Some insights on the practical control of hyperbolic systems

Jean Auriol

L2S, CNRS, Université Paris-Saclay, UMR 8506

May 8, 2023

Introduction: Hyperbolic systems

- Phenomena with **finite propagation speeds**: waves, balance laws, conservation laws.

$$\partial_{tt} w(t,x) - c^2 \partial_{xx} w(t,x) = 0.$$

- Examples: mass, charge, energy, momentum
- Complex engineering problems: stabilization, observer design, parameter estimation.

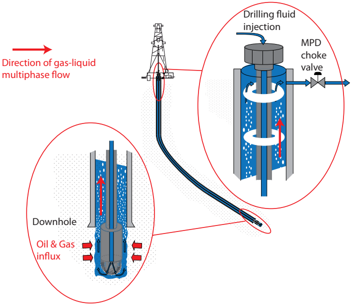
Stringent operating, environmental and economical requirement

- Phenomena with **finite propagation speeds**: waves, balance laws, conservation laws.

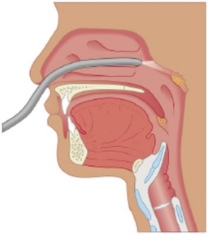
$$\partial_{tt} w(t,x) - c^2 \partial_{xx} w(t,x) = 0.$$

- Examples: mass, charge, energy, momentum
- Complex engineering problems: stabilization, observer design, parameter estimation.

Stringent operating, environmental and economical requirement



Control and estimation of mechanical vibrations



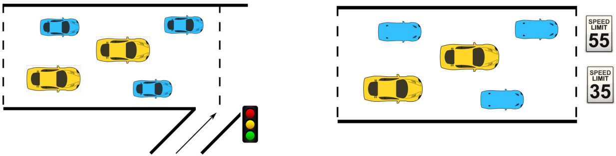
Control of a micro-endoscope

- Phenomena with **finite propagation speeds**: waves, balance laws, conservation laws.

$$\partial_{tt} w(t,x) - c^2 \partial_{xx} w(t,x) = 0.$$

- Examples: mass, charge, energy, momentum
- Complex engineering problems: stabilization, observer design, parameter estimation.

Stringent operating, environmental and economical requirement



Traffic congestion control (avoid stop-and-go oscillations)

Introduction: general objective

Objective

Develop a **systematic framework** for the **practical control** of hyperbolic systems

- Design of explicit control laws: **constructive methods.**
- Easily implementable strategies: **low computational burden.**
- Possible real implementation: **performance specifications.**

Introduction: general objective

Objective

Develop a **systematic framework** for the **practical control** of hyperbolic systems

- Design of explicit control laws: **constructive methods.**
- Easily implementable strategies: **low computational burden.**
- Possible real implementation: **performance specifications.**

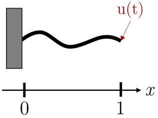
Multiple theoretical approaches

- Optimization controllers [Russel, Lions]
- Lyapunov-based controllers [Bastin, Coron, Prieur]
- Flatness-based methods [Meurer]
- **Backstepping controllers** [Krstic]

Breakthroughs but several practical **limitations.**

Toy problem: clamped string

Toy system: **clamped string** with indefinite in-domain damping and space-varying coefficients
$\longrightarrow$ towards generalization to more complex systems



$$\rho(x)\frac{\partial^2 w}{\partial t^2}(t,x) = \frac{\partial}{\partial x}\left(E(x)\frac{\partial w}{\partial x}(t,x)\right) - \kappa(x)\frac{\partial w}{\partial t}(t,x)$$

$\rho(x)$ mass density, $E(x)$ Young's modulus $\in C^1([0,1])^+$, $\kappa(x) \in C^0([0,1])$ in-domain damping.

Boundary conditions:

- No movement in $x = 0$: $\frac{\partial w}{\partial t}|_{x=0}(t) = 0$
- Torque control input in $x = 1$: $E(1)\frac{\partial w}{\partial x}|_{x=1}(t) = U(t)$

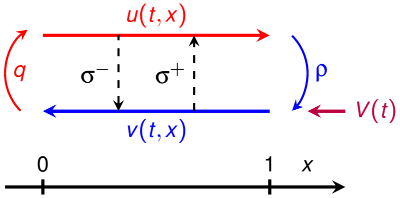
Initial conditions: $w(0,x) = w_0(x) \in C^1([0,1])$.

Riemann coordinates: $u(t,x) = w_t(t,x) - \sqrt{\frac{E(x)}{\rho(x)}}w_x(t,x)$, $v(t,x) = w_t(t,x) + \sqrt{\frac{E(x)}{\rho(x)}}w_x(t,x)$

System under consideration

System of **scalar balance laws**: simple test case to present generic concepts

$$u_t(t,x)+\lambda(x)u_x(t,x)=\sigma^{++}(x)u(t,x)+\sigma^{+}(x)v(t,x),$$
$$v_t(t,x)-\mu(x)v_x(t,x)=\sigma^{--}(x)v(t,x)+\sigma^{-}(x)u(t,x),$$
$$u(t,0)=qv(t,0) \quad v(t,1)=\rho u(t,1)+V(t).$$



- Diagonal terms can be removed with exp. change of coordinates.
- Couplings $\rightarrow$ instability.
- Distributed states and boundary control.

Outline of the presentation

1. An introduction to the backstepping approach
2. Design of robust control laws for hyperbolic systems
3. Development of easily parametrizable target systems.
4. Integration, approximation, and model reduction.

Backstepping for PDEs

- Extension of finite-dimensional backstepping [Krstic et al.; 1995]
- Introduced for parabolic PDEs [Balogh, Krstic; 2002]
- Second-order hyperbolic PDEs [Krstic et al.; 2006]
- First-order hyperbolic PDEs [Krstic, Smyshlyaev; 2008]
- Systems of First-order hyperbolic PDEs [Vazquez; 2012], [Di Meglio et al.; 2013]

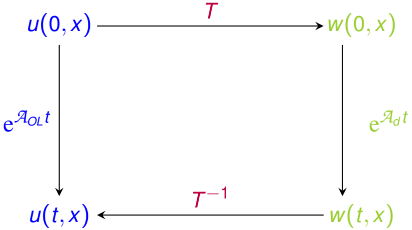
Backstepping methodology

Main idea Use an integral transform (classically Volterra transform of the *second kind*):

$$w(t,x) = u(t,x) - \int_0^x p(x,y)u(t,y)dy$$

to map the original system (to stabilize) to a stable target system .

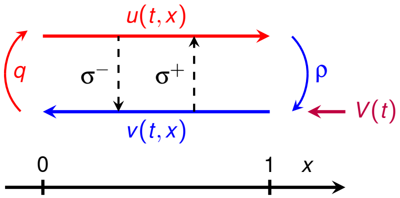
$\longrightarrow$ Constructive design of control laws!



Limitations

- Choice of an adequate target system
- Proof of existence and invertibility of an adequate transform
- Control effort, closed-loop properties and implementation

$$u_t(t,x) + \lambda u_x(t,x) = \sigma^+ v(t,x),$$
$$v_t(t,x) - \mu v_x(t,x) = \sigma^- u(t,x),$$
$$u(t,0) = qv(t,0) \quad v(t,1) = \rho u(t,1) + V(t).$$

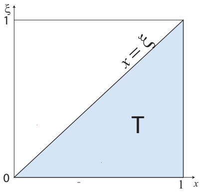


$$u_t(t,x) + \lambda u_x(t,x) = \sigma^+ v(t,x),$$
$$v_t(t,x) - \mu v_x(t,x) = \sigma^- u(t,x),$$
$$u(t,0) = qv(t,0) \quad v(t,1) = \rho u(t,1) + V(t).$$

- Map the original system to a *target system* for which the stability analysis is easier.
- Variable change: integral transformation.

Example: $\alpha(t,x) = u(t,x) - \int_0^x K^{uu}(x,\xi)u(t,\xi) + K^{uv}(x,\xi)v(t,\xi)d\xi$

$$\beta(t,x) = v(t,x) - \int_0^x K^{vu}(x,\xi)u(t,\xi) + K^{vv}(x,\xi)v(t,\xi)d\xi$$



Example of two scalar equations: backstepping transformation

$$
\begin{aligned}
&u_t(t,x) + \lambda u_x(t,x) = \sigma^+ v(t,x), \\
&v_t(t,x) - \mu v_x(t,x) = \sigma^- u(t,x), \\
&u(t,0) = qv(t,0) \quad v(t,1) = \rho u(t,1) + V(t).
\end{aligned}
$$

- Map the original system to a *target system* for which the stability analysis is easier.
- Variable change: integral transformation.

Example:

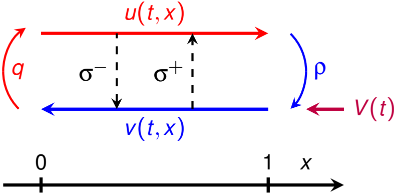
$$
\begin{aligned}
\alpha(t,x) &= u(t,x) - \int_0^x K^{uu}(x,\xi)u(t,\xi) + K^{uv}(x,\xi)v(t,\xi)d\xi \\
\beta(t,x) &= v(t,x) - \int_0^x K^{vu}(x,\xi)u(t,\xi) + K^{vv}(x,\xi)v(t,\xi)d\xi
\end{aligned}
$$

Difficulties:

- Find the target system.
- Existence of the kernel K (set of PDEs to be satisfied).

Objective: Move the in-domain coupling terms at the actuated boundary.

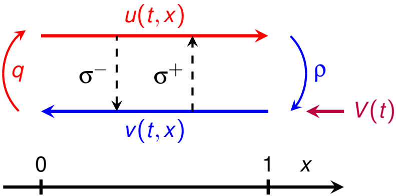
$$u_t(t,x) + \lambda u_x(t,x) = \sigma^+ v(t,x),$$
$$v_t(t,x) - \mu v_x(t,x) = \sigma^- u(t,x).$$



$$u(t,0) = qv(t,0)$$
$$v(t,1) = \rho u(t,1) + V(t)$$

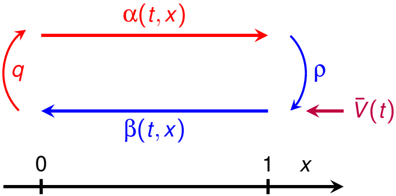
Objective: Move the in-domain coupling terms at the actuated boundary.

$$u_t(t,x)+\lambda u_x(t,x)=\sigma^+ v(t,x),$$
$$v_t(t,x)-\mu v_x(t,x)=\sigma^- u(t,x).$$



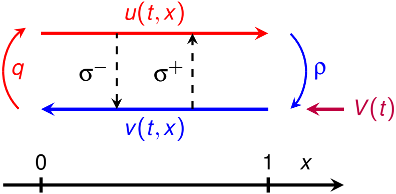
$$u(t,0)=qv(t,0)$$
$$v(t,1)=\rho u(t,1)+V(t)$$

$$\alpha_t(t,x)+\lambda\alpha_x(t,x)=0,$$
$$\beta_t(t,x)-\mu\beta_x(t,x)=0.$$



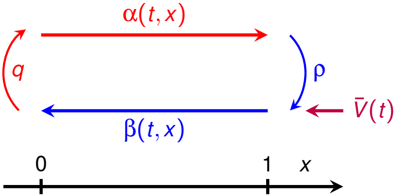
Objective: Move the in-domain coupling terms at the actuated boundary.

$$u_t(t,x) + \lambda u_x(t,x) = \sigma^+ v(t,x),$$
$$v_t(t,x) - \mu v_x(t,x) = \sigma^- u(t,x).$$



$$u(t,0) = qv(t,0)$$
$$v(t,1) = \rho u(t,1) + V(t)$$

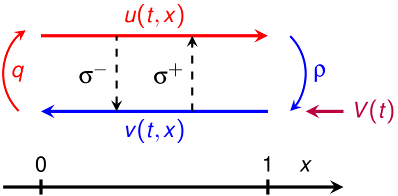
$$\alpha_t(t,x) + \lambda \alpha_x(t,x) = 0,$$
$$\beta_t(t,x) - \mu \beta_x(t,x) = 0.$$



$$\alpha(t,0) = q\beta(t,0)$$
$$\beta(t,1) = \rho\alpha(t,1) + V(t)$$
$$- \int_0^1 N^\alpha(\xi)\alpha(t,\xi) + N^\beta(\xi)\beta(t,\xi) d\xi.$$

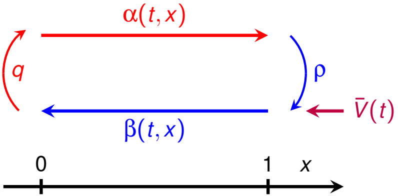
Objective: Move the in-domain coupling terms at the actuated boundary.

$$u_t(t,x)+\lambda u_x(t,x)=\sigma^+ v(t,x),$$
$$v_t(t,x)-\mu v_x(t,x)=\sigma^- u(t,x).$$



$$u(t,0)=qv(t,0)$$
$$v(t,1)=\rho u(t,1)+V(t)$$

$$\alpha_t(t,x)+\lambda\alpha_x(t,x)=0,$$
$$\beta_t(t,x)-\mu\beta_x(t,x)=0.$$



$$\alpha(t,0)=q\beta(t,0)$$
$$\beta(t,1)=\rho\alpha(t,1)+V(t)$$
$$-\int_0^1 N^\alpha(\xi)\alpha(t,\xi)+N^\beta(\xi)\beta(t,\xi)d\xi.$$

Natural control law

$$V(t)=-\rho\alpha(t,1)+\int_0^1\left(N^\alpha(\xi)\alpha(t,\xi)+N^\beta(\xi)\beta(t,\xi)\right)d\xi.$$

Backstepping transformation

$$\beta(t,x) = v(t,x) - \int_0^x K^{vu}(x,\xi)u(t,\xi) + K^{vv}(x,\xi)v(t,\xi)d\xi.$$

Backstepping transformation

$$\beta(t,x) = v(t,x) - \int_0^x K^{vu}(x,\xi)u(t,\xi) + K^{vv}(x,\xi)v(t,\xi)d\xi.$$

We have: $v_t - \mu v_x = \sigma^- u$

We want: $\beta_t - \mu\beta_x = 0$

Backstepping transformation

$$\beta(t,x) = v(t,x) - \int_0^x K^{vu}(x,\xi)u(t,\xi) + K^{vv}(x,\xi)v(t,\xi)d\xi.$$

We have: $v_t - \mu v_x = \sigma^- u$ We want: $\beta_t - \mu\beta_x = 0$

Differentiation w.r.t space

$$\begin{aligned} -\mu\beta_x(t,x) = & -\mu v_x(t,x) + \mu K^{vu}(x,x)u(t,x) + \mu K^{vv}(x,x)v(t,x) \\ & + \int_0^x \mu K_x^{vu}(x,\xi)u(t,\xi) + \mu K_x^{vv}(x,\xi)v(t,\xi)d\xi \end{aligned}$$

$$\beta(t,x) = v(t,x) - \int_0^x K^{vu}(x,\xi)u(t,\xi) + K^{vv}(x,\xi)v(t,\xi)d\xi.$$

We have: $v_t - \mu v_x = \sigma^- u$ We want: $\beta_t - \mu\beta_x = 0$

Differentiation w.r.t space

$$\begin{aligned} -\mu\beta_x(t,x) = &-\mu v_x(t,x) + \mu K^{vu}(x,x)u(t,x) + \mu K^{vv}(x,x)v(t,x) \\ &+ \int_0^x \mu K_x^{vu}(x,\xi)u(t,\xi) + \mu K_x^{vv}(x,\xi)v(t,\xi)d\xi \end{aligned}$$

Differentiation w.r.t time

$$\begin{aligned} \beta_t(t,x) = v_t(t,x) &- \int_0^x -K^{vu}(x,\xi)\lambda u_x(t,\xi) + K^{vv}(x,\xi)\mu v_x(x,\xi) \\ &- \int_0^x K^{vu}(x,\xi)\sigma^+ v(t,\xi) + K^{vv}(x,\xi)\sigma^- u(t,\xi)d\xi \end{aligned}$$

Backstepping transformation

$$\beta(t,x) = v(t,x) - \int_0^x K^{vu}(x,\xi)u(t,\xi) + K^{vv}(x,\xi)v(t,\xi)d\xi.$$

We have: $v_t - \mu v_x = \sigma^- u$

We want: $\beta_t - \mu\beta_x = 0$

Differentiation w.r.t space

$$-\mu\beta_x(t,x) = -\mu v_x(t,x) + \mu K^{vu}(x,x)u(t,x) + \mu K^{vv}(x,x)v(t,x) + \int_0^x \mu K_x^{vu}(x,\xi)u(t,\xi) + \mu K_x^{vv}(x,\xi)v(t,\xi)d\xi$$

Differentiation w.r.t time

$$\beta_t(t,x) = v_t(t,x) - \int_0^x -K^{vu}(x,\xi)\lambda u_x(t,\xi) + K^{vv}(x,\xi)\mu v_x(x,\xi) - \int_0^x K^{vu}(x,\xi)\sigma^+ v(t,\xi) + K^{vv}(x,\xi)\sigma^- u(t,\xi)d\xi$$

Integration by parts + use of B.Cs.

Backstepping transformation

$$\beta(t,x) = v(t,x) - \int_0^x K^{vu}(x,\xi)u(t,\xi) + K^{vv}(x,\xi)v(t,\xi)d\xi.$$

We have: $v_t - \mu v_x = \sigma^- u$

We want: $\beta_t - \mu\beta_x = 0$

Differentiation w.r.t space

$$\begin{aligned}-\mu\beta_x(t,x) = &-\mu v_x(t,x) + \mu K^{vu}(x,x)u(t,x) + \mu K^{vv}(x,x)v(t,x) \\ &+ \int_0^x \mu K_x^{vu}(x,\xi)u(t,\xi) + \mu K_x^{vv}(x,\xi)v(t,\xi)d\xi\end{aligned}$$

$$\begin{aligned}&\beta_t(t,x) = \mu v_x(t,x) + \sigma^- u(t,x) + \lambda K^{vu}(x,x)u(t,x) - \mu K^{vv}(x,x)v(t,x) \\ &-\lambda K^{vu}(x,0)u(t,0) - \mu K^{vv}(x,0)v(t,0) - \int_0^x K_\xi^{vu}(x,\xi)\lambda u(t,\xi)d\xi \\ &- \int_0^x K_\xi^{vv}(x,\xi)\mu v(t,\xi) + K^{vu}(x,\xi)\sigma^+ v(t,\xi) + K^{vv}(x,\xi)\sigma^- u(t,\xi)d\xi\end{aligned}$$

$$\beta(t,x) = v(t,x) - \int_0^x K^{vu}(x,\xi)u(t,\xi) + K^{vv}(x,\xi)v(t,\xi)d\xi.$$

We have: $v_t - \mu v_x = \sigma^- u$

We want: $\beta_t - \mu\beta_x = 0$

Differentiation w.r to space

$$-\mu\beta_x(t,x) = -\mu v_x(t,x) + \mu K^{vu}(x,x)u(t,x) + \mu K^{vv}(x,x)v(t,x) + \int_0^x \mu K_x^{vu}(x,\xi)u(t,\xi) + \mu K_x^{vv}(x,\xi)v(t,\xi)d\xi$$

$$\beta_t(t,x) = \mu v_x(t,x) + \sigma^- u(t,x) + \lambda K^{vu}(x,x)u(t,x) - \mu K^{vv}(x,x)v(t,x) - \lambda K^{vu}(x,0)u(t,0) - \mu K^{vv}(x,0)v(t,0) - \int_0^x K_\xi^{vu}(x,\xi)\lambda u(t,\xi)d\xi - \int_0^x K_\xi^{vv}(x,\xi)\mu v(t,\xi) + K^{vu}(x,\xi)\sigma^+ v(t,\xi) + K^{vv}(x,\xi)\sigma^- u(t,\xi)d\xi$$

Kernel equations

$$
\begin{aligned}
0 &= \beta_t(t,x) - \mu\beta_x(t,x) \\
&= (\sigma^{-+} + \lambda K^{vu}(x,x) + \mu K^{vu}(x,x))u(t,x) + (\lambda K^{vu}(x,0)q - \mu K^{vv}(x,0))v(t,0) \\
&- \int_0^x (\lambda K^{vu}_\xi(x,\xi) - \mu K^{vu}_x(x,\xi) + K^{vv}(x,\xi)\sigma^-)u(t,\xi)d\xi \\
&- \int_0^x (\mu K^{vv}_\xi(x,\xi) + \mu K^{vv}_x(x,\xi) + K^{vu}(x,\xi)\sigma^+)v(t,\xi)d\xi.
\end{aligned}
$$

Kernel equations

$$
\begin{aligned}
0 &= \beta_t(t,x) - \mu\beta_x(t,x) \\
&= (\sigma^{-+} + \lambda K^{vu}(x,x) + \mu K^{vu}(x,x))u(t,x) + (\lambda K^{vu}(x,0)q - \mu K^{vv}(x,0))v(t,0) \\
&- \int_0^x (\lambda K^{vu}_\xi(x,\xi) - \mu K^{vu}_x(x,\xi) + K^{vv}(x,\xi)\sigma^-)u(t,\xi)d\xi \\
&- \int_0^x (\mu K^{vv}_\xi(x,\xi) + \mu K^{vv}_x(x,\xi) + K^{vu}(x,\xi)\sigma^+)v(t,\xi)d\xi.
\end{aligned}
$$

$$
\begin{cases}
\mu K^{vv}_\xi(x,\xi) + \mu K^{vv}_x(x,\xi) = -K^{vu}(x,\xi)\sigma^-, \\
\lambda K^{vu}_\xi(x,\xi) - \mu K^{vu}_x(x,\xi) = -K^{vv}(x,\xi)\sigma^-, \\
\lambda K^{vu}(x,0)q = \mu K^{vv}(x,0), \quad K^{vu}(x,x) = \frac{\sigma^-}{\lambda+\mu}
\end{cases}
$$

Kernel equations

$$
\begin{aligned}
0 &= \beta_t(t,x) - \mu\beta_x(t,x) \\
&= (\sigma^{-+} + \lambda K^{vu}(x,x) + \mu K^{vu}(x,x))u(t,x) + (\lambda K^{vu}(x,0)q - \mu K^{vv}(x,0))v(t,0) \\
&- \int_0^x (\lambda K^{vu}_\xi(x,\xi) - \mu K^{vu}_x(x,\xi) + K^{vv}(x,\xi)\sigma^-)u(t,\xi)d\xi \\
&- \int_0^x (\mu K^{vv}_\xi(x,\xi) + \mu K^{vv}_x(x,\xi) + K^{vu}(x,\xi)\sigma^+)v(t,\xi)d\xi.
\end{aligned}
$$

$$
\begin{cases}
\mu K^{vv}_\xi(x,\xi) + \mu K^{vv}_x(x,\xi) = -K^{vu}(x,\xi)\sigma^-, \\
\lambda K^{vu}_\xi(x,\xi) - \mu K^{vu}_x(x,\xi) = -K^{vv}(x,\xi)\sigma^-, \\
\lambda K^{vu}(x,0)q = \mu K^{vv}(x,0), \quad K^{vu}(x,x) = \frac{\sigma^-}{\lambda+\mu}
\end{cases}
$$

- Integral equations + successive approximations $\rightarrow$ Well-posedness.
- Invertibility of the Volterra transformation.

Finite-time stabilization?

$$\alpha_t(t,x) + \lambda\alpha_x(t,x) = 0, \quad \beta_t(t,x) - \mu\beta_x(t,x) = 0,$$

$$\alpha(t,0) = q\beta(t,0),$$

$$\beta(t,1) = \rho\alpha(t,1) - \int_0^1 \left(N^\alpha(\xi)\alpha(t,\xi) + N^\beta(\xi)\beta(t,\xi) \right) d\xi + V(t).$$

Finite-time stabilization?

$$\alpha_t(t,x) + \lambda\alpha_x(t,x) = 0, \quad \beta_t(t,x) - \mu\beta_x(t,x) = 0,$$

$$\alpha(t,0) = q\beta(t,0),$$

$$\beta(t,1) = \rho\alpha(t,1) - \int_0^1 \left(N^\alpha(\xi)\alpha(t,\xi) + N^\beta(\xi)\beta(t,\xi) \right) d\xi + V(t).$$

$$V(t) = -\rho\alpha(t,1) + \int_0^1 \left(N^\alpha(\xi)\alpha(t,\xi) + N^\beta(\xi)\beta(t,\xi) \right) d\xi$$

$$\beta(t,x) = \beta(t - \frac{1-x}{\mu}, 1) = 0, \quad \alpha(t,x) = \alpha(t - \frac{x}{\lambda}, 0) = q\beta(t - \frac{x}{\lambda}, 0) = 0,$$

The system is finite-time stable!

Finite-time stabilization?

$$\alpha_t(t,x) + \lambda\alpha_x(t,x) = 0, \quad \beta_t(t,x) - \mu\beta_x(t,x) = 0,$$

$$\alpha(t,0) = q\beta(t,0),$$

$$\beta(t,1) = \rho\alpha(t,1) - \int_0^1 \left(N^\alpha(\xi)\alpha(t,\xi) + N^\beta(\xi)\beta(t,\xi)\right) d\xi + V(t).$$

$$V(t) = -\rho\alpha(t,1) + \int_0^1 \left(N^\alpha(\xi)\alpha(t,\xi) + N^\beta(\xi)\beta(t,\xi)\right) d\xi$$

What if we add a delay?

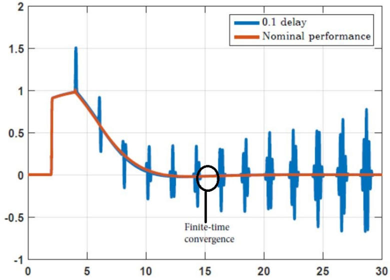
Finite-time stabilization?

$$\alpha_t(t,x)+\lambda\alpha_x(t,x)=0, \quad \beta_t(t,x)-\mu\beta_x(t,x)=0,$$

$$\alpha(t,0)=q\beta(t,0),$$

$$\beta(t,1)=\rho\alpha(t,1)-\int_0^1\left(N^\alpha(\xi)\alpha(t,\xi)+N^\beta(\xi)\beta(t,\xi)\right)d\xi+V(t).$$

$$V(t)=-\rho\alpha(t,1)+\int_0^1\left(N^\alpha(\xi)\alpha(t,\xi)+N^\beta(\xi)\beta(t,\xi)\right)d\xi$$



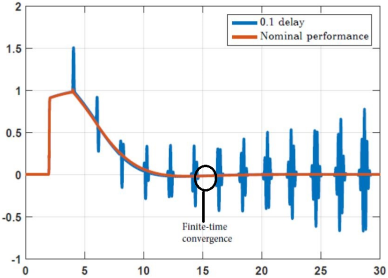
Finite-time stabilization?

$$\alpha_t(t,x) + \lambda\alpha_x(t,x) = 0, \quad \beta_t(t,x) - \mu\beta_x(t,x) = 0,$$

$$\alpha(t,0) = q\beta(t,0),$$

$$\beta(t,1) = \rho\alpha(t,1) - \int_0^1 \left(N^\alpha(\xi)\alpha(t,\xi) + N^\beta(\xi)\beta(t,\xi)\right) d\xi + V(t).$$

$$V(t) = -\rho\alpha(t,1) + \int_0^1 \left(N^\alpha(\xi)\alpha(t,\xi) + N^\beta(\xi)\beta(t,\xi)\right) d\xi$$



Finite time convergence: performance $\gg$ robustness.

$$\alpha_t(t,x) + \lambda\alpha_x(t,x) = 0$$
$$\beta_t(t,x) - \mu\beta_x(t,x) = 0$$

$$\alpha(t,0) = q\beta(t,0)$$
$$\beta(t,1) = \rho\alpha(t,1) - \int_0^1 \left(N^\alpha\alpha(t,\xi) + N^\beta(\xi)\beta(t,\xi)\right) d\xi + V(t)$$

Robustness analysis

$$\alpha_t(t,x) + \lambda\alpha_x(t,x) = 0 \rightarrow \text{Transport equation}$$
$$\beta_t(t,x) - \mu\beta_x(t,x) = 0 \rightarrow \text{Transport equation}$$

$$\alpha(t,0) = q\beta(t,0)$$
$$\beta(t,1) = \rho\alpha(t,1) - \int_0^1 \left(N^\alpha(\xi)\alpha(t,\xi) + N^\beta(\xi)\beta(t,\xi) \right) d\xi + V(t)$$

Robustness analysis

$$\alpha_t(t,x) + \lambda \alpha_x(t,x) = 0 \rightarrow \text{Transport equation}$$
$$\beta_t(t,x) - \mu \beta_x(t,x) = 0 \rightarrow \text{Transport equation}$$

$$\alpha(t,0) = q\beta(t,0)$$
$$\beta(t,1) = \rho\alpha(t,1) - \int_0^1 \left(N^{\alpha}(\xi)\alpha(t,\xi) + N^{\beta}(\xi)\beta(t,\xi) \right) d\xi + V(t)$$

Difference equation satisfied by $\beta(t,1)$

For all $t > \frac{1}{\lambda} + \frac{1}{\mu} = \tau$, we have

$$\beta(t,1) = \rho q \beta(t-\tau,1) - \int_0^{\tau} N(\xi)\beta(t-\xi,1) d\xi + V(t)$$

$$\beta(t,1) = \rho q \beta(t-\tau,1) - \int_0^\tau N(\xi)\beta(t-\xi,1)d\xi + V(t)$$

Delay δ acting on $V(t) = -\rho q \beta(t-\tau,1) + \int_0^1 N(\xi)\beta(t-\xi,1)d\xi$.

$$\beta(t,1) = \rho q \beta(t-\tau,1) - \int_0^\tau N(\xi)\beta(t-\xi,1)d\xi + V(t)$$

Delay δ acting on $V(t) = -\rho q \beta(t-\tau,1) + \int_0^1 N(\xi)\beta(t-\xi,1)d\xi$.

Closed-loop system:

$$\beta(t,1) = \rho q \beta(t-\tau,1) - \rho q \beta(t-\tau-\delta,1) - \int_0^\tau N(\xi)(\beta(t-\xi,1) - \beta(t-\delta-\xi,1))d\xi.$$

Delay-robust stabilization

$$\beta(t,1) = \rho q \beta(t-\tau,1) - \int_0^\tau N(\xi)\beta(t-\xi,1)d\xi + V(t)$$

Delay δ acting on $V(t) = -\rho q \beta(t-\tau,1) + \int_0^1 N(\xi)\beta(t-\xi,1)d\xi$.

Closed-loop system:

$$\beta(t,1) = \rho q \beta(t-\tau,1) - \rho q \beta(t-\tau-\delta,1) - \int_0^\tau N(\xi)(\beta(t-\xi,1) - \beta(t-\delta-\xi,1))d\xi.$$

$\Rightarrow$ Problem if $|\rho q| \geq \frac{1}{2}$

Delay-robust stabilization

$$\beta(t,1) = \rho q \beta(t-\tau,1) - \int_0^\tau N(\xi)\beta(t-\xi,1)d\xi + V(t)$$

Delay δ acting on $V(t) = -\rho q \beta(t-\tau,1) + \int_0^1 N(\xi)\beta(t-\xi,1)d\xi$.

Closed-loop system:

$$\beta(t,1) = \rho q \beta(t-\tau,1) - \rho q \beta(t-\tau-\delta,1) - \int_0^\tau N(\xi)(\beta(t-\xi,1) - \beta(t-\delta-\xi,1))d\xi.$$

$\Rightarrow$ Problem if $|\rho q| \geq \frac{1}{2}$

Solution: renounce to finite-time stabilization

$$\boxed{V(t) = -\tilde{\rho} q \beta(t-\tau,1) + \int_0^1 N(\xi)\beta(t-\xi,1)d\xi}$$

Delay-robustness under the NSC $|\tilde{\rho}| < \frac{1-|\rho q|}{|q|}$.

Open-loop analysis: $V(t) \equiv 0$

$$\beta(t,1) = \rho q \beta(t-\tau,1) - \int_0^{\tau} N(\xi)\beta(t-\xi,1)d\xi$$

Open-loop analysis: $V(t) \equiv 0$

$$\beta(t,1) = \rho q \beta(t-\tau,1) - \int_0^\tau N(\xi)\beta(t-\xi,1)d\xi$$

Open loop analysis

If the open loop transfer function has an infinite number of poles in the RHP, the system cannot be delay-robustly stabilized.

Open-loop analysis: $V(t) \equiv 0$

$$\beta(t,1) = \rho q \beta(t-\tau,1) - \int_0^{\tau} N(\xi)\beta(t-\xi,1)d\xi$$

Open loop analysis

If the open loop transfer function has an infinite number of poles in the RHP, the system cannot be delay-robustly stabilized.

Open-loop characteristic equation:

$$D(s) = \underbrace{1-\rho q e^{-\tau s}}_{F(s)} + \underbrace{\int_0^{\tau} N(\xi) e^{-\xi s} d\xi}_{H(s)} = 0$$

Open-loop analysis: $V(t) \equiv 0$

$$\beta(t,1) = \rho q \beta(t-\tau,1) - \int_0^\tau N(\xi)\beta(t-\xi,1)d\xi$$

Open loop analysis

If the open loop transfer function has an infinite number of poles in the RHP, the system cannot be delay-robustly stabilized.

Open-loop characteristic equation:

$$D(s) = \underbrace{1-\rho q e^{-\tau s}}_{F(s)} + \underbrace{\int_0^\tau N(\xi)e^{-\xi s}d\xi}_{H(s)} = 0$$

If $|\rho q| > 1$: F has an infinite number of zeros in the RHP and H is proper. Thus D has an infinite number of zeros in the RHP

Open-loop analysis: $V(t) \equiv 0$

$$\beta(t,1) = \rho q \beta(t-\tau,1) - \int_0^\tau N(\xi)\beta(t-\xi,1)d\xi$$

Open loop analysis

If the open loop transfer function has an infinite number of poles in the RHP, the system cannot be delay-robustly stabilized.

Open-loop characteristic equation:

$$D(s) = \underbrace{1-\rho q e^{-\tau s}}_{F(s)} + \underbrace{\int_0^\tau N(\xi)e^{-\xi s}d\xi}_{H(s)} = 0$$

If $|\rho q| > 1$: F has an infinite number of zeros in the RHP and H is proper. Thus D has an infinite number of zeros in the RHP

$\Rightarrow$ Delay-robust stabilization is impossible if $|\rho q| > 1$.

Delay-robust state feedback

Delay-robust control law

$$V(t) = -\tilde{\rho} u(t,1) + \int_0^1 \left(K_{\tilde{\rho}}^{vu}(1,\xi) u(t,\xi) + K_{\tilde{\rho}}^{vv}(1,\xi) v(t,\xi) \right) d\xi$$

Delay-robust state feedback

Delay-robust control law

$$V(t) = -\tilde{\rho}u(t,1) + \int_0^1 \left(K_{\tilde{\rho}}{}^{vu}(1,\xi)u(t,\xi) + K_{\tilde{\rho}}{}^{vv}(1,\xi)v(t,\xi)\right) d\xi$$

Three different situations: trade-off performance/robustness

- If $|\rho q| \geq 1 \rightarrow$ delay-robust stabilization is impossible.
- If $1 > |\rho q| \geq \frac{1}{2} \rightarrow$ renounce to finite-time stabilization
- If $\frac{1}{2} > |\rho q| \rightarrow$ finite-time stabilization is possible.

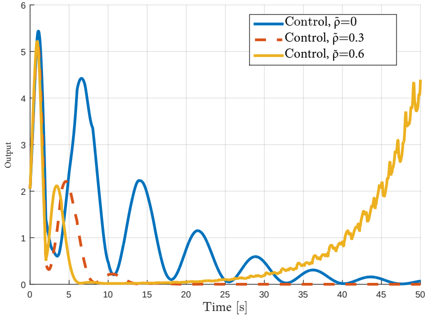
Delay-robust state feedback

Delay-robust control law

$$V(t) = -\tilde{\rho}u(t,1) + \int_0^1 \left(K_{\tilde{\rho}}^{vu}(1,\xi)u(t,\xi) + K_{\tilde{\rho}}^{vv}(1,\xi)v(t,\xi)\right) d\xi$$

Three different situations: trade-off performance/robustness

- If $|\rho q| \geq 1 \rightarrow$ delay-robust stabilization is impossible.
- If $1 > |\rho q| \geq \frac{1}{2} \rightarrow$ renounce to finite-time stabilization
- If $\frac{1}{2} > |\rho q| \rightarrow$ finite-time stabilization is possible.



Different robustness problems

$$
\begin{aligned}
&u_t(t,x)+\lambda u_x(t,x)=\sigma^{+-}v(t,x)\\
&v_t(t,x)-\mu v_x(t,x)=\sigma^{-+}u(t,x)\\
&u(t,0)=qv(t,0)\quad v(t,1)=\rho u(t,1)+V(t)
\end{aligned}
$$

Measurement: $y(t)=u(t,1)$.

Different robustness problems

$$u_t(t,x) + \lambda u_x(t,x) = \sigma^{+-} v(t,x)$$
$$v_t(t,x) - \mu v_x(t,x) = \sigma^{-+} u(t,x)$$
$$u(t,0) = qv(t,0) \quad v(t,1) = \rho u(t,1) + V(t - \delta_0)$$

Measurement: $y(t) = u(t,1)$.

- Delay acting on the actuation.

$$u_t(t,x) + \lambda u_x(t,x) = \sigma^{+-} v(t,x)$$
$$v_t(t,x) - \mu v_x(t,x) = \sigma^{-+} u(t,x)$$
$$u(t,0) = q v(t,0) \quad v(t,1) = \rho u(t,1) + V(t - \delta_0)$$

Measurement: $y(t) = u(t - \delta_1, 1)$.

- Delay acting on the actuation.
- Delay acting on the measurement.

Different robustness problems

$$
\begin{aligned}
&u_t(t,x) + \bar{\lambda} u_x(t,x) = \sigma^{+-} v(t,x) \\
&v_t(t,x) - \bar{\mu} v_x(t,x) = \sigma^{-+} u(t,x) \\
&u(t,0) = q v(t,0) \quad v(t,1) = \rho u(t,1) + V(t-\delta_0)
\end{aligned}
$$

Measurement: $y(t) = u(t-\delta_1, 1)$.

- Delay acting on the actuation.
- Delay acting on the measurement.
- Uncertainties on the transport velocities.

Different robustness problems

$$u_t(t,x) + \bar{\lambda} u_x(t,x) = \bar{\sigma}^{+-} v(t,x)$$
$$v_t(t,x) - \bar{\mu} v_x(t,x) = \bar{\sigma}^{-+} u(t,x)$$
$$u(t,0) = \bar{q} v(t,0) \quad v(t,1) = \bar{\rho} u(t,1) + (1+\delta_V) V(t-\delta_0)$$

Measurement: $y(t) = u(t-\delta_1, 1)$.

- Delay acting on the actuation.
- Delay acting on the measurement.
- Uncertainties on the transport velocities.
- Uncertainties on the coupling terms.
- Neglected dynamics...

Different robustness problems

$$u_t(t,x) + \lambda u_x(t,x) = \sigma^{+-} v(t,x)$$
$$v_t(t,x) - \mu v_x(t,x) = \sigma^{-+} u(t,x)$$
$$u(t,0) = qv(t,0) \quad v(t,1) = \rho u(t,1) + V(t)$$

Measurement: $y(t) = u(t,1)$.

- Backstepping method to design stabilizing control laws and **dual state-observers**.
- The **robustness** can be proved using the time-delay representation.
- Introduction of simple **degrees of freedom** in the design.
- **Trade-offs** performance-robustness, disturbance rejection-noise sensitivity.

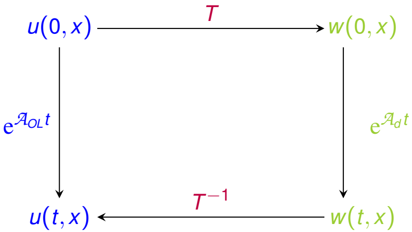
Backstepping methodology

Main idea Use an integral transform (classically Volterra transform of the *second kind*):

$$w(t,x) = u(t,x) - \int_0^x p(x,y)u(t,y)dy$$

to map the original system (to stabilize) to a stable target system .

$\longrightarrow$ Constructive design of control laws!



Limitations

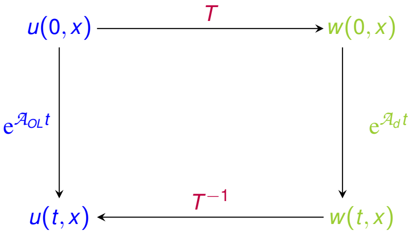
- Choice of an adequate target system
- **Proof of existence and invertibility of an adequate transform**
- Control effort, closed-loop properties

Backstepping methodology

Main idea Use an integral transform (classically Volterra transform of the *second kind*):

$$w(t,x) = u(t,x) - \int_0^x p(x,y)u(t,y)dy$$

to map the original system (to stabilize) to a stable target system .
$\longrightarrow$ Constructive design of control laws!



Limitations

- **Choice of an adequate target system**
- Proof of existence and invertibility of an adequate transform
- Control effort, **closed-loop properties** and implementation

What is a "good" target system?

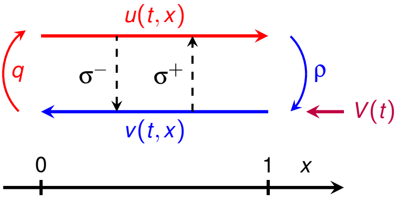
- Should at least be **exponentially stable!**
- If too **simple**: impossible to reach. If too **complex**: analysis is difficult

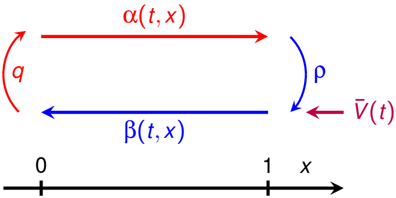
What is a "good" target system?

- Should at least be **exponentially stable!**
- If too **simple**: impossible to reach. If too **complex**: analysis is difficult

$$u_t(t,x)+\lambda u_x(t,x)=\sigma^+ v(t,x),$$
$$v_t(t,x)-\mu v_x(t,x)=\sigma^- u(t,x).$$

$$\alpha_t(t,x)+\lambda\alpha_x(t,x)=0,$$
$$\beta_t(t,x)-\mu\beta_x(t,x)=0.$$





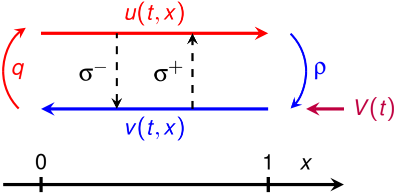
$$u(t,0)=qv(t,0)$$
$$v(t,1)=\rho u(t,1)+V(t)$$

$$\alpha(t,0)=q\beta(t,0)$$
$$\beta(t,1)=\rho\alpha(t,1)+\bar{V}(t)$$

What is a "good" target system?

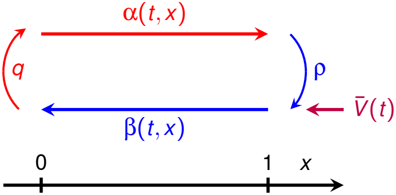
- Should at least be **exponentially stable!**
- If too **simple**: impossible to reach. If too **complex**: analysis is difficult

$$u_t(t,x) + \lambda u_x(t,x) = \sigma^+ v(t,x),$$
$$v_t(t,x) - \mu v_x(t,x) = \sigma^- u(t,x).$$



$$u(t,0) = qv(t,0)$$
$$v(t,1) = \rho u(t,1) + V(t)$$

$$\alpha_t(t,x) + \lambda \alpha_x(t,x) = 0,$$
$$\beta_t(t,x) - \mu \beta_x(t,x) = 0.$$



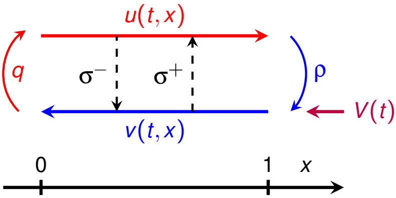
$$\alpha(t,0) = q\beta(t,0)$$
$$\beta(t,1) = \rho\alpha(t,1) + \bar{V}(t)$$

- Simple target system (no in-domain couplings).
- We may have removed stabilizing terms.

What is a "good" target system?

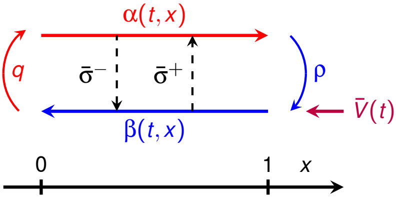
- Should at least be **exponentially stable!**
- If too **simple**: impossible to reach. If too **complex**: analysis is difficult

$$u_t(t,x)+\lambda u_x(t,x)=\sigma^+ v(t,x),$$
$$v_t(t,x)-\mu v_x(t,x)=\sigma^- u(t,x).$$



$$u(t,0)=qv(t,0)$$
$$v(t,1)=\rho u(t,1)+V(t)$$

$$\alpha_t(t,x)+\lambda\alpha_x(t,x)=\bar{\sigma}^+(x)\beta(t,x),$$
$$\beta_t(t,x)-\mu\beta_x(t,x)=\bar{\sigma}^-(x)\alpha(t,x).$$



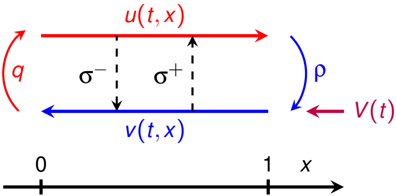
$$\alpha(t,0)=q\beta(t,0)$$
$$\beta(t,1)=\rho\alpha(t,1)+\bar{V}(t)$$

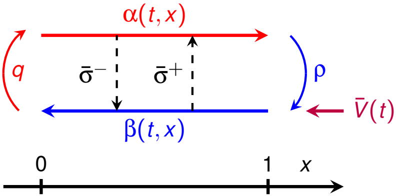
What is a "good" target system?

- Should at least be **exponentially stable!**
- If too **simple**: impossible to reach. If too **complex**: analysis is difficult

$$u_t(t,x) + \lambda u_x(t,x) = \sigma^+ v(t,x),$$
$$v_t(t,x) - \mu v_x(t,x) = \sigma^- u(t,x).$$

$$\alpha_t(t,x) + \lambda \alpha_x(t,x) = \bar{\sigma}^+(x)\beta(t,x),$$
$$\beta_t(t,x) - \mu \beta_x(t,x) = \bar{\sigma}^-(x)\alpha(t,x).$$





$$u(t,0) = qv(t,0)$$
$$v(t,1) = \rho u(t,1) + V(t)$$

$$\alpha(t,0) = q\beta(t,0)$$
$$\beta(t,1) = \rho\alpha(t,1) + \bar{V}(t)$$

- In-domain couplings.
- Smaller control effort?

What is a "good" target system?

- Should at least be **exponentially stable!**
- If too **simple**: impossible to reach. If too **complex**: analysis is difficult

Introduce d.o.f in the design to obtain a class of easily parametrizable target system?

Port-Hamiltonian Framework

- Context: Interaction of physical systems with environment $\leftrightarrow$ power flow through ports.
- Takes advantage of **physical properties** (passivity, dissipativity) of systems.
- Could be useful to introduce good target systems candidates.

Port-Hamiltonian Framework

- Context: Interaction of physical systems with environment $\leftrightarrow$ power flow through ports.
- Takes advantage of **physical properties** (passivity, dissipativity) of systems.
- Could be useful to introduce good target systems candidates.

Idea: combining PHS and backstepping

Take advantage of both methodologies to design boundary feedback laws:

- Use the Port-Hamiltonian framework to determine target systems with **physical meaning**
- Use the backstepping methodology to **map** the original system to the target system

Control objective

Design the control law $V(t)$ s.t closed-loop system is equivalent to a target system with **specified properties**.

PHS formulation of clamped string

System equations:

$$\begin{cases} \rho(x)\frac{\partial^2 w}{\partial t^2}(t,x) = \frac{\partial}{\partial x}\left(E(x)\frac{\partial w}{\partial x}(t,x)\right) - \kappa(x)\frac{\partial w}{\partial t}(t,x), \\ \frac{\partial w}{\partial t}|_{x=0}(t) = 0, \quad E(1)\frac{\partial w}{\partial x}|_{x=1}(t) = U(t), \end{cases}$$

Energy state variables:

$$\begin{cases} X_1(x,t) = \frac{\partial w}{\partial x}(x,t) & \text{: strain} \\ X_2(x,t) = \rho(x)\frac{\partial w}{\partial t}(x,t) & \text{: momentum density} \end{cases}$$

Port-Hamiltonian System

$$\frac{\partial}{\partial t}\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = \begin{pmatrix} 0 & \frac{\partial}{\partial x}\left(\frac{1}{\rho(x)}\cdot\right) \\ \frac{\partial}{\partial x}(E(x)\cdot) & -c(x) \end{pmatrix}\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \Leftrightarrow \frac{\partial X}{\partial t} = P_1\frac{\partial}{\partial x}(\mathcal{H}(x)X) + G_0(\mathcal{H}(x)X)$$

with *Hamiltonian density* $\mathcal{H}(x) = \begin{pmatrix} E(x) & 0 \\ 0 & \frac{1}{\rho(x)} \end{pmatrix}$, $c(x) = \frac{\kappa(x)}{\rho(x)}$, $P_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $G_0 = \begin{pmatrix} 0 & 0 \\ 0 & -c(x)\rho(x) \end{pmatrix}$.

System energy: $\mathcal{E}(t) = \frac{1}{2}\int_0^1 \left(E(x)X_1(t,x)^2 + \rho^{-1}(x)X_2(t,x)^2\right)\mathrm{d}x.$

Exchange through the actuated boundary in OL: $\frac{d\mathcal{E}}{dt}(t) = -\int_0^1 \left(\kappa(x)(\frac{X_2(t,x)}{\rho(x)})^2\right)\mathrm{d}x.$

Control objective

Change of internal power through actuation at the boundary $x = 1$:

- $\kappa > 0$: OL system **stable** $\mapsto$ fasten stabilization;
- else $\mapsto$ stabilize the string dynamics

} impose a specific decay rate to $\mathcal{E}$, using *distributed damping assignment*

Control Objective

Design control law $U(t)$ s.t dynamics of X equivalent to the dynamics of $\bar{X} = (\bar{X}_1, \bar{X}_2)$ satisfying

$$\frac{\partial}{\partial t}\begin{pmatrix}\bar{X}_1\\ \bar{X}_2\end{pmatrix} = \begin{pmatrix} 0 & \frac{\partial}{\partial x}\left(\frac{1}{\rho(x)}\cdot\right) \\ \frac{\partial}{\partial x}\left(E(x)\cdot\right) & -K \end{pmatrix}\begin{pmatrix}\bar{X}_1\\ \bar{X}_2\end{pmatrix},$$

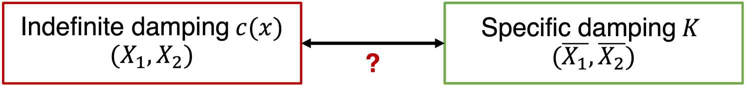
Class of target systems parametrized by K

In CL, energy decreases $\propto K$:

$$\frac{d\bar{\mathcal{E}}}{dt} = -K\int_0^1 \left(\frac{\bar{X}_2(x,t)}{\rho(x)}\right)^2 dx.$$

Control strategy

Overall objective: Determine an invertible transform $\mathcal{T}$ mapping initial PHS to target system.



Control strategy

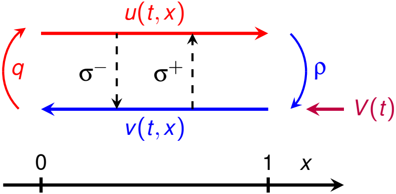
Overall objective: Determine an invertible transform $\mathcal{T}$ mapping initial PHS to target system.

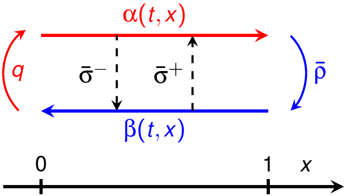
Indefinite damping $c(x)$ (X_1, X_2)	$\longleftrightarrow$?	Specific damping K $(\overline{X_1}, \overline{X_2})$

- Expression in the Riemann coordinates + Backstepping transformation

$$u_t(t,x) + \lambda u_x(t,x) = \sigma^+ v(t,x),$$
$$v_t(t,x) - \mu v_x(t,x) = \sigma^- u(t,x).$$

$$\alpha_t(t,x) + \lambda \alpha_x(t,x) = \bar{\sigma}^+(x)\beta(t,x),$$
$$\beta_t(t,x) - \mu \beta_x(t,x) = \bar{\sigma}^-(x)\alpha(t,x).$$





$$u(t,0) = qv(t,0)$$
$$v(t,1) = \rho u(t,1) + V(t)$$

$$\alpha(t,0) = q\beta(t,0)$$
$$\beta(t,1) = \bar{\rho}\alpha(t,1)$$

Backstepping transformations

Main idea

Use an invertible Volterra integral transform to replace in-domain coupling terms $\sigma^{\pm}(x)$ by *adequate* terms $\bar{\sigma}^{\pm}(x)$).

Volterra transform of the second kind

$$\begin{pmatrix} \alpha(t,x) \\ \beta(t,x) \end{pmatrix} = \begin{pmatrix} u(t,x) \\ v(t,x) \end{pmatrix} - \int_0^x \mathcal{K}(x,y) \begin{pmatrix} u(t,y) \\ v(t,y) \end{pmatrix} (y) dy$$

Kernel $\mathcal{K} = \left(\begin{smallmatrix} K^{++} & K^{+-} \\ K^{-+} & K^{--} \end{smallmatrix} \right)$ uniquely defined (kernel equations).

Control input

Control input $V(t)$ directly follows from the backstepping methodology

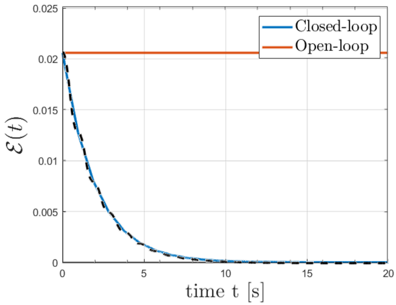
$$V(t) = (\bar{\rho} - \rho)u(t,1) + \int_0^1 (K^{-+}(1,y) - \bar{\rho}K^{++}(1,y))u(t,y) + (K^{--}(1,y) - \bar{\rho}K^{+-}(1,y))v(t,y)\mathrm{d}y.$$

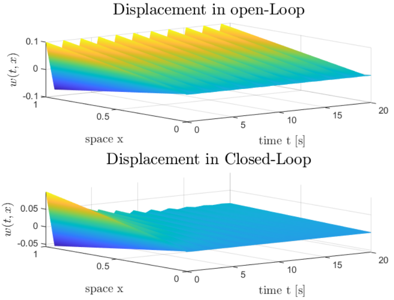
We obtain $U(t)$ for the initial system.

Simulation results

$$\begin{cases} \rho(x)\frac{\partial^2 w}{\partial t^2}(t,x) = \frac{\partial}{\partial x}\left(E(x)\frac{\partial w}{\partial x}(t,x)\right), \\ \frac{\partial w}{\partial t}|_{x=0}(t) = 0, \quad E(1)\frac{\partial w}{\partial x}|_{x=1}(t) = U(t), \end{cases}$$

Constant cœfficients $\rho = 936\text{kg.m}^{-3}$, $E = 4.14\text{GPa}$.





- PHS framework introduction of **degrees of freedom** in the design

Development of easily parametrizable target systems

- PHS framework introduction of **degrees of freedom** in the design

- Next steps:
  - Design of **analytical tools** to quantify the performance of the closed-loop system w.r.t a given set of specifications.
  - **Tuning methods** to use the available degrees of freedom best w.r.t this set of performance specifications.
  - **Toolbox** analogous to what exists for finite-dimensional systems.

- PHS framework introduction of **degrees of freedom** in the design
- Next steps:
  - Design of **analytical tools** to quantify the performance of the closed-loop system w.r.t a given set of specifications.
  - **Tuning methods** to use the available degrees of freedom best w.r.t this set of performance specifications.
  - **Toolbox** analogous to what exists for finite-dimensional systems.
- Advantages compared to simple PID controllers?

Late-lumping approximation

$$
\begin{aligned}
u_t(t,x) + \lambda u_x(t,x) &= \sigma^+ v(t,x), \\
v_t(t,x) - \mu v_x(t,x) &= \sigma^- u(t,x), \\
u(t,0) = qv(t,0) \quad v(t,1) &= \rho u(t,1) + V(t).
\end{aligned}
$$

Backstepping controller: $V(t) = \int_0^1 K(y)u(t,y) + L(1,y)v(t,y)\mathrm{d}y.$

- **Computational effort** related to the numerical implementation.

Late-lumping approximation

$$u_t(t,x)+\lambda u_x(t,x)=\sigma^+ v(t,x),$$
$$v_t(t,x)-\mu v_x(t,x)=\sigma^- u(t,x),$$
$$u(t,0)=qv(t,0) \quad v(t,1)=\rho u(t,1)+V(t).$$

Backstepping controller: $V(t)=\int_0^1 K(y)u(t,y)+L(1,y)v(t,y)\mathrm{d}y.$

- **Computational effort** related to the numerical implementation.
- Finite-dimensional approximation of the output-feedback controller: **model-reduction.**

Late-lumping approximation

$$
\begin{aligned}
&u_t(t,x)+\lambda u_x(t,x)=\sigma^+ v(t,x),\\
&v_t(t,x)-\mu v_x(t,x)=\sigma^- u(t,x),\\
&u(t,0)=qv(t,0) \quad v(t,1)=\rho u(t,1)+V(t).
\end{aligned}
$$

Backstepping controller: $V(t)=\int_0^1 K(y)u(t,y)+L(1,y)v(t,y)\mathrm{d}y.$

- **Computational effort** related to the numerical implementation.
- Finite-dimensional approximation of the output-feedback controller: **model-reduction.**
- **Late-lumping** controllers: guarantees of convergence? Advantages compared to early-lumping strategies?

Approximation of the control input

- V_n : **Approximation** of the control input $V(t)$

- Is the PDE system with approximated control input still stable?

$$
\begin{aligned}
&u_t(t,x) + \lambda u_x(t,x) = \sigma^+ v(t,x), \\
&v_t(t,x) - \mu v_x(t,x) = \sigma^- u(t,x), \\
&u(t,0) = q v(t,0) \quad v(t,1) = \rho u(t,1) + V_n(t).
\end{aligned}
$$

Approximation of the control input

- V_n : **Approximation** of the control input $V(t)$
- Is the PDE system with approximated control input still stable?

$$
\begin{aligned}
&u_t(t,x)+\lambda u_x(t,x)=\sigma^+ v(t,x),\\
&v_t(t,x)-\mu v_x(t,x)=\sigma^- u(t,x),\\
&u(t,0)=qv(t,0) \quad v(t,1)=\rho u(t,1)+V_n(t).
\end{aligned}
$$

- Same backstepping transformation

$$
\begin{aligned}
&\alpha_t(t,x)+\lambda\alpha_x(t,x)=0,\\
&\beta_t(t,x)-\mu\beta_x(t,x)=0,\\
&\alpha(t,0)=q\beta(t,0) \quad \beta(t,1)=\rho\alpha(t,1)+V_n(t)-V(t).
\end{aligned}
$$

Approximation of the control input

- V_n : **Approximation** of the control input $V(t)$

- Is the PDE system with approximated control input still stable?

$$
\begin{aligned}
&u_t(t,x) + \lambda u_x(t,x) = \sigma^+ v(t,x), \\
&v_t(t,x) - \mu v_x(t,x) = \sigma^- u(t,x), \\
&u(t,0) = qv(t,0) \quad v(t,1) = \rho u(t,1) + V_n(t).
\end{aligned}
$$

- Same backstepping transformation

$$
\begin{aligned}
&\alpha_t(t,x) + \lambda \alpha_x(t,x) = 0, \\
&\beta_t(t,x) - \mu \beta_x(t,x) = 0, \\
&\alpha(t,0) = q\beta(t,0) \quad \beta(t,1) = \rho\alpha(t,1) + V_n(t) - V(t).
\end{aligned}
$$

- Stability analysis using a **Lyapunov function**.

$$\alpha_t(t,x) + \lambda\alpha_x(t,x) = 0,$$
$$\beta_t(t,x) - \mu\beta_x(t,x) = 0,$$
$$\alpha(t,0) = q\beta(t,0) \quad \beta(t,1) = \rho\alpha(t,1) + V_n(t) - V(t).$$

Lyapunov function: $W(t) = \int_0^1 \frac{e^{-\nu x}}{\lambda}\alpha^2(t,x) + \frac{q^2 e^{\nu x}}{\mu}\beta^2(t,x)dx$: equivalent to L^2-norm ($\nu > 0$).

Lyapunov analysis

$$\alpha_t(t,x) + \lambda\alpha_x(t,x) = 0,$$
$$\beta_t(t,x) - \mu\beta_x(t,x) = 0,$$
$$\alpha(t,0) = q\beta(t,0) \quad \beta(t,1) = \rho\alpha(t,1) + V_n(t) - V(t).$$

Lyapunov function: $W(t) = \int_0^1 \frac{e^{-\nu x}}{\lambda}\alpha^2(t,x) + \frac{q^2 e^{\nu x}}{\mu}\beta^2(t,x)dx$: equivalent to L^2-norm ($\nu > 0$).

- Differentiation w.r.t time

$$\dot{W}(t) \leq -\nu_1 W(t) + C(V_n(t) - V(t))^2.$$

$$\alpha_t(t,x)+\lambda\alpha_x(t,x)=0,$$
$$\beta_t(t,x)-\mu\beta_x(t,x)=0,$$
$$\alpha(t,0)=q\beta(t,0) \quad \beta(t,1)=\rho\alpha(t,1)+V_n(t)-V(t).$$

Lyapunov function: $W(t)=\int_0^1 \frac{e^{-\nu x}}{\lambda}\alpha^2(t,x)+\frac{q^2e^{\nu x}}{\mu}\beta^2(t,x)dx$: equivalent to L^2-norm ($\nu>0$).

- Differentiation w.r.t time

$$\dot{W}(t)\leq -\nu_1 W(t)+C(V_n(t)-V(t))^2.$$

- Stability if **uniform convergence** of the approximation scheme

$$|V_n(t)-V(t)|\leq C_n||(u,v)||_{L^2}, \quad \text{where } C_n\to 0$$
$$\Rightarrow \quad \dot{W}(t)\leq -\eta W(t), \quad \text{for } n \text{ large enough.}$$

Lyapunov analysis

$$\alpha_t(t,x) + \lambda\alpha_x(t,x) = 0,$$
$$\beta_t(t,x) - \mu\beta_x(t,x) = 0,$$
$$\alpha(t,0) = q\beta(t,0) \quad \beta(t,1) = \rho\alpha(t,1) + V_n(t) - V(t).$$

Lyapunov function: $W(t) = \int_0^1 \frac{e^{-\nu x}}{\lambda}\alpha^2(t,x) + \frac{q^2 e^{\nu x}}{\mu}\beta^2(t,x)dx$: equivalent to L^2-norm ($\nu > 0$).

- Differentiation w.r.t time

$$\dot{W}(t) \leq -\nu_1 W(t) + C(V_n(t) - V(t))^2.$$

- Stability if **uniform convergence** of the approximation scheme

$$|V_n(t) - V(t)| \leq C_n ||(u,v)||_{L^2}, \quad \text{where } C_n \to 0$$
$$\Rightarrow \quad \dot{W}(t) \leq -\eta W(t), \quad \text{for } n \text{ large enough.}$$

- Approximations schemes: Galerkin approximation, machine-learning (DeepONet)
$\to$ recent publications, comparisons on test-case studies, no general results.

- **Backstepping controllers fo hyperbolic system**
  - Volterra change of coordinates $\rightarrow$ **target system** with a simpler structure.
  - **Robustness analysis** using a time-delay representation.
  - Extensions to **networks** of hyperbolic systems.

Conclusions and Perspectives

- **Backstepping controllers fo hyperbolic system**
  - Volterra change of coordinates $\rightarrow$ **target system** with a simpler structure.
  - **Robustness analysis** using a time-delay representation.
  - Extensions to **networks** of hyperbolic systems.

- **Development of easily parametrizable target systems**
  - PHS framework to introduce physical **degrees of freedom**.
  - Developments of **analytical tools** to quantify closed-loop performance.
  - Tuning methods to use the available dof best w.r.t a set of **performance specifications**.

Conclusions and Perspectives

- **Backstepping controllers fo hyperbolic system**
  - Volterra change of coordinates $\rightarrow$ **target system** with a simpler structure.
  - **Robustness analysis** using a time-delay representation.
  - Extensions to **networks** of hyperbolic systems.

- **Development of easily parametrizable target systems**
  - PHS framework to introduce physical **degrees of freedom**.
  - Developments of **analytical tools** to quantify closed-loop performance.
  - Tuning methods to use the available dof best w.r.t a set of **performance specifications**.

- **Approximation of the control law to reduce the numerical burden**
  - **Model reduction** strategies: Galerkin approximation, Machine-learning emulations.
  - Stability guarantees using **Lyapunov analysis**.
  - Necessary to obtain Lyapunov function for general hyperbolic system or TDS representation.

$$z(t) = \sum_{k=1}^{N} A_k z(t-\tau_k) + \int_0^{\tau} f(\nu) z(t-\nu) d\nu.$$

Conclusions and Perspectives

- **Backstepping controllers fo hyperbolic system**
  - Volterra change of coordinates $\rightarrow$ **target system** with a simpler structure.
  - **Robustness analysis** using a time-delay representation.
  - Extensions to **networks** of hyperbolic systems.

- **Development of easily parametrizable target systems**
  - PHS framework to introduce physical **degrees of freedom**.
  - Developments of **analytical tools** to quantify closed-loop performance.
  - Tuning methods to use the available dof best w.r.t a set of **performance specifications**.

- **Approximation of the control law to reduce the numerical burden**
  - **Model reduction** strategies: Galerkin approximation, Machine-learning emulations.
  - Stability guarantees using **Lyapunov analysis**.
  - Necessary to obtain Lyapunov function for general hyperbolic system or TDS representation.

$$z(t) = \sum_{k=1}^{N} A_k z(t-\tau_k) + \int_0^{\tau} f(\nu) z(t-\nu) d\nu.$$

- **Benchmark and experimental validation**.