE-ODE cascade
3. The abstract controller that we constructed could be reinterpreted as a PDE-ODE system

Example: What did we learn?

1. Representation of the PDE as an abstract system was “easy”
2. Choices of L, K_1 and K_2 could be completed by solving stabilization problems for a PDE and a PDE-ODE cascade
3. The abstract controller that we constructed could be reinterpreted as a PDE-ODE system

Why was this a “simple” example?

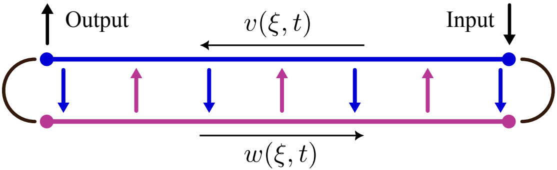
- The control and observation were **inside the spatial domain**

A More Challenging Case

Consider the 2×2 hyperbolic system

$$\begin{aligned}
&\partial_t v(\xi,t) - \mu_1 \partial_\xi v(\xi,t) = a_1(\xi) w(\xi,t)\\
&\partial_t w(\xi,t) + \mu_2 \partial_\xi w(\xi,t) = a_2(\xi) v(\xi,t)\\
&v(0,t) = q_0 w(0,t), \qquad v(1,t) = q_1 w(1,t) + u(t)\\
&y(t) = v(0,t)
\end{aligned}$$

with $\mu_1, \mu_2 > 0$, $q_0, q_1 \in \mathbb{R}$.



- The model has **boundary control** and **boundary observation**

Representation as an Abstract System

The hyperbolic system

$$\begin{aligned} &\partial_t v - \mu_1 \partial_\xi v = a_1 w, && v(0,t) = q_0 w(0,t) + w_{dist}(t), \\ &\partial_t w + \mu_2 \partial_\xi w = a_2 v, && v(1,t) = q_1 w(1,t) + u(t), \\ &y(t) = v(0,t) \end{aligned}$$

Choose $x(t) = (v(\cdot,t), w(\cdot,t))^T \rightsquigarrow$ A **regular linear system**

$$\begin{aligned} &\dot{x}(t) = Ax(t) + Bu(t) + B_d w_{dist}(t), && x(0) = x_0 \in X \\ &y(t) = C_\Lambda x(t) \end{aligned}$$

on $X = L^2(0,1) \times L^2(0,1)$ with $u(t), y(t) \in \mathbb{C}$.

Question

“What’s so difficult?”

Representation as an Abstract System

$$
\begin{aligned}
\partial_t v - \mu_1 \partial_\xi v &= a_1 w, & v(0,t) &= q_0 w(0,t), \\
\partial_t w + \mu_2 \partial_\xi w &= a_2 v, & w(1,t) &= q_1 v(1,t) + u(t), \\
y(t) &= v(0,t)
\end{aligned}
$$

regular linear system on $X = L^2(0,1)^2$

$$
\begin{aligned}
\dot{x}(t) &= Ax(t) + Bu(t) \qquad x(0) = x_0 \in X \\
y(t) &= C_\Lambda x(t)
\end{aligned}
$$

Question (“What’s so difficult?”)

(1) ***Boundary input and output*** $\rightsquigarrow$ *the abstract class is “**large**”*

(2) *Stabilization requires **boundary feedback**.*

Same applies to undamped wave, plate and beam equations.

Overview

Question ("What's so difficult?")

(1) ***Boundary input and output*** $\rightsquigarrow$ *the abstract class is "**large**"*

(2) *Stabilization requires **boundary feedback**.*

The good news: An IMP for regular linear systems exists [P. '16]

The bad news: The existing controller constructions do not manage boundary feedbacks well.

Overview

Question ("What's so difficult?")

(1) ***Boundary input and output*** $\rightsquigarrow$ *the abstract class is "**large**"*

(2) *Stabilization requires **boundary feedback**.*

The good news: An IMP for regular linear systems exists [P. '16]

The bad news: The existing controller constructions do not manage boundary feedbacks well.

Result (P. 2023+)

New abstract controller design techniques for regular linear systems allowing boundary feedback.

Controller Design for the Hyperbolic System

Theorem (P. 2023+)

The Robust Output Regulation Problem can be solved with the internal model controller ***which formally resembles***

$$\begin{aligned}
\dot{z}_1(t) &= G_1 z_1(t) + G_2(y(t) - y_{ref}(t)) \\
\dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + L(\hat{y}(t) - y(t) + y_{ref}(t)) \\
\hat{y}(t) &= C\hat{x}(t) \\
u(t) &= K_1 z_1(t) + K_2 \hat{x}(t)
\end{aligned}$$

with matrices G_1, G_2, K_1 and ***unbounded*** *operators L and K_2.*

- Matrices (G_1, G_2) contain the internal model
- L contains output injections for the hyperbolic system
- K_1 and K_2 stabilize a cascade of the system and the IM

Choosing output injection L

L can be constructed by choosing $\ell_b \in \mathbb{R}$, $\ell_1, \ell_2 : [0,1] \to \mathbb{R}$ so that

$$\begin{aligned}
&\partial_t v - \mu_1 \partial_\xi v = a_1 w + \ell_1(\xi) v(0,t), \\
&\partial_t w + \mu_2 \partial_\xi w = a_2 v + \ell_2(\xi) v(0,t), \\
&v(0,t) = q_0 w(0,t) + \ell_b v(0,t), \qquad w(1,t) = q_1 v(1,t)
\end{aligned}$$

is exponentially stable.

This problem can be solved with **backstepping**.

Choosing Feedback Operators K_1 and K_2

Choose $K_1 \in \mathbb{R}^{2\times 2}$, $k_b \in \mathbb{R}$, $k_1, k_2 : [0,1] \to \mathbb{R}$ that the cascade

$$
\begin{aligned}
\dot{z}_1(t) &= G_1 z_1(t) + G_2 v(0,t) \\
\partial_t v - \mu_1 \partial_\xi v &= a_1 w, \\
\partial_t w + \mu_2 \partial_\xi w &= a_2 v, \qquad\qquad v(0,t) = q_0 w(0,t)
\end{aligned}
$$

$$
\begin{aligned}
w(1,t) = q_1 v(1,t) &+ K_1 z_1(t) + k_b v(1,t) \\
&+ \int_0^1 k_1(\xi) v(\xi,t) + k_2(\xi) w(\xi,t) d\xi
\end{aligned}
$$

is exponentially stable.

This problem can either be solved directly with **backstepping** cascade techniques, or reduced using **forwarding** to stabilization of the hyperbolic system.

The PDE Controller

Finally, rewrite the abstract controller as a PDE-ODE system:

$$\dot{z}_1(t) = \begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix} z_1(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} (v(0,t) - y_{ref}(t))$$

$$\partial_t \hat{v} - \mu_1 \partial_\xi \hat{v} = a_1 \hat{w} + \ell_1(\xi)(\hat{v}(0,t) - v(0,t) + y_{ref}(t)),$$

$$\partial_t \hat{w} + \mu_2 \partial_\xi \hat{w} = a_2 \hat{v} + \ell_2(\xi)(\hat{v}(0,t) - v(0,t) + y_{ref}(t)),$$

$$\hat{v}(0,t) = q_0 \hat{w}(0,t) + \ell_b(\hat{v}(0,t) - v(0,t) + y_{ref}(t))$$

$$\hat{w}(1,t) = q_1 \hat{v}(1,t) + K_1 z_1(t) + k_b \hat{v}(1,t) + \int_0^1 k_1(\xi)\hat{v}(\xi,t) + k_2(\xi)\hat{w}(\xi,t) d\xi$$

(Here we use the knowledge of X, A, B, C etc.)

Discussion on the Examples

- We focused on two 1D examples with differing properties
- Note that **boundary control** is not always difficult!

Easy cases:

- If the system is **already stable**, can use a very easy **finite-dimensional controller** by Rebarber–Weiss '03 (for a large class of systems)
- For parabolic PDEs, you can often design a **finite-dim. controller** using Galerkin approximations even in the unstable case [P.–Phan '20, '21, Huhtala–P.–Hu '22]
- In PDEs with boundary control, **actuator and sensor dynamics** make things easier*!

Historical Highlights Related to PDEs

Internal Model Principle (*characterization* of controllers) for PDEs:

- **Starting point:** IMP for linear finite-dimensional systems
  - Francis–Wonham '75, Davison '76:
- **Extension to PDEs** with distributed control and observation
  - Bhat '76: Geometric approach, PhD with Koivo and Wonham
  - Immonen '05–'07: Approach using Sylvester equations
  - P.–Pohjolainen '10: The "classical" form of the IMP
- **Extension to PDEs** with boundary control and observation
  - P. '14, '16: The class of Regular Linear Systems
  - Humaloja–Kurula–P. '19: Boundary Control Systems
- **Frequency domain extensions** of the IMP
  - Nett '84, Yamamoto–Hara '88, Vidyasagar '88, Laakkonen '13

Developments: Internal Model based control design for PDEs

- **Classes of linear PDEs** with distributed/boundary control
  - Pohjolainen '83, Logemann–Townley '97, Hämäläinen–Pohjolainen '00, '06, '10, Rebarber–Weiss '03, Boulite et. al. '09, Harkort–Deutscher '11, '17, P. '16, '17
- **Parabolic PDEs**
  - Chentouf et. al. '08, '10, Deutscher '13, '15, '16, Guo-Meng '20, Huhtala–P.–Hu '22, ...
- **Hyperbolic PDEs**
  - Guo–Guo '13, '16, Guo–Krstic '17, '18, Humaloja–Kurula–P. '18,'19, Deutscher–Gabriel '18–21, Wang et. al. '18, '21, Guo-Meng '21, '22, ...
- **PIDEs, Coupled systems, Networks, ...**

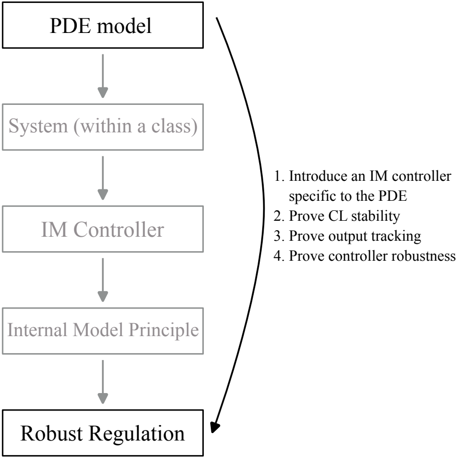
In addition: Controllers for regulation without robustness

- Schumacher '83, Byrnes et. al. '00, Boulite–Saij et. al. '13, '18, Natarajan et. al. '14, Xu–Dubljevic '16, '17, ...

Our "Abstract Approach" vs. "PDE-Based Approach"

In the "*PDE-based approach*", an IM controller is designed for each PDE separately, followed by proofs for closed-loop stability, tracking and robustness.

In comparison, our "abstract approach" **avoids repetition** since regulation and robustness are guaranteed by the IMP, and do not need to be **proved separately** for each PDE!



Conclusions

- An overview of the "abstract approach" to internal model control of linear PDEs.
- Examples of the control design process.
- An invitation to use the techniques in regulation of PDEs!
- Here the examples were chosen to be simple, but in principle, *if you can stabilize, you can do internal model control*.

Things to Discuss

- How could the abstract results be made more user-friendly?
- How could we popularize internal model control?
- Interesting applications?
- Non-linear or semi-linear cases?