Backstepping Control of PDEs:

Foundations, Recent Results and Open Problems

Rafael Vazquez (Univ. Sevilla, Spain)

Miroslav Krstic (Univ. California San Diego, USA)

Florent Di Meglio (MINES ParisTech, France)

Long Hu (Shandong University, China)

Recife Workshop on Control & Stabilization of PDEs

Recife, February 13-17, 2017

Backstepping for PDEs

In ODEs, a particular approach to stabilization of dynamic systems with “triangular” structure.

Wildly successful in the area of ODE nonlinear control.

Backstepping for PDEs

In ODEs, a particular approach to stabilization of dynamic systems with “triangular” structure.

Wildly successful in the area of ODE nonlinear control.

For PDEs, roughly speaking, backstepping is a *constructive* method that achieves Lyapunov stabilization by **transforming** the system into a stable “target system,” which is often achieved by collectively shifting all the eigenvalues in a favorable direction in the complex plane, rather than by assigning individual eigenvalues.

Backstepping allows this task can be achieved in a rather elegant way where the control gains are easy to compute, symbolically, numerically, or even explicitly.

Backstepping for PDEs

1. Identify the undesirable terms in the PDE.

Backstepping for PDEs

1. Identify the undesirable terms in the PDE.

2. Choose a target system in which the undesirable terms are to be eliminated by state transformation and feedback, as in feedback linearization.

Backstepping for PDEs

1. Identify the <u>undesirable terms in the PDE</u>.

2. Choose a <u>target system</u> in which the undesirable terms are to be eliminated by state transformation and feedback, as in feedback linearization.

3. Find the <u>state transformation</u> typically as *identity minus a Volterra operator* (in x).
   Volterra operator = integral operator from 0 up to x (rather than from 0 to 1).
   A Volterra transformation is "triangular" or "spatially causal."

Backstepping for PDEs

1. Identify the undesirable terms in the PDE.

2. Choose a target system in which the undesirable terms are to be eliminated by state transformation and feedback, as in feedback linearization.

3. Find the state transformation typically as *identity minus a Volterra operator* (in x).
   Volterra operator = integral operator from 0 up to x (rather than from 0 to 1).
   A Volterra transformation is "triangular" or "spatially causal."

4. Obtain boundary feedback from the Volterra transformation. The transformation alone cannot eliminate the undesirable terms, but the transformantion brings them to the boundary, so control can cancel them.

Backstepping for PDEs

Gain fcn of boundary controller = kernel of Volterra transformation.

Volterra kernel satisfies a *linear* PDE.

Backstepping for PDEs

Gain fcn of boundary controller = kernel of Volterra transformation.

Volterra kernel satisfies a *linear* PDE.

Backstepping is not “one-size-fits-all.” Requires structure-specific effort by designer.

Reward: elegant controller, clear closed-loop behavior.

Outline

- Foundations of backstepping: basic design for a 1D parabolic equation
- Coupled hyperbolic systems
- Coupled parabolic systems
- Extension to n-balls
- Bilateral design
- Some open problems

Backstepping for PDEs—An example

Start with one of the simplest unstable PDEs, a (constant-coefficient) reaction-diffusion equation:

$$\begin{aligned} u_t(x,t) &= u_{xx}(x,t) + \boxed{\lambda u(x,t)} && (1) \\ u(0,t) &= 0 && (2) \\ u(1,t) &= U(t) = \text{control} && (3) \end{aligned}$$

The open-loop system (1), (2) (with $u(1,t) = 0$) is unstable with arbitrarily many unstable eigenvalues for sufficiently large $\lambda > 0$.

Since the term λu is the source of instability, the natural objective for a boundary feedback is to “eliminate” this term.

Backstepping solution presented in Smyshlyaev & Krstic, IEEE TAC 2004

Backstepping for PDEs—An example

Target system (exp. stable)

$$
\begin{aligned}
w_t(x,t) &= w_{xx}(x,t) && (4)\\
w(0,t) &= 0 && (5)\\
w(1,t) &= 0 && (6)
\end{aligned}
$$

State transformation

$$\boxed{w(x,t) = u(x,t) - \int_0^x k(x,y)u(y,t)\,dy} \tag{7}$$

Feedback control

$$u(1,t) = \int_0^1 k(1,y)u(y,t)\,dy \tag{8}$$

Task: find kernel $k(x,y)$.

Backstepping for PDEs—An example

Task: find the function $k(x,y)$ (which we call "gain kernel") that makes the plant (1), (2) with the controller (8) equivalent to the target system (4)–(6).

We introduce the following notation:

$$
\begin{aligned}
k_x(x,x) &= \frac{\partial}{\partial x} k(x,y)|_{y=x} \\
k_y(x,x) &= \frac{\partial}{\partial y} k(x,y)|_{y=x} \\
\frac{d}{dx} k(x,x) &= k_x(x,x) + k_y(x,x).
\end{aligned}
$$

Backstepping for PDEs—An example

Differentiating the transformation (7) with respect to x gives

$$
\begin{aligned}
w_x(x) &= u_x(x) - k(x,x)u(x) - \int_0^x k_x(x,y)u(y)\,dy \\
w_{xx}(x) &= u_{xx}(x) - \frac{d}{dx}(k(x,x)u(x)) - k_x(x,x)u(x) - \int_0^x k_{xx}(x,y)u(y)\,dy \\
&= u_{xx}(x) - u(x)\frac{d}{dx}k(x,x) - k(x,x)u_x(x) - k_x(x,x)u(x) \\
&\quad - \int_0^x k_{xx}(x,y)u(y)\,dy\,.
\end{aligned} \tag{9}
$$

Backstepping for PDEs—An example

Next, we differentiate the transformation (7) with respect to time:

$$
\begin{aligned}
w_t(x) &= u_t(x) - \int_0^x k(x,y)u_t(y)dy \\
&= u_{xx}(x) + \lambda u(x) - \int_0^x k(x,y)\left(u_{yy}(y) + \lambda u(y)\right)\,dy \\
&= u_{xx}(x) + \lambda u(x) - k(x,x)u_x(x) + k(x,0)u_x(0) \\
&\quad + \int_0^x k_y(x,y)u_y(y)\,dy - \int_0^x \lambda k(x,y)u(y)\,dy \quad \text{(integration by parts)} \\
&= u_{xx}(x) + \lambda u(x) - k(x,x)u_x(x) + k(x,0)u_x(0) + k_y(x,x)u(x) - k_y(x,0)u(0) \\
&\quad - \int_0^x k_{yy}(x,y)u(y)\,dy - \int_0^x \lambda k(x,y)u(y)\,dy. \quad \text{(integration by parts)} \qquad (10)
\end{aligned}
$$

Subtracting (9) from (10), we get

$$
\boxed{
\begin{aligned}
w_t - w_{xx} &= \left[\lambda + 2\frac{d}{dx}k(x,x)\right]u(x) + k(x,0)u_x(0) \\
&\quad + \int_0^x \left(k_{xx}(x,y) - k_{yy}(x,y) - \lambda k(x,y)\right)u(y)\,dy \\
&= 0
\end{aligned}
}
$$

Backstepping for PDEs—An example

For this to hold for all u, three conditions have to be satisfied:

$$\begin{aligned} k_{xx}(x,y) - k_{yy}(x,y) - \lambda k(x,y) &= 0 && (11) \\ k(x,0) &= 0 && (12) \\ \lambda + 2\frac{d}{dx}k(x,x) &= 0. && (13) \end{aligned}$$

We simplify (13) by integrating it with respect to x and noting from (12) that $k(0,0) = 0$, which gives us

$$\boxed{\begin{aligned} k_{xx}(x,y) - k_{yy}(x,y) &= \lambda k(x,y) \\ k(x,0) &= 0 \\ k(x,x) &= -\frac{\lambda}{2}x \end{aligned}} \qquad (14)$$

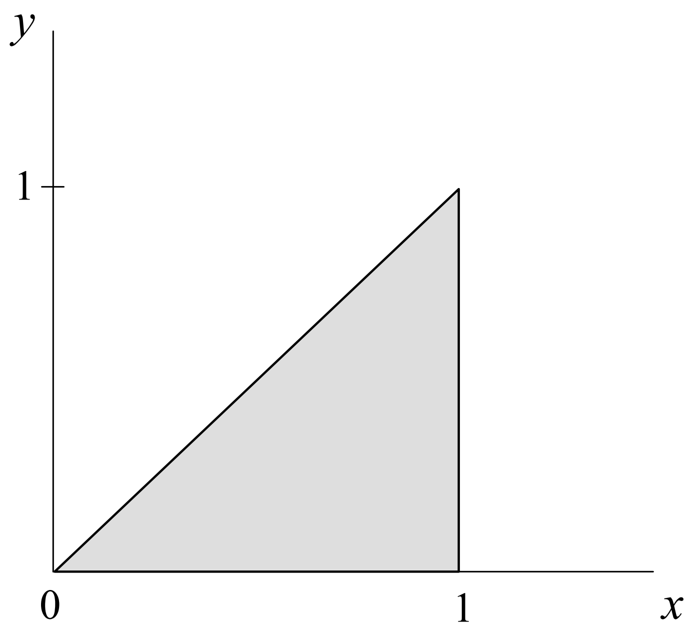
Backstepping for PDEs—An example

These three conditions form a well posed PDE of hyperbolic type in the “Goursat form.”

One can think of the k-PDE as a wave equation with an extra term λk.

x plays the role of time and y of space.

In quantum physics such PDEs are called Klein-Gordon PDEs.



Domain of the PDE for gain kernel $k(x, y)$.

The boundary conditions are prescribed on hypotenuse and the lower cathetus of the triangle.

The value of $k(x, y)$ on the vertical cathetus gives us the control gain $k(1, y)$.

Backstepping for PDEs—An example

To find a solution of the k-PDE (14) we first convert it into an integral equation.

Introducing the change of variables

$$\boxed{\xi = x + y, \qquad \eta = x - y} \tag{15}$$

we have

$$\begin{aligned} k(x,y) &= G(\xi,\eta) \\ k_x &= G_\xi + G_\eta \\ k_{xx} &= G_{\xi\xi} + 2G_{\xi\eta} + G_{\eta\eta} \\ k_y &= G_\xi - G_\eta \\ k_{yy} &= G_{\xi\xi} - 2G_{\xi\eta} + G_{\eta\eta}\,. \end{aligned}$$

Backstepping for PDEs—An example

Thus, the gain kernel PDE becomes

$$\begin{aligned} G_{\xi\eta}(\xi,\eta) &= \frac{\lambda}{4}G(\xi,\eta) && (16)\\ G(\xi,\xi) &= 0 && (17)\\ G(\xi,0) &= -\frac{\lambda}{4}\xi\,. && (18)\end{aligned}$$

Integrating (16) with respect to η from 0 to η, we get

$$G_\xi(\xi,\eta) = G_\xi(\xi,0) + \int_0^\eta \frac{\lambda}{4}G(\xi,s)\,ds = -\frac{\lambda}{4} + \int_0^\eta \frac{\lambda}{4}G(\xi,s)\,ds\,. \qquad (19)$$

Next, we integrate (19) with respect to ξ from η to ξ to get the integral equation

$$\boxed{G(\xi,\eta) = -\frac{\lambda}{4}(\xi-\eta) + \frac{\lambda}{4}\int_\eta^\xi \int_0^\eta G(\tau,s)\,ds\,d\tau} \qquad (20)$$

The G-integral eqn is easier to analyze than the k-PDE.

Backstepping for PDEs—An example

Start with an initial guess

$$G^0(\xi,\eta) = 0 \tag{21}$$

and set up the recursive formula for (20) as follows:

$$\boxed{G^{n+1}(\xi,\eta) = -\frac{\lambda}{4}(\xi-\eta) + \frac{\lambda}{4}\int_\eta^\xi \int_0^\eta G^n(\tau,s)\,ds\,d\tau} \tag{22}$$

If this functional iteration converges, we can write the solution $G(\xi,\eta)$ as

$$G(\xi,\eta) = \lim_{n\to\infty} G^n(\xi,\eta)\,. \tag{23}$$

Backstepping for PDEs—An example

Let us denote the difference between two consecutive terms as

$$\Delta G^n(\xi,\eta) = G^{n+1}(\xi,\eta) - G^n(\xi,\eta)\,. \tag{24}$$

Then

$$\Delta G^{n+1}(\xi,\eta) = \frac{\lambda}{4}\int_\eta^\xi \int_0^\eta \Delta G^n(\tau,s)\,ds\,d\tau \tag{25}$$

and (23) can be alternatively written as

$$G(\xi,\eta) = \sum_{n=0}^{\infty} \Delta G^n(\xi,\eta)\,. \tag{26}$$

Computing ΔG^n from (25) starting with

$$\Delta G^0 = G^1(\xi,\eta) = -\frac{\lambda}{4}(\xi-\eta)\,, \tag{27}$$

we can observe the pattern which leads to the following formula:

$$\boxed{\Delta G^n(\xi,\eta) = -\frac{(\xi-\eta)\xi^n\eta^n}{n!(n+1)!}\left(\frac{\lambda}{4}\right)^{n+1}} \tag{28}$$

This formula can be verified by induction.

Backstepping for PDEs—An example

The solution to the integral equation is given by

$$G(\xi,\eta) = -\sum_{n=0}^{\infty} \frac{(\xi-\eta)\xi^n\eta^n}{n!(n+1)!}\left(\frac{\lambda}{4}\right)^{n+1}. \tag{29}$$

To compute the series (29), note that a first order modified Bessel function of the first kind can be represented as

$$I_1(x) = \sum_{n=0}^{\infty} \frac{(x/2)^{2n+1}}{n!(n+1)!}. \tag{30}$$

Backstepping for PDEs—An example

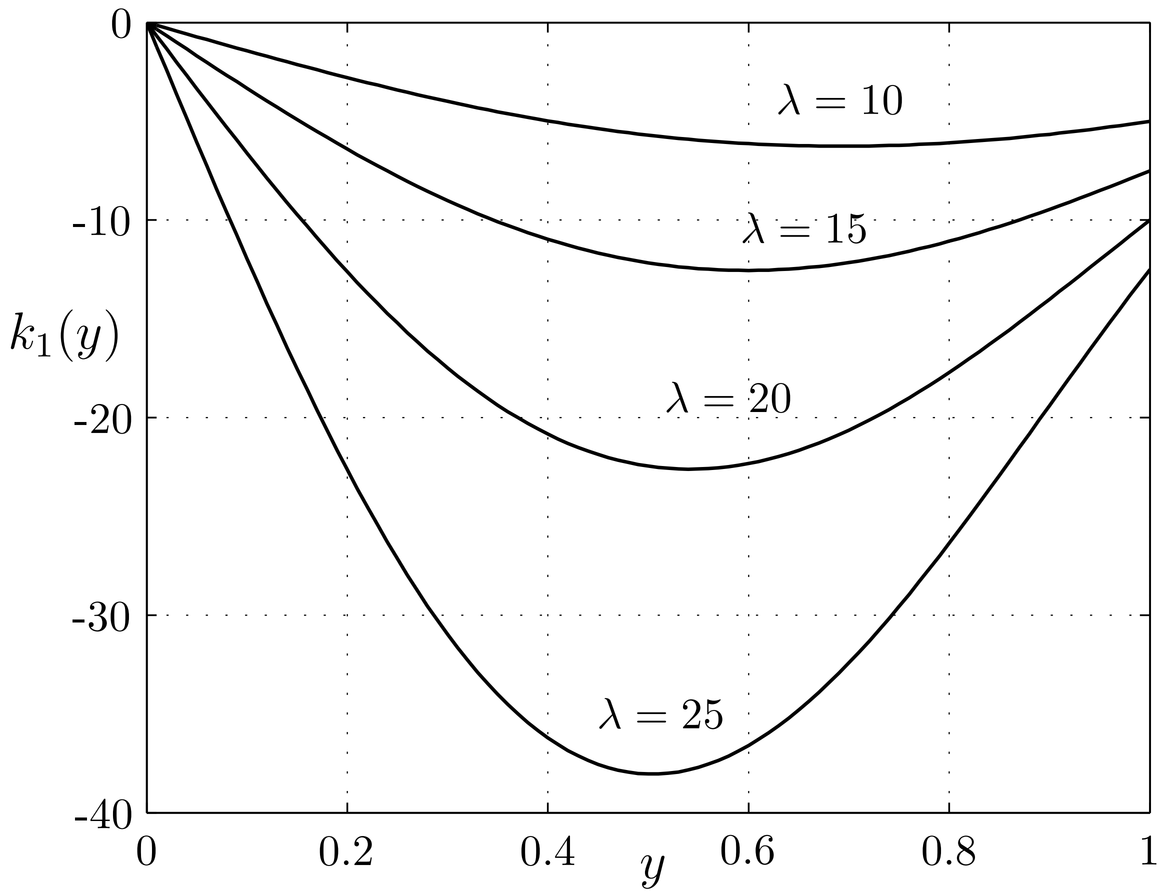
Comparing (30) with (29) we obtain

$$G(\xi,\eta) = -\frac{\lambda}{2}(\xi-\eta)\frac{I_1(\sqrt{\lambda\xi\eta})}{\sqrt{\lambda\xi\eta}} \tag{31}$$

or, returning to the original x, y variables,

$$\boxed{k(x,y) = -\lambda y \frac{I_1\left(\sqrt{\lambda(x^2-y^2)}\right)}{\sqrt{\lambda(x^2-y^2)}}} \tag{32}$$

Backstepping for PDEs—An example



Control gain $k(1, y)$ for different values of λ

As λ gets larger, the plant becomes more unstable which requires more control effort.

Low gain near the boundaries is logical: near $x = 0$ the state is small even without control because of the boundary condition $u(0) = 0$; near $x = 1$ the control has the most impact.

Backstepping for PDEs—An example

We need to establish that stability of the w-target system (4)–(6) implies stability of the u-closed-loop system (1), (2), (8), by showing that the transformation $u \mapsto w$ is invertible.

Invertibility is obvious by seeing the backstepping transformation as an integral equation in u.

Postulate an inverse transformation in the form

$$u(x) = w(x) + \int_0^x l(x,y)w(y)\,dy\,, \tag{33}$$

where $l(x,y)$ is the transformation kernel.

Given the direct transformation (7) and the inverse transformation (33), the kernels $k(x,y)$ and $l(x,y)$ satisfy

$$\boxed{l(x,y) = k(x,y) + \int_y^x k(x,\xi)l(\xi,y)\,d\xi} \tag{34}$$

Backstepping for PDEs—An example

One can find also kernel equations for $l(x,y)$:

$$\boxed{\begin{aligned} l_{xx}(x,y) - l_{yy}(x,y) &= -\lambda l(x,y) \\ l(x,0) &= 0 \\ l(x,x) &= -\frac{\lambda}{2}x \end{aligned}} \tag{35}$$

Comparing this PDE with the PDE (14) for $k(x,y)$, we see that

$$l(x,y;\lambda) = -k(x,y;-\lambda)\,. \tag{36}$$

Backstepping for PDEs—An example

From (32) we have

$$l(x,y) \quad = \quad -\lambda y \frac{I_1\left(\sqrt{-\lambda(x^2-y^2)}\right)}{\sqrt{-\lambda(x^2-y^2)}} = -\lambda y \frac{I_1\left(j\sqrt{\lambda(x^2-y^2)}\right)}{j\sqrt{\lambda(x^2-y^2)}},$$

or, using the properties of I_1,

$$\boxed{l(x,y) = -\lambda y \frac{J_1\left(\sqrt{\lambda(x^2-y^2)}\right)}{\sqrt{\lambda(x^2-y^2)}}} \tag{37}$$

Summary of control design for the reaction-diffusion equation

Plant
$$u_t = u_{xx} + \lambda u \quad (38)$$
$$u(0) = 0 \quad (39)$$

Controller
$$u(1) = -\int_0^1 y\lambda \frac{I_1\left(\sqrt{\lambda(1-y^2)}\right)}{\sqrt{\lambda(1-y^2)}} u(y)\, dy \quad (40)$$

Transformation
$$w(x) = u(x) + \int_0^x \lambda y \frac{I_1\left(\sqrt{\lambda(x^2-y^2)}\right)}{\sqrt{\lambda(x^2-y^2)}} u(y)\, dy \quad (41)$$

$$u(x) = w(x) - \int_0^x \lambda y \frac{J_1\left(\sqrt{\lambda(x^2-y^2)}\right)}{\sqrt{\lambda(x^2-y^2)}} w(y)\, dy \quad (42)$$

Target system
$$w_t = w_{xx} \quad (43)$$
$$w(0) = 0 \quad (44)$$
$$w(1) = 0 \quad (45)$$

Outline

- Foundations of backstepping: basic design for a 1D parabolic equation
- Coupled hyperbolic systems
- Coupled parabolic systems
- Extension to n-balls
- Bilateral design
- Some open problems

Backstepping control of coupled hyperbolic 1-D systems

$$u_t(t,x) + \Sigma^+ u_x(t,x) = \Lambda^{++} u(t,x) + \Lambda^{+-} v(t,x)$$
$$v_t(t,x) - \Sigma^- v_x(t,x) = \Lambda^{-+} u(t,x) + \Lambda^{--} v(t,x)$$

with the following boundary conditions

$$u(t,0) = 0, \qquad v(t,1) = U(t)$$

where

$$u = \begin{pmatrix} u_1 & \cdots & u_n \end{pmatrix}^T, \qquad v = \begin{pmatrix} v_1 & \cdots & v_m \end{pmatrix}^T$$

$$\Sigma^+ = \begin{pmatrix} \varepsilon_1 & & 0 \\ & \ddots & \\ 0 & & \varepsilon_n \end{pmatrix}, \qquad \Sigma^- = \begin{pmatrix} \mu_1 & & 0 \\ & \ddots & \\ 0 & & \mu_m \end{pmatrix}$$

with

$$-\mu_1 < \cdots < -\mu_m < 0 < \varepsilon_1 \leq \cdots \leq \varepsilon_n$$

Backstepping control of coupled hyperbolic 1-D systems

Backstepping transformation

$$\alpha(t,x) = u(t,x)$$
$$\beta(t,x) = v(t,x) - \int_0^x [L(x,\xi)u(\xi) + K(x,\xi)v(\xi)]\, d\xi$$

L and K defined on the triangular domain $\mathcal{T}$.

Target system

$$\alpha_t(t,x) + \Sigma^+\alpha_x(t,x) = \Lambda^{++}\alpha(t,x) + \Lambda^{+-}\beta(t,x) + \int_0^x D^+(x,\xi)\alpha(\xi)d\xi + \int_0^x D^-(x,\xi)\beta(\xi)d\xi$$
$$\beta_t(t,x) - \Sigma^-\beta_x(t,x) = G(x)\beta(0)$$

with boundary conditions

$$\alpha(t,0) = \beta(t,1) = 0$$

Backstepping control of coupled hyperbolic 1-D systems

Structure of G is **lower-diagonal with diagonal of zeros**

$$G(x) = \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ g_{2,1}(x) & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ g_{m,1}(x) & \cdots & g_{m,m-1}(x) & 0 \end{pmatrix}$$

It can be shown to make stable

$$\beta_t(t,x) - \Sigma^- \beta_x(t,x) = G(x)\beta(0)$$

From there follows target system stability.

Backstepping control of coupled hyperbolic 1-D systems

Structure of G is **lower-diagonal with diagonal of zeros**

$$G(x) = \begin{pmatrix} 0 & \cdots & \cdots & 0 \\ g_{2,1}(x) & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ g_{m,1}(x) & \cdots & g_{m,m-1}(x) & 0 \end{pmatrix}$$

It can be shown to make stable

$$\beta_t(t,x) - \Sigma^- \beta_x(t,x) = G(x)\beta(0)$$

From there follows target system stability.

G(x) is not chosen, but computed from the kernels.

Backstepping control of coupled hyperbolic 1-D systems

Kernel equations

$$
\begin{aligned}
0 =& \Sigma^- L_x(x,\xi) - L_\xi(x,\xi)\Sigma^+ - L(x,\xi)\Lambda^{++} - K(x,\xi)\Lambda^{-+} \\
0 =& \Sigma^- K_x(x,\xi) + K_\xi(x,\xi)\Sigma^- - K(x,\xi)\Lambda^{--} - L(x,\xi)\Lambda^{+-}
\end{aligned}
$$

with boundary conditions

$$
\begin{aligned}
0 =& L(x,x)\Sigma^+ + \Sigma^- L(x,x) + \Lambda^{-+} \\
0 =& \Sigma^- K(x,x) - K(x,x)\Sigma^- + \Lambda^{--} \\
0 =& G(x) - K(x,0)\Sigma^-
\end{aligned}
$$

Too many boundary conditions?

Backstepping control of coupled hyperbolic 1-D systems

Kernel equations

$$
\begin{aligned}
0 =& \Sigma^{-} L_x(x,\xi) - L_\xi(x,\xi)\Sigma^{+} - L(x,\xi)\Lambda^{++} - K(x,\xi)\Lambda^{-+} \\
0 =& \Sigma^{-} K_x(x,\xi) + K_\xi(x,\xi)\Sigma^{-} - K(x,\xi)\Lambda^{--} - L(x,\xi)\Lambda^{+-}
\end{aligned}
$$

with boundary conditions

$$
\begin{aligned}
0 =& L(x,x)\Sigma^{+} + \Sigma^{-} L(x,x) + \Lambda^{-+} \\
0 =& \Sigma^{-} K(x,x) - K(x,x)\Sigma^{-} + \Lambda^{--} \\
0 =& G(x) - K(x,0)\Sigma^{-}
\end{aligned}
$$

Again, too many boundary conditions?

No, in fact more boundary conditions are needed $\longrightarrow$ Nonuniqueness!

Backstepping control of coupled hyperbolic 1-D systems

Developing the equations:

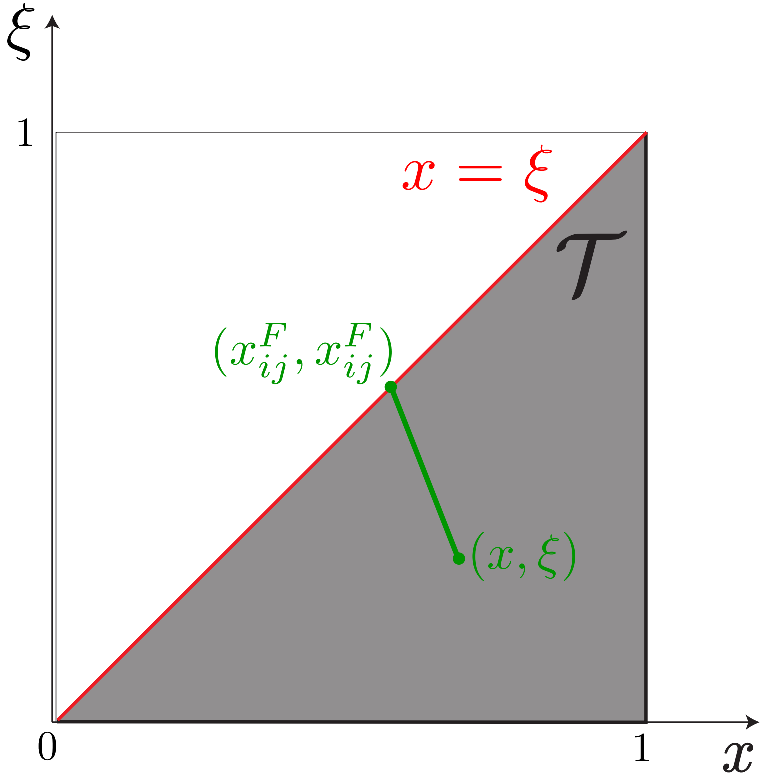
$$\begin{aligned}
\mu_i \partial_x L_{ij}(x,\xi) - \varepsilon_j \partial_\xi L_{ij}(x,\xi) &= \sum_{k=1}^{n} \lambda_{kj}^{++} L_{ik}(x,\xi) + \sum_{p=1}^{m} \lambda_{pj}^{-+} K_{ip}(x,\xi) \\
\mu_i \partial_x K_{ij}(x,\xi) + \mu_j \partial_\xi K_{ij}(x,\xi) &= \sum_{p=1}^{m} \lambda_{pj}^{--} K_{ip}(x,\xi) + \sum_{k=1}^{n} \lambda_{kj}^{+-} L_{ik}(x,\xi)
\end{aligned}$$

with boundary conditions:

$$\begin{aligned}
\forall 1 \le i \le m, 1 \le j \le n, &\quad L_{ij}(x,x) = -\frac{\lambda_{ij}^{-+}}{\mu_i + \varepsilon_j} \\
\forall 1 \le i, j \le m, i \ne j, &\quad K_{ij}(x,x) = -\frac{\lambda_{ij}^{--}}{\mu_i - \mu_j} \\
\forall 1 \le i \le j \le m, &\quad K_{ij}(x,0) = 0 \\
\forall 1 \le j < i \le m, &\quad K_{ij}(1,\xi) = l_{ij} \\
\forall 1 \le j < i \le m, &\quad g_{ij}(x) = \mu_j K_{ij}(x,0)
\end{aligned}$$

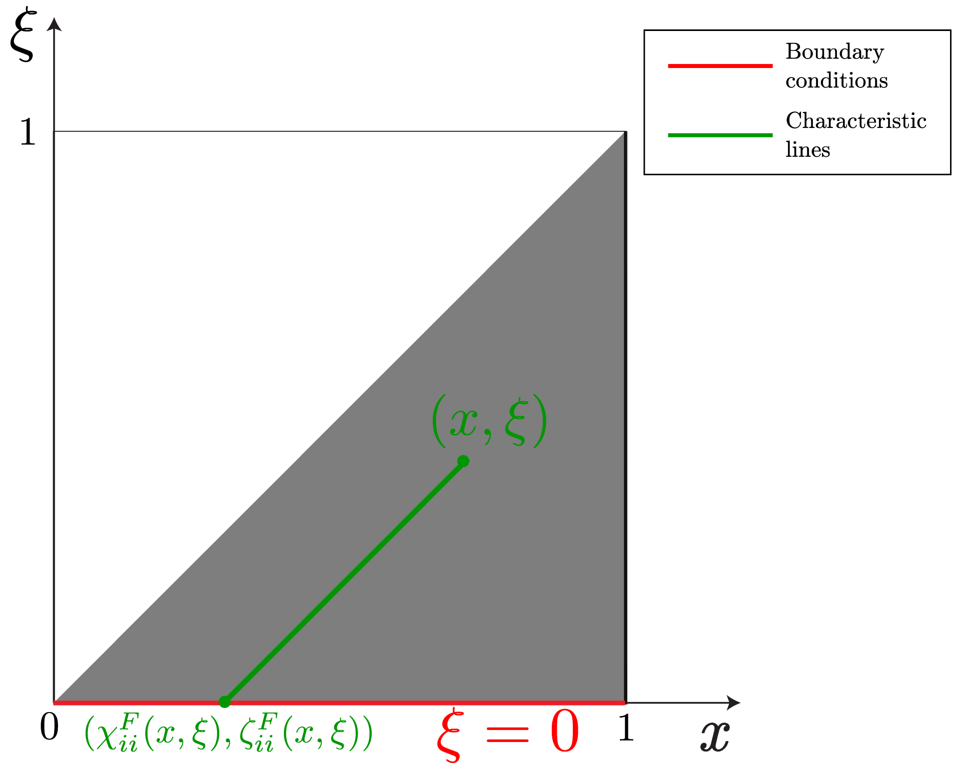
Well-posedness depends on the characteristics!

Characteristics for L_{ij}



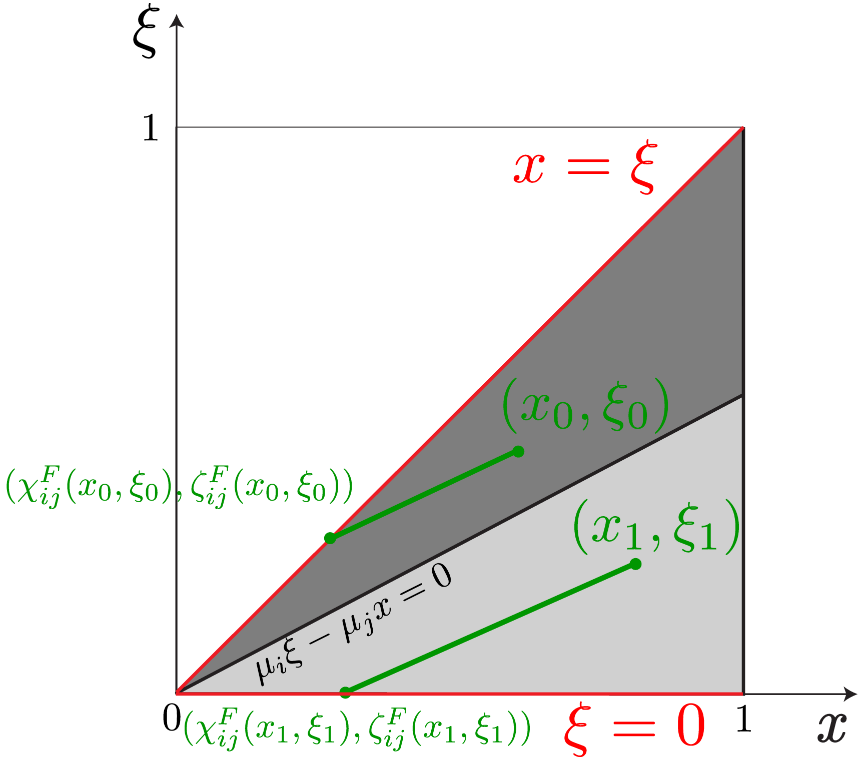
$$
\begin{aligned}
\mu_i \partial_x L_{ij}(x,\xi) - \varepsilon_j \partial_\xi L_{ij}(x,\xi) &= \sum_{k=1}^{n} \lambda_{kj}^{++} L_{ik}(x,\xi) + \sum_{p=1}^{m} \lambda_{pj}^{-+} K_{ip}(x,\xi) \\
L_{ij}(x,x) &= -\frac{\lambda_{ij}^{-+}}{\mu_i + \varepsilon_j}
\end{aligned}
$$

Characteristics for K_{ii}



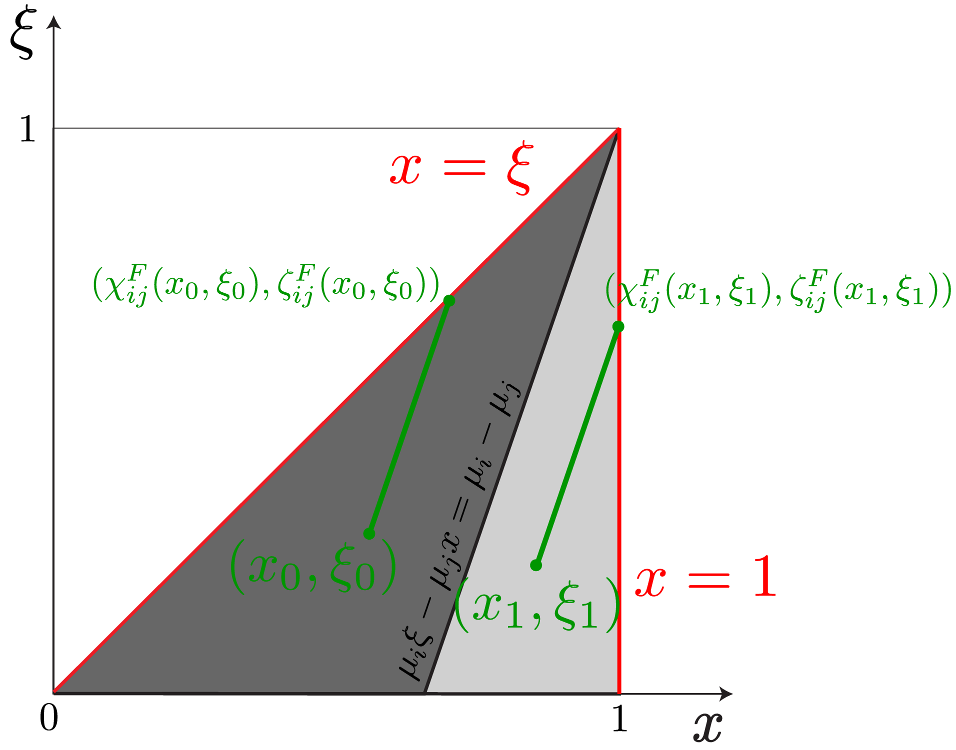
$$\begin{aligned} \mu_i \partial_x K_{ii}(x,\xi) + \mu_i \partial_\xi K_{ii}(x,\xi) &= \sum_{p=1}^{m} \lambda_{pi}^{--} K_{ip}(x,\xi) + \sum_{k=1}^{n} \lambda_{ki}^{+-} L_{ik}(x,\xi) \\ K_{ii}(x,0) &= 0 \end{aligned}$$

Characteristics for $K_{ij}, i < j$



$$
\begin{aligned}
\mu_i \partial_x K_{ij}(x,\xi) + \mu_j \partial_\xi K_{ij}(x,\xi) &= \sum_{p=1}^{m} \lambda_{pj}^{--} K_{ip}(x,\xi) + \sum_{k=1}^{n} \lambda_{kj}^{+-} L_{ik}(x,\xi) \\
K_{ij}(x,x) &= -\frac{\lambda_{ij}^{--}}{\mu_i - \mu_j} \\
K_{ij}(x,0) &= 0
\end{aligned}
$$

Characteristics for $K_{ij}, i > j$



$$
\begin{aligned}
\mu_i \partial_x K_{ij}(x,\xi) + \mu_j \partial_\xi K_{ij}(x,\xi) &= \sum_{p=1}^{m} \lambda_{pj}^{--} K_{ip}(x,\xi) + \sum_{k=1}^{n} \lambda_{kj}^{+-} L_{ik}(x,\xi) \\
K_{ij}(x,x) &= -\frac{\lambda_{ij}^{--}}{\mu_i - \mu_j} \\
K_{ij}(1,\xi) &= l_{ij} \\
g_{ij}(x) &= \mu_j K_{ij}(x,0)
\end{aligned}
$$

Backstepping control of coupled hyperbolic 1-D systems

The presented approach produces piecewise continuous and differentiable kernels.

Backstepping control of coupled hyperbolic 1-D systems

The presented approach produces piecewise continuous and differentiable kernels.

There are potential lines of discontinuity, which complicate kernel calculation, but do not affect the stability result.

Backstepping control of coupled hyperbolic 1-D systems

The presented approach produces piecewise continuous and differentiable kernels.

There are potential lines of discontinuity, which complicate kernel calculation, but do not affect the stability result.

Next we see how we can produce a strikingly similar result for reaction-diffusion equations.

Outline

- Foundations of backstepping: basic design for a 1D parabolic equation
- Coupled hyperbolic systems
- Coupled parabolic systems
- Extension to n-balls
- Bilateral design
- Some open problems

Coupled parabolic systems

Consider

$$u_t(t,x) = \Sigma u_{xx}(t,x) + \Lambda(x)u(t,x)$$

$x \in [0,1]$, $t > 0$, $u \in \mathbb{R}^n$

$$\Sigma = \begin{bmatrix} \varepsilon_1 & 0 & \dots & 0 \\ 0 & \varepsilon_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \varepsilon_n \end{bmatrix}, \quad \Lambda(x) = \begin{bmatrix} \lambda_{11}(x) & \lambda_{12}(x) & \dots & \lambda_{1n}(x) \\ \lambda_{21}(x) & \lambda_{22}(x) & \dots & \lambda_{2n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{n1}(x) & \lambda_{n2}(x) & \dots & \lambda_{nn}(x) \end{bmatrix}$$

with $\varepsilon_i > 0$ ordered, i.e., $\varepsilon_1 > \varepsilon_2 > \dots > \varepsilon_n > 0$, and boundary conditions

$$\begin{aligned} u(0,t) &= 0, \\ u(1,t) &= U(t) \end{aligned}$$

with $U \in \mathbb{R}^n$.

Backstepping approach

Consider the Backstepping Transformation :

$$w(t,x) = u(t,x) - \int_0^x K(x,\xi)u(t,\xi)d\xi$$

with $K(x,\xi)$ a $n \times n$ matrix of kernels, and w verifies the Target System :

$$w_t(t,x) = \Sigma w_{xx}(t,x) - Cw(t,x) - G(x)w_x(0,t),$$

with C and $G(x)$:

$$C = \begin{bmatrix} c_1 & 0 & \dots & 0 \\ 0 & c_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & c_n \end{bmatrix}, \quad G = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ g_{21}(x) & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ g_{(n-1)1}(x) & g_{(n-1)2}(x) & \dots & 0 & 0 \\ g_{n1}(x) & g_{n2}(x) & \dots & g_{n(n-1)}(x) & 0 \end{bmatrix}$$

where $c_1, c_2, \dots, c_n > 0$. Control law is then

$$U(t) = \int_0^1 K(1,\xi)u(t,\xi)d\xi$$

<u>The challenge is to prove that $K(x,\xi)$ exists and has good properties</u> $\longrightarrow$ Kernel equations

Kernel equations

$$\Sigma K_{xx} - K_{\xi\xi}\Sigma = K\Lambda(\xi) + CK,$$

with b.c.

$$\begin{aligned} G(x) &= -K(x,0)\Sigma, \\ K(x,x)\Sigma &= \Sigma K(x,x), \\ C + \Lambda(x) &= -\Sigma K_x(x,x) - \Sigma \frac{d}{dx}K(x,x) - K_\xi(x,x)\Sigma. \end{aligned}$$

Kernel equations

$$\Sigma K_{xx} - K_{\xi\xi}\Sigma = K\Lambda(\xi) + CK,$$

with b.c.

$$\begin{aligned} G(x) &= -K(x,0)\Sigma, \\ K(x,x)\Sigma &= \Sigma K(x,x), \\ C + \Lambda(x) &= -\Sigma K_x(x,x) - \Sigma \frac{d}{dx} K(x,x) - K_\xi(x,x)\Sigma. \end{aligned}$$

First b.c. with structure of G becomes:

$$K_{ij}(x,0) = 0, \qquad \forall j \geq i,$$

and

$$g_{ij}(x) = -K_{ij}(x,0)\varepsilon_j, \qquad \forall j < i,$$

Kernel equations

$$\Sigma K_{xx} - K_{\xi\xi}\Sigma = K\Lambda(\xi) + CK,$$

with b.c.

$$\begin{aligned} G(x) &= -K(x,0)\Sigma, \\ K(x,x)\Sigma &= \Sigma K(x,x), \\ C + \Lambda(x) &= -\Sigma K_x(x,x) - \Sigma \frac{d}{dx} K(x,x) - K_\xi(x,x)\Sigma. \end{aligned}$$

Second b.c. is:

$$K_{ij}(x,x) = 0, \qquad \forall j \neq i,$$

(no boundary condition for $K_{ii}(x,x)$)

Kernel equations

$$\Sigma K_{xx} - K_{\xi\xi}\Sigma = K\Lambda(\xi) + CK,$$

with b.c.

$$\begin{aligned} G(x) &= -K(x,0)\Sigma, \\ K(x,x)\Sigma &= \Sigma K(x,x), \\ C + \Lambda(x) &= -\Sigma K_x(x,x) - \Sigma \frac{d}{dx} K(x,x) - K_\xi(x,x)\Sigma. \end{aligned}$$

Third boundary condition:

$$0 = \lambda_{ij}(x) + \delta_{ij}c_i + K_{ij\xi}(x,x)\varepsilon_j + \varepsilon_i K_{ijx}(x,x) + \varepsilon_i \frac{d}{dx}\left(K_{ij}(x,x)\right),$$

Duplicating the kernel equations

Key idea ("duplication"): define

$$L(x,\xi) \quad = \quad \sqrt{\Sigma}K_x(x,\xi)+K_\xi(x,\xi)\sqrt{\Sigma} \longrightarrow L_{ij}(x,x)=\sqrt{\varepsilon_i}K_{ijx}(x,x)+\sqrt{\varepsilon_j}K_{ij\xi}(x,x)$$

Then we can rewrite the "duplicated" kernel equations as

$$\begin{aligned} \sqrt{\Sigma}K_x+K_\xi\sqrt{\Sigma} &= L \\ \sqrt{\Sigma}L_x-L_\xi\sqrt{\Sigma} &= K\Lambda(\xi)+CK \end{aligned}$$

Same structure as in the coupled hyperbolic result!

Third boundary condition becomes:

$$i=j\text{: } 0=\lambda_{ii}(x)+c_i+2\varepsilon_i(K_{iix}(x,x)+K_{ii\xi}(x,x)) \longrightarrow L_{ii}(x,x)=-\frac{\lambda_{ii}(x)+c_i}{2\sqrt{\varepsilon_i}}$$

$$i\neq j\text{: } 0=\lambda_{ij}(x)+(\varepsilon_i-\varepsilon_j)K_{ijx}(x,x) \longrightarrow L_{ij}(x,x)=-\frac{\lambda_{ij}(x)}{\sqrt{\varepsilon_i}+\sqrt{\varepsilon_j}}$$

Duplicating the kernel equations

The boundary conditions therefore are:

- If $i = j$

$$\begin{aligned} L_{ii}(x,x) &= -\frac{\lambda_{ii}(x) + c_i}{2\sqrt{\varepsilon_i}} \\ K_{ii}(x,0) &= 0 \end{aligned}$$

- If $i < j$

$$\begin{aligned} K_{ij}(x,x) &= K_{ij}(x,0) = 0 \\ L_{ij}(x,x) &= -\frac{\lambda_{ij}(x)}{\sqrt{\varepsilon_i} + \sqrt{\varepsilon_j}} \end{aligned}$$

- Finally if $i > j$

$$\begin{aligned} K_{ij}(x,x) &= 0 \\ K_{ij}(1,\xi) &= l_{ij}(\xi) \\ L_{ij}(x,x) &= -\frac{\lambda_{ij}(x)}{\sqrt{\varepsilon_i} + \sqrt{\varepsilon_j}} \end{aligned}$$

  and the additional condition $g_{ij}(x) = -K_{ij}(x,0)\varepsilon_j$

Same structure as in the coupled hyperbolic result!

Extension to reaction-advection-diffusion systems with spatially-varying coefficients

The method can be extended to

$$u_t = \partial_x(\Sigma(x)u_x) + \Phi(x)u_x + \Lambda(x)u$$

Outline

- Foundations of backstepping: basic design for a 1D parabolic equation
- Coupled hyperbolic systems
- Coupled parabolic systems
- Extension to n-balls
- Bilateral design
- Some open problems

Reaction-diffusion equation on an n-dimensional ball

Let the state $u = u(t,\vec{x})$, with $\vec{x} = [x_1, x_2, \ldots, x_n]^T$, verify

$$\frac{\partial u}{\partial t} \;=\; \varepsilon \left(\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} + \ldots + \frac{\partial^2 u}{\partial x_n^2} \right) + \lambda u = \varepsilon \triangle_n u + \lambda u,$$

for constant $\varepsilon > 0$, $\lambda(r,\vec{\theta})$, and for $t > 0$, in the n-ball $B^n(R)$ defined as

$$B^n(R) = \{\vec{x} \in \mathbb{R}^n : \|\vec{x}\| < R\},$$

with b.c. on the boundary of $B^n(R)$, the $(n-1)$-sphere $S^{n-1}(R)$:

$$S^{n-1}(R) = \{\vec{x} \in \mathbb{R}^n : \|\vec{x}\| = R\}.$$

The b.c. is of Dirichlet type:

$$u(t,\vec{x})\Big|_{\vec{x} \in S^{n-1}(R)} = U(t,\vec{x})$$

where $U(t,\vec{x})$ is the actuation variable.

Ultraspherical coordinates

The n-ball domain is well described in n-dimensional spherical coordinates, also known as ultraspherical coordinates:

- one radial coordinate r, $r \in [0, R)$.
- $n-1$ angular coordinates: $\vec{\theta} = [\theta_1, \theta_2, \ldots, \theta_{n-1}]^T$, with $\theta_1 \in [0, 2\pi)$ and $\theta_i \in [0, \pi]$ for $2 \leq i \leq n-1$.

Definition:

$$
\begin{aligned}
x_1 &= r\cos\theta_1 \sin\theta_2 \sin\theta_3 \ldots \sin\theta_{n-1}, \\
x_2 &= r\sin\theta_1 \sin\theta_2 \sin\theta_3 \ldots \sin\theta_{n-1}, \\
x_3 &= r\cos\theta_2 \sin\theta_3 \ldots \sin\theta_{n-1}, \\
&\vdots \\
x_{n-1} &= r\cos\theta_{n-2} \sin\theta_{n-1}, \\
x_n &= r\cos\theta_{n-1}.
\end{aligned}
$$

Laplacian in ultraspherical coordinates

Writing the reaction diffusion equation in ultraspherical coordinates

$$
\begin{aligned}
u_t &= \frac{\varepsilon}{r^{n-1}}\partial_r\left(r^{n-1}\partial_r u\right) + \frac{1}{r^2}\triangle^*_{n-1}u + \lambda u, \\
u(t,R,\vec{\theta}\,) &= U(t,\vec{\theta}\,),
\end{aligned}
$$

where $\triangle^*_{n-1}$ is called the Laplace-Beltrami operator and represents the Laplacian over the $(n-1)$-sphere.

It is defined recursively as

$$
\begin{aligned}
\triangle^*_1 &= \frac{\partial^2}{\partial\theta_1^2}, \\
\triangle^*_n &= \frac{1}{\sin^{n-1}\theta_n}\frac{\partial}{\partial\theta_n}\left(\sin^{n-1}\theta_n\frac{\partial}{\partial\theta_n}\right) + \frac{\triangle^*_{n-1}}{\sin^2\theta_n},
\end{aligned}
$$

Example:

$$
\triangle^*_2 = \frac{1}{\sin\theta_2}\frac{\partial}{\partial\theta_2}\left(\sin\theta_2\frac{\partial}{\partial\theta_2}\right) + \frac{1}{\sin^2\theta_2}\frac{\partial^2}{\partial\theta_1^2}.
$$

Designing a boundary feedback law

- Exploit periodicity in $\vec{\theta}$ by using Spherical Harmonics

- Apply the backstepping method to each harmonic coefficient

- Solve the backstepping kernel equations to find a feedback law for each harmonic

- Re-assemble the feedback law in Spherical Harmonics back to physical space

Spherical Harmonics

Develop u and U in term of Spherical Harmonics coefficients u_l^m and U_l^m:

$$u(t,r,\vec{\theta}) = \sum_{l=0}^{l=\infty} \sum_{m=0}^{m=N(l,n)-1} u_l^m(r,t) Y_{lm}^n(\vec{\theta}), \quad U(t,\vec{\theta}) = \sum_{l=0}^{l=\infty} \sum_{m=0}^{m=N(l,n)-1} U_l^m(t) Y_{lm}^n(\vec{\theta}),$$

$N(l,n)$: number of (linearly independent) n-dimensional spherical harmonics of degree l

$$N(l,n) = \frac{2l+n-2}{l} \begin{pmatrix} l+n-3 \\ l-1 \end{pmatrix}, \quad l > 0; \qquad N(0,n) = 1$$

$Y_{lm}^n(\vec{\theta})$: m-th order n-dimensional spherical harmonic of degree l

Coefficients are defined as:

$$u_l^m(r,t) = \int_0^{\pi} \ldots \int_0^{\pi} \int_0^{2\pi} u(t,r,\vec{\theta}) \bar{Y}_{lm}^n(\vec{\theta}) \sin^{n-2}\theta_{n-1} \sin^{n-3}\theta_{n-2} \ldots \sin\theta_2 d\vec{\theta},$$

$$U_l^m(t) = \int_0^{\pi} \ldots \int_0^{\pi} \int_0^{2\pi} U(t,\vec{\theta}) \bar{Y}_{lm}^n(\vec{\theta}) \sin^{n-2}\theta_{n-1} \sin^{n-3}\theta_{n-2} \ldots \sin\theta_2 d\vec{\theta},$$

($d\vec{\theta} = d\theta_{n-1} d\theta_{n-2} \ldots d\theta_2 d\theta_1$, $\bar{Y}_{lm}^n$ is the complex conjugate of Y_{lm}^n)

Spherical Harmonics

The n-dimensional spherical harmonics are **eigenfunctions** for the Laplacian $\triangle^*_{n-1}$:

$$\triangle^*_{n-1} Y^n_{lm} = -l(l+n-2)Y^n_{lm}.$$

Thus, each harmonic coefficient $u^m_l(t,r)$ for $l \in \mathbb{N}$ and $0 \le m \le N(l,n)$, verifies

$$\partial_t u^m_l = \frac{\varepsilon}{r^{n-1}} \partial_r \left(r^{n-1} \partial_r u^m_l \right) - l(l+n-2)\frac{\varepsilon}{r^2} u^m_l + \lambda u^m_l,$$

evolving in $r \in [0,R]$, $t > 0$, with boundary conditions

$$u^m_l(t,R) \quad = \quad U^m_l(t),$$

The PDEs for the harmonics are not coupled: we can independently design each U^m_l and later assemble all of the them to find an expression for U.

Backstepping control of Spherical Harmonics coefficients

To design $U_l^m(t)$ seek transformation of

$$\partial_t u_l^m = \frac{\varepsilon}{r^{n-1}}\partial_r\left(r^{n-1}\partial_r u_l^m\right) - l(l+n-2)\frac{\varepsilon}{r^2}u_l^m + \lambda u_l^m$$

into the (stable) target system

$$\partial_t w_l^m = \frac{\varepsilon}{r^{n-1}}\partial_r\left(r^{n-1}\partial_r w_l^m\right) - l(l+n-2)\frac{\varepsilon}{r^2}w_l^m$$

with boundary conditions

$$w_l^m(t,R) = 0$$

Backstepping control of Spherical Harmonics coefficients

To design $U_l^m(t)$ seek transformation of

$$\partial_t u_l^m = \frac{\varepsilon}{r^{n-1}}\partial_r\left(r^{n-1}\partial_r u_l^m\right) - l(l+n-2)\frac{\varepsilon}{r^2}u_l^m + \lambda u_l^m$$

into the (stable) target system

$$\partial_t w_l^m = \frac{\varepsilon}{r^{n-1}}\partial_r\left(r^{n-1}\partial_r w_l^m\right) - l(l+n-2)\frac{\varepsilon}{r^2}w_l^m$$

with boundary conditions

$$w_l^m(t,R) = 0$$

The transformation is

$$w_l^m(t,r) = u_l^m(t,r) - \int_0^r K_{lm}^n(r,\rho)u_l^m(t,\rho)d\rho$$

with kernels K_{lm}^n to be found.

Backstepping control of Spherical Harmonics coefficients

To design $U_l^m(t)$ seek transformation of

$$\partial_t u_l^m \quad = \quad \frac{\varepsilon}{r^{n-1}} \partial_r \left(r^{n-1} \partial_r u_l^m \right) - l(l+n-2) \frac{\varepsilon}{r^2} u_l^m + \lambda u_l^m$$

into the (stable) target system

$$\partial_t w_l^m \quad = \quad \frac{\varepsilon}{r^{n-1}} \partial_r \left(r^{n-1} \partial_r w_l^m \right) - l(l+n-2) \frac{\varepsilon}{r^2} w_l^m$$

with boundary conditions

$$w_l^m(t,R) \quad = \quad 0$$

The transformation is

$$w_l^m(t,r) = u_l^m(t,r) - \int_0^r K_{lm}^n(r,\rho) u_l^m(t,\rho) d\rho$$

with kernels K_{lm}^n to be found.

Substituting at $r = R$ we find $U_l^m(t)$ as

$$U_l^m(t)(t) = \int_0^R K_{lm}^n(R,\rho) u_l^m(t,\rho) d\rho$$

Kernel equation

The control kernels $K_{lm}^n(r,\rho)$ are found, for a given $n \geq 2$ and each l,m, from

$$\frac{1}{r^{n-1}}\partial_r\left(r^{n-1}\partial_r K_{lm}^n\right) - \partial_\rho\left(\rho^{n-1}\partial_\rho\left(\frac{K_{lm}^n}{\rho^{n-1}}\right)\right) - l(l+n-2)\left(\frac{1}{r^2}-\frac{1}{\rho^2}\right)K_{lm}^n = \frac{\lambda}{\varepsilon}K_{lm}^n.$$

with BC

$$\begin{aligned}
\lambda + 2\varepsilon\frac{d}{dr}\left(K_{lm}^n(r,r)\right) &= 0 \\
K_{lm}^n(r,0) &= 0 \\
(n-2)\partial_\rho K_{lm}^n(r,\rho)|_{\rho=0} &= 0
\end{aligned}$$

Kernel equation

The control kernels $K^n_{lm}(r,\rho)$ are found, for a given $n \geq 2$ and each l,m, from

$$\frac{1}{r^{n-1}}\partial_r\left(r^{n-1}\partial_r K^n_{lm}\right) - \partial_\rho\left(\rho^{n-1}\partial_\rho\left(\frac{K^n_{lm}}{\rho^{n-1}}\right)\right) - l(l+n-2)\left(\frac{1}{r^2}-\frac{1}{\rho^2}\right)K^n_{lm} = \frac{\lambda}{\varepsilon}K^n_{lm}.$$

with BC

$$\begin{aligned}
\lambda + 2\varepsilon\frac{d}{dr}\left(K^n_{lm}(r,r)\right) &= 0\\
K^n_{lm}(r,0) &= 0\\
(n-2)\partial_\rho K^n_{lm}(r,\rho)|_{\rho=0} &= 0
\end{aligned}$$

The first BC integrates (using $K^n_{lm}(0,0)=0$) to

$$K^n_{lm}(r,r) = -\int_0^r \frac{\lambda}{2\varepsilon}d\rho = -\frac{\lambda r}{2\varepsilon}$$

Explicit Kernel equation solution and feedback law

It is found that

$$K_{lm}^{n}(r,\rho) = -\rho\left(\frac{\rho}{r}\right)^{l+n-2}\frac{\lambda}{\varepsilon}\frac{\mathrm{I}_1\left[\sqrt{\frac{\lambda}{\varepsilon}(r^2-\rho^2)}\right]}{\sqrt{\frac{\lambda}{\varepsilon}(r^2-\rho^2)}}$$

Thus the feedback law for each spherical harmonic is

$$U_l^m(t) = \int_0^R K_{lm}^n(R,\rho)u_l^m(t,\rho)d\rho = \int_0^R -\rho\left(\frac{\rho}{R}\right)^{l+n-2}\frac{\lambda}{\varepsilon}\frac{\mathrm{I}_1\left[\sqrt{\frac{\lambda}{\varepsilon}(R^2-\rho^2)}\right]}{\sqrt{\frac{\lambda}{\varepsilon}(R^2-\rho^2)}}u_l^m(t,\rho)d\rho$$

Explicit feedback law

Using some spherical harmonics machinery one obtains an explicit feedback law

$$
\begin{aligned}
U(t,\theta) &= -\frac{\lambda}{\varepsilon}\int_0^R \rho \frac{\mathrm{I}_1\left[\sqrt{\frac{\lambda}{\varepsilon}(R^2-\rho^2)}\right]}{\sqrt{\frac{\lambda}{\varepsilon}(R^2-\rho^2)}} \\
&\quad \times \left[\int_0^{\pi}\!\!\cdots\int_0^{\pi}\int_0^{2\pi} P(R,\rho,\vec{\theta},\vec{\phi})\,u(t,\rho,\vec{\phi})\rho^{n-1}\sin^{n-2}\phi_{n-1}\sin^{n-3}\phi_{n-2}\ldots\sin\phi_2\, d\vec{\phi}\right] d\rho
\end{aligned}
$$

where $P(R,\rho,\vec{\theta},\vec{\phi})$ is the Poisson kernel for the n-ball.

Back in rectangular coordinates

$$
U(t,\vec{x}) = -\frac{1}{\mathrm{Area}(S^{n-1})}\sqrt{\frac{\lambda}{\varepsilon}}\int_{B^n(R)} \mathrm{I}_1\left[\sqrt{\frac{\lambda}{\varepsilon}(R^2-\|\vec{\xi}\|^2)}\right]\frac{\sqrt{R^2-\|\vec{\xi}\|^2}}{\|\vec{x}-\vec{\xi}\|^n}u(t,\vec{\xi})\,d\vec{\xi},
$$

where the integral is extended to the complete n-ball $B^n(R)$ and $\vec{x}\in S^{n-1}(R)$.

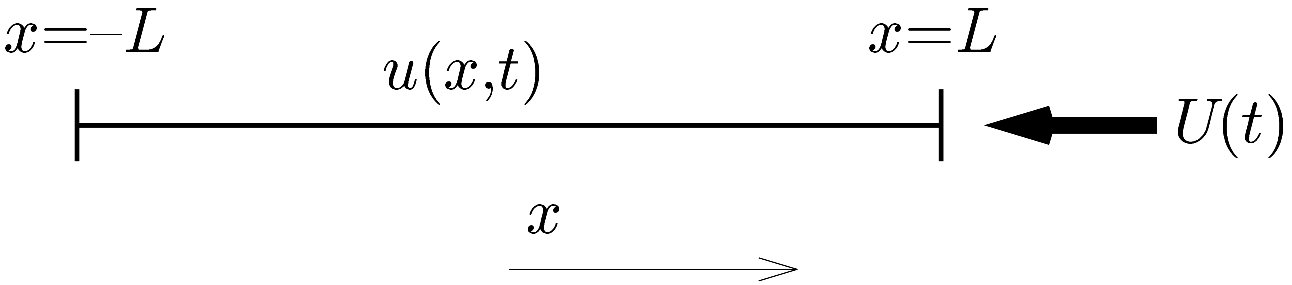
Outline

- Foundations of backstepping: basic design for a 1D parabolic equation
- Coupled hyperbolic systems
- Coupled parabolic systems
- Extension to n-balls
- Bilateral design
- Some open problems

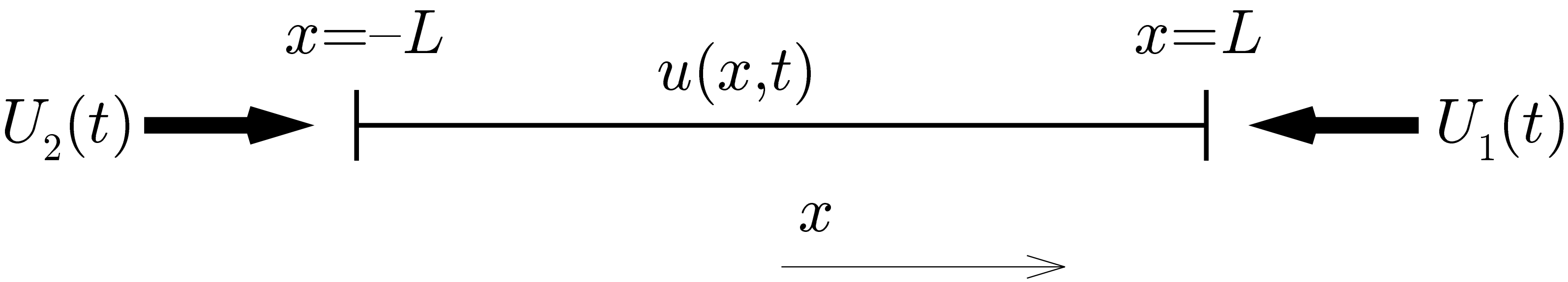
Bilateral vs. unilateral boundary control

1-D PDE boundary control problems, $x \in [-L, L]$

Unilateral boundary control: One side controlled, one side uncontrolled



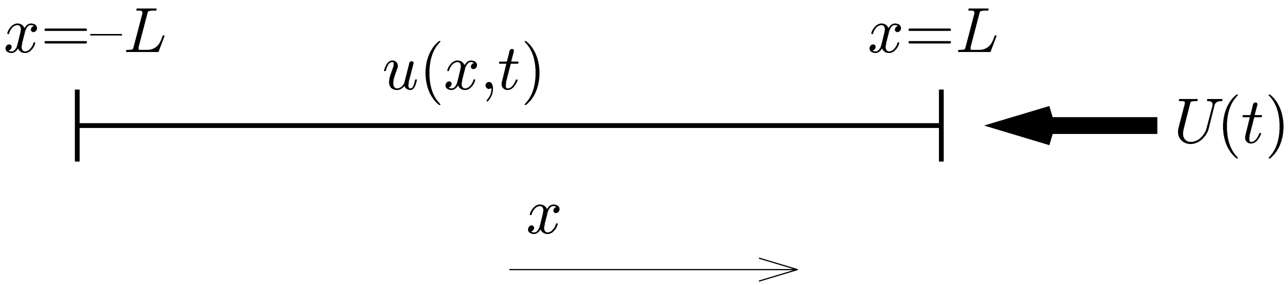
Bilateral boundary control: Both sides controlled



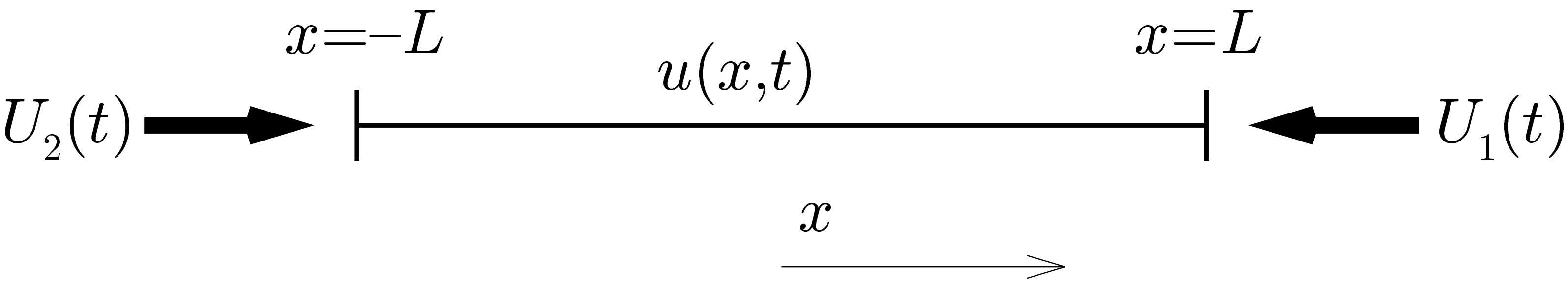
Bilateral vs. unilateral boundary control

1-D PDE boundary control problems, $x \in [-L, L]$

Unilateral boundary control: One side controlled, one side uncontrolled



Bilateral boundary control: Both sides controlled



- One more actuator, but total effort may be less(more on this later)
- Backstepping design has to be modified (transformation needs to be changed)
- Interesting in practice, opens the door for fault-tolerant designs
- Similar results for boundary observer with sensors on both ends

Reaction-diffusion PDEs

Bilateral problem:

$$\begin{aligned} u_t &= \varepsilon u_{xx} + \lambda(x)u \\ u(t,L) &= U_1(t) \\ u(t,-L) &= U_2(t) \end{aligned}$$

Reaction-diffusion PDEs

Bilateral problem:

$$
\begin{aligned}
u_t &= \varepsilon u_{xx} + \lambda(x) u \\
u(t,L) &= U_1(t) \\
u(t,-L) &= U_2(t)
\end{aligned}
$$

Bilateral backstepping design:

$$
w(t,x) = u(t,x) - \int_{-x}^{x} K(x,\xi) u(t,\xi) d\xi
$$

with $w(t,x)$ (target variable) verifying

$$
\begin{aligned}
w_t &= \varepsilon w_{xx} \\
w(t,L) &= w(t,-L) = 0
\end{aligned}
$$

Reaction-diffusion PDEs

Bilateral problem:

$$\begin{aligned} u_t &= \varepsilon u_{xx} + \lambda(x) u \\ u(t, L) &= U_1(t) \\ u(t, -L) &= U_2(t) \end{aligned}$$

Bilateral backstepping design:

$$w(t,x) = u(t,x) - \int_{-x}^{x} K(x,\xi) u(t,\xi) d\xi \quad \longrightarrow \quad U_1(t) = \int_{-L}^{L} K(L,\xi) u(t,\xi) d\xi, \;\; U_2(t) = -\int_{-L}^{L} K(-L,\xi) u(t,\xi) d\xi$$

with $w(t,x)$ (target variable) verifying

$$\begin{aligned} w_t &= \varepsilon w_{xx} \\ w(t, L) &= w(t, -L) = 0 \end{aligned}$$

Reaction-diffusion PDEs

Bilateral problem:

$$\begin{aligned} u_t &= \varepsilon u_{xx} + \lambda(x)u \\ u(t,L) &= U_1(t) \\ u(t,-L) &= U_2(t) \end{aligned}$$

Bilateral backstepping design:

$$w(t,x) = u(t,x) - \int_{-x}^{x} K(x,\xi)u(t,\xi)d\xi \quad \longrightarrow \quad U_1(t) = \int_{-L}^{L} K(L,\xi)u(t,\xi)d\xi,\ U_2(t) = -\int_{-L}^{L} K(-L,\xi)u(t,\xi)d\xi$$

with $w(t,x)$ (target variable) verifying

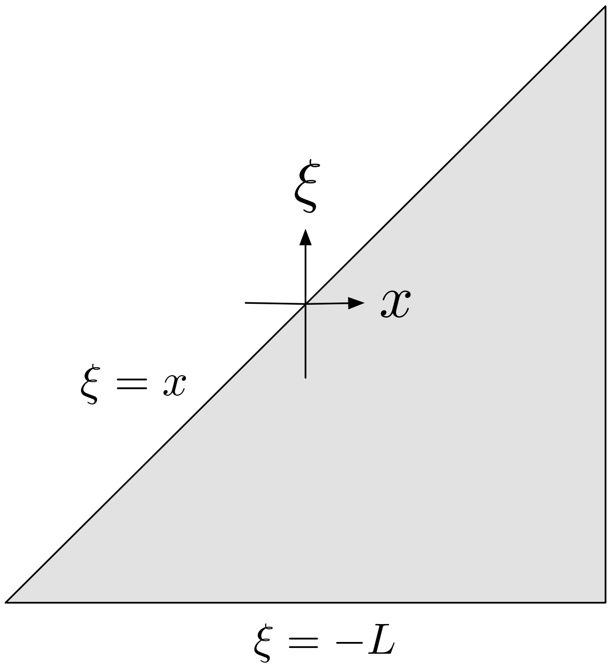
$$\begin{aligned} w_t &= \varepsilon w_{xx} \\ w(t,L) &= w(t,-L) = 0 \end{aligned}$$

Kernel equations:

$$\begin{aligned} K_{xx}(x,\xi) - K_{\xi\xi}(x,\xi) &= \frac{\lambda(\xi)}{\varepsilon}K(x,\xi) \quad \text{for} \quad x \in [-L,L],\ \xi \in [-|x|,|x|] \\ K(x,x) &= -\int_0^x \frac{\lambda(\xi)}{2\varepsilon}d\xi \\ K(x,-x) &= 0 \end{aligned}$$

Reaction-diffusion PDEs

Domain for unilateral kernel equations

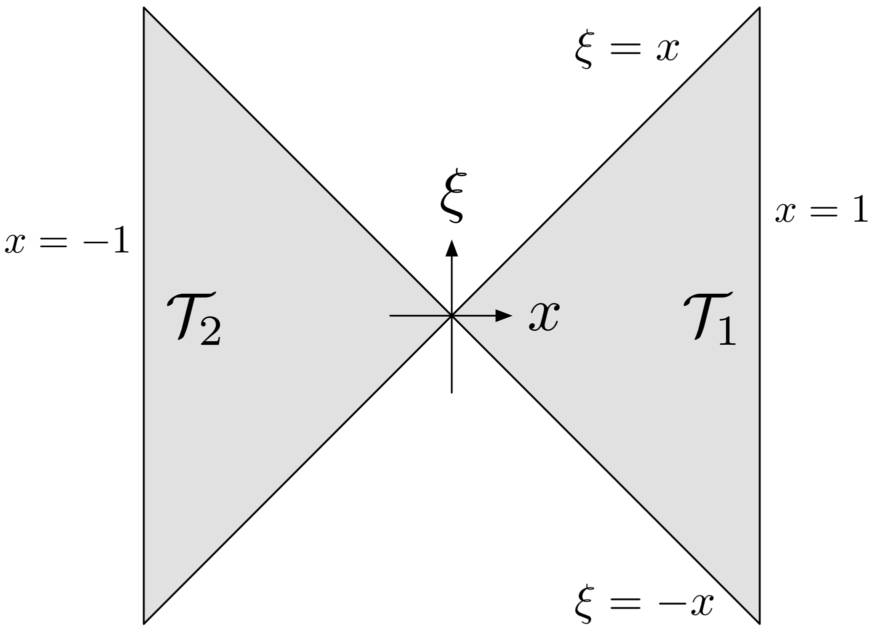


- Right sided triangle

- Information propagates from diagonal ($\xi = x$) and lower ($\xi = 0$) boundary up to $x = L$ (control kernel)

Reaction-diffusion PDEs

Domain for bilateral kernel equations

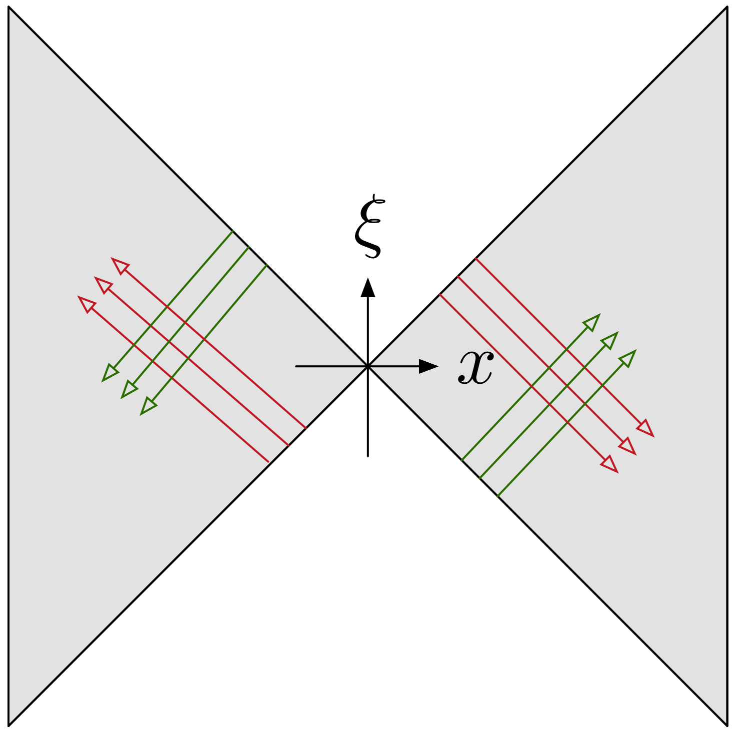


- Hourglass shape , two independent domains $\mathcal{T}_1, \mathcal{T}_2$

- Information propagates from both diagonals ($\xi = x$, $\xi = -x$) up to the boundaries $x = L, -L$ (control kernels)

Reaction-diffusion PDEs

Domain for bilateral kernel equations: characteristics



Both boundaries are characteristic! Actually an easier problem. The boundary conditions have to match at zero for continuity

Reaction-diffusion PDEs

Explicit control law for unilateral problem (Smyshlyaev & Krstic, IEEE TAC 2004) for constant λ

$$U = -\int_{-L}^{L} \sqrt{\frac{\lambda}{\varepsilon}} \frac{\xi}{\sqrt{4L^2 - (\xi + L)^2}} \mathrm{I}_1 \left[\sqrt{\frac{\lambda}{\varepsilon}(4L^2 - (\xi + L)^2)} \right] u(t,\xi) d\xi$$

Reaction-diffusion PDEs

Explicit control law for unilateral problem (Smyshlyaev & Krstic, IEEE TAC 2005) for constant λ

$$U = -\int_{-L}^{L} \sqrt{\frac{\lambda}{\varepsilon}} \frac{\xi}{\sqrt{4L^2 - (\xi + L)^2}} \mathrm{I}_1 \left[\sqrt{\frac{\lambda}{\varepsilon}(4L^2 - (\xi + L)^2)} \right] u(t,\xi) d\xi$$

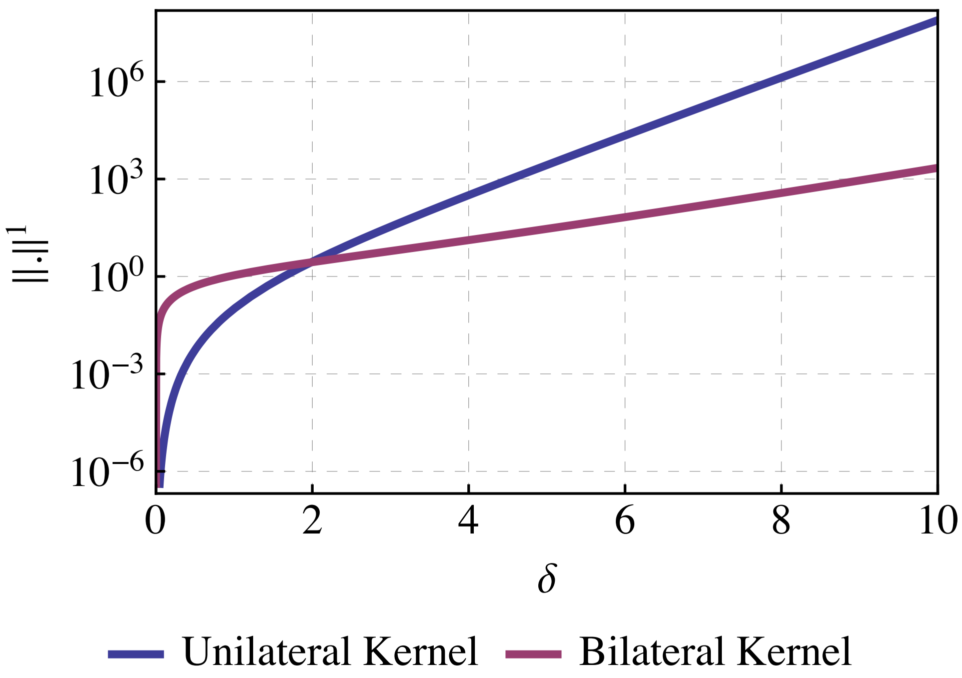
Explicit control law for bilateral problem

$$\begin{aligned} U_1 &= -\frac{1}{2}\sqrt{\frac{\lambda}{\varepsilon}} \int_{-L}^{L} \sqrt{\frac{L+\xi}{L-\xi}} \mathrm{I}_1 \left[\sqrt{\frac{\lambda}{\varepsilon}(L^2 - \xi^2)} \right] u(t,\xi) d\xi \\ U_2 &= -\frac{1}{2}\sqrt{\frac{\lambda}{\varepsilon}} \int_{-L}^{L} \sqrt{\frac{L-\xi}{L+\xi}} \mathrm{I}_1 \left[\sqrt{\frac{\lambda}{\varepsilon}(L^2 - \xi^2)} \right] u(t,\xi) d\xi \end{aligned}$$

Reaction-diffusion PDEs

Comparing unilateral and bilateral control laws

Writing $\delta = L\sqrt{\frac{\lambda}{\varepsilon}}$



For sufficiently large $\delta = L\sqrt{\frac{\lambda}{\varepsilon}}$ the bilateral control law requires less total actuation

(long domains, slow diffusion, and/or highly unstable plants)

Outline

- Foundations of backstepping: basic design for a 1D parabolic equation
- Coupled hyperbolic systems
- Coupled parabolic systems
- Extension to n-balls
- Bilateral design
- Some open problems

Some open problems

- Underactuated coupled hyperbolic and parabolic systems.

- Robustness properties of backstepping controllers.

- Non-strict-feedback terms (terms that are not “spatially-causal”).

- Reaction-diffusion equation in the n-ball with non-constant diffusion.

Design on the disk with $\lambda(r, \theta)$

$$u_t = \frac{\varepsilon}{r}(ru_r)_r + \frac{\varepsilon}{r^2}u_{\theta\theta} + \lambda(r,\theta)u,$$

It is not possible to use spherical harmonics (they are no longer eigenfunctions that decouple the problem).

Pose a physical-space transformation:

$$w = u - \int_0^r \int_{-\pi}^{\pi} K(r,\rho,\theta,\psi)u(\rho,\psi)d\psi d\rho,$$

to transform the u equation into the target system

$$w_t = \frac{\varepsilon}{r}(rw_r)_r + \frac{\varepsilon}{r^2}w_{\theta\theta},$$

Design on the disk with $\lambda(r,\theta)$

The kernel verifies the ultrahyperbolic equation

$$K_{rr}+\frac{K_r}{r}-K_{\rho\rho}+\frac{K_\rho}{\rho}-\frac{K}{\rho^2}-\frac{K_{\theta\theta}}{r^2}+\frac{K_{\psi\psi}}{\rho^2}=\frac{\lambda(\rho,\psi)}{\varepsilon}K$$

with BC

$$\begin{aligned}
K(r,\rho,0,\psi) &= K(r,\rho,\pi,\psi) \\
K(r,\rho,\theta,0) &= K(r,\rho,\theta,\pi) \\
K(r,0,\theta,\psi) &= 0, \\
\int_{-\pi}^{\pi} K(r,r,\theta,\psi)u(r,\psi)d\psi &= -\int_0^r \frac{\lambda(\rho,\theta)}{2\varepsilon}d\rho u(r,\theta),
\end{aligned}$$

this last boundary condition can be verified if

$$\lim_{\rho\to r} K(r,\rho,\theta,\psi) = -\delta(\theta-\psi)\int_0^r \frac{\lambda(\rho,\theta)}{2\varepsilon}d\rho.$$

Design on the disk with $\lambda(r,\theta)$

The kernel verifies the ultrahyperbolic equation

$$K_{rr}+\frac{K_r}{r}-K_{\rho\rho}+\frac{K_\rho}{\rho}-\frac{K}{\rho^2}-\frac{K_{\theta\theta}}{r^2}+\frac{K_{\psi\psi}}{\rho^2}=\frac{\lambda(\rho,\psi)}{\varepsilon}K$$

with BC

$$\begin{aligned} K(r,\rho,0,\psi) &= K(r,\rho,\pi,\psi) \\ K(r,\rho,\theta,0) &= K(r,\rho,\theta,\pi) \\ K(r,0,\theta,\psi) &= 0, \\ \int_{-\pi}^{\pi} K(r,r,\theta,\psi)u(r,\psi)d\psi &= -\int_0^r \frac{\lambda(\rho,\theta)}{2\varepsilon}d\rho u(r,\theta), \end{aligned}$$

this last boundary condition can be verified if

$$\lim_{\rho\to r} K(r,\rho,\theta,\psi) = -\delta(\theta-\psi)\int_0^r \frac{\lambda(\rho,\theta)}{2\varepsilon}d\rho.$$

We don't know how to solve, only know there is a solution for constant λ!

$$K(r,\rho,\theta,\psi)=-\rho\frac{\lambda}{2\pi\varepsilon}\frac{\mathrm{I}_1\left[\sqrt{\frac{\lambda}{\varepsilon}(r^2-\rho^2)}\right]}{\sqrt{\frac{\lambda}{\varepsilon}(r^2-\rho^2)}}\frac{r^2-\rho^2}{r^2+\rho^2-2r\rho\cos(\theta-\psi)}$$

Muito Obrigado!

Questions?

Some references:

- A. Smyshlyaev and M. Krstic, “Closed form boundary state feedbacks for a class of partial integro-differential equations,” IEEE Transactions on Automatic Control, 2004.
- Long Hu, Florent Di Meglio, Rafael Vazquez, Miroslav Krstic, ”Control of Homodirectional and General Heterodirectional Linear Coupled Hyperbolic PDEs,” IEEE Transactions on Automatic Control, 2016.
- R. Vazquez and M. Krstic, ”Boundary Control of Coupled Reaction-Advection-Diffusion Systems with Spatially-Varying Coefficients,” IEEE Transactions on Automatic Control, 2017.
- Long Hu, Rafael Vazquez, Florent Di Meglio, Miroslav Krstic, ”Boundary exponential stabilization of 1-D inhomogeneous quasilinear hyperbolic systems,” under review, 2016. Preprint in ArXiv.
- R. Vazquez and M. Krstic, ”Explicit output-feedback boundary control of reaction-diffusion PDEs on arbitrary-dimensional balls,” ESAIM:COCV, 2016.
- R. Vazquez and M. Krstic, ”Bilateral Boundary Control of One-Dimensional First- and Second-Order PDEs using Infinite-Dimensional Backstepping, ” Conference on Decision and Control, Las Vegas, 2016.

Contact: rvazquez1@us.es