Offline Learning of Gain Maps Enables **Online**-Adaptive PDE Control

Miroslav Krstic

• AFOSR 2024 •

Project covers

1. Stabilization of **ensemble** PDEs (last year's presentation)
2. **Neural operators** for PDE control (today)
3. Control of **population** dynamics PDEs (next year)

Offline Learning of Gain Maps Enables **Online**-Adaptive PDE Control

Miroslav Krstic

AFOSR 2024

Pubs over the last 12 months

(on DeepONet-PDE backstepping)

[1] L. Bhan, Y. Shi, and M. Krstic, “Neural operators for bypassing gain and control computations in PDE backstepping,” *IEEE Transactions on Automatic Control*, vol. 69, pp. 5310-5325, 2024.

[2] M. Krstic, L. Bhan, Y. Shi, "Neural operators of backstepping controller and observer gain functions for reaction-diffusion PDEs,” *Automatica*, paper 111649, 2024.

[3] J. Qi, J. Zhang, and M. Krstic, “Neural operators for PDE backstepping control of first-order hyperbolic PIDE with recycle and delay,” *System & Control Letters*, paper 105714, 2024.

[4] S.-S. Wang, M. Diagne, and M. Krstic, “Deep learning of delay-compensated backstepping for reaction-diffusion PDEs,” *IEEE Transactions on Automatic Control*, under review.

[5] M. Lamarque, L. Bhan, R. Vazquez, and M. Krstic, “Gain scheduling with a neural operator for a transport PDE with nonlinear recirculation,” *IEEE Transactions on Automatic Control*, under review.

[6] M. Lamarque, L. Bhan, Y.-Y. Shi, and M. Krstic, “Adaptive neural-operator backstepping control of a benchmark hyperbolic PDE,” *Automatica*, under review.

[7] S.-S. Wang, M. Diagne, and M. Krstic, “Backstepping neural operators for 2x2 hyperbolic PDEs,” *Automatica*, under review.

[8] L. Bhan, Y.-Y. Shi, and M. Krstic, “Adaptive control of reaction-diffusion PDEs via neural operator-approximated gain kernels,” *Systems & Control Letters*, under review.

Pubs in this talk

[1] L. Bhan, Y. Shi, and M. Krstic, “Neural operators for bypassing gain and control computations in PDE backstepping,” *IEEE Transactions on Automatic Control*, vol. 69, pp. 5310-5325, 2024.

[2] M. Krstic, L. Bhan, Y. Shi, "Neural operators of backstepping controller and observer gain functions for reaction-diffusion PDEs,” *Automatica*, paper 111649, 2024.

[3] J. Qi, J. Zhang, and M. Krstic, “Neural operators for PDE backstepping control of first-order hyperbolic PIDE with recycle and delay,” *System & Control Letters*, paper 105714, 2024.

[4] S.-S. Wang, M. Diagne, and M. Krstic, “Deep learning of delay-compensated backstepping for reaction-diffusion PDEs,” *IEEE Transactions on Automatic Control*, under review.

[5] M. Lamarque, L. Bhan, R. Vazquez, and M. Krstic, “Gain scheduling with a neural operator for a transport PDE with nonlinear recirculation,” *IEEE Transactions on Automatic Control*, under review.

[6] M. Lamarque, L. Bhan, Y.-Y. Shi, and M. Krstic, “**Adaptive** neural-operator backstepping control of a benchmark **hyperbolic** PDE,” *Automatica*, under review.

[7] S.-S. Wang, M. Diagne, and M. Krstic, “Backstepping neural operators for 2x2 hyperbolic PDEs,” *Automatica*, under review.

[8] L. Bhan, Y.-Y. Shi, and M. Krstic, “**Adaptive** control of **reaction-diffusion** PDEs via neural operator-approximated gain kernels,” *Systems & Control Letters*, nder review.

ML use here

Encode the **map**

$$\boxed{\boxed{\text{PDE model} \quad \mapsto \quad \text{PDE control gains}}}$$

using ML, for existing (rigorous) model-based PDE control designs

ML use here

Encode the **map**

$$\boxed{\text{PDE model} \quad \mapsto \quad \text{PDE control gains}}$$

using ML, for existing (rigorous) model-based PDE control designs

Benefit: $1000\times$ speedup of implementation. Certifications retained.

Deep **Neural Operators**

DeepONet universal approximation THEOREM

(Lu, Jin, Karniadakis, **2021**)

DeepONet universal approximation THEOREM

(Lu, Jin, Karniadakis, **2021**)

Let $\mathcal{S}$ be a **continuous operator**.

DeepONet universal approximation THEOREM

(Lu, Jin, Karniadakis, **2021**)

Let $\mathcal{S}$ be a **continuous operator**.

For every $\epsilon > 0$, there exists a (deep) neural operator $\hat{\mathcal{S}}$ (dependent on ϵ) s.t.

$$\boxed{\left|\mathcal{S}(\cdot) - \hat{\mathcal{S}}(\cdot)\right| < \epsilon}$$

DeepONet universal approximation THEOREM

(Lu, Jin, Karniadakis, **2021**)

Let $\mathcal{S}$ be a **continuous operator**.

For every $\epsilon > 0$, there exists a (deep) neural operator $\hat{\mathcal{S}}$ (dependent on ϵ) s.t.

$$\boxed{\left| \mathcal{S}(\cdot) - \hat{\mathcal{S}}(\cdot) \right| < \epsilon}$$

$\forall$ input functions $(\cdot)$ in a **compact** set of cont. functions.

Example of "nonlinear operator"

feedback: nonlin. mapping of

open-loop response $h(t) \quad \mapsto \quad$ *closed-loop* response

Example of "nonlinear operator"

feedback: nonlin. mapping of

open-loop response $h(t) \quad \mapsto \quad$ *closed-loop* response

We can find closed-loop response for EACH individual $h(t)$,

but can we find a "**once-and-for-all** formula" **for all** $h(t)$?

$\mathcal{S}$ensitivity & $\mathcal{K}$omplementary sensitivity operators

$\mathcal{S}$ensitivity & $\mathcal{K}$omplementary sensitivity **operators**

$$\mathcal{S} : \quad h \;\mapsto\; \mathcal{L}^{-1} \circ \boxed{\frac{1}{1 - \mathcal{L}(h)}}$$

Operator input h: plant impulse response function

$\mathcal{S}$ensitivity & $\mathcal{K}$omplementary sensitivity operators

$$\mathcal{S}: \quad h \;\mapsto\; \mathcal{L}^{-1} \circ \frac{1}{1 - \mathcal{L}(h)}$$

Operator input h: plant impulse response function

Operator output:

- $\mathcal{S}(h)$ = closed-loop impulse response to *measurement disturbance*

$\mathcal{S}$ensitivity & $\mathcal{K}$omplementary sensitivity operators

$$\mathcal{S}: \quad h \quad \mapsto \quad \mathcal{L}^{-1} \circ \frac{1}{1 - \mathcal{L}(h)}$$

$$\mathcal{K}: \quad h \quad \mapsto \quad h - \mathcal{S}(h) \quad = \quad \mathcal{L}^{-1} \circ \frac{-\mathcal{L}(h)}{1 - \mathcal{L}(h)}$$

Operator input h: plant impulse response function

Operator output:

- $\mathcal{S}(h)$ = closed-loop impulse response to *measurement disturbance*
- $\mathcal{K}(h)$ = closed-loop impulse response to *command*

$\mathcal{S}$ensitivity & $\mathcal{K}$omplementary sensitivity operators

$$\mathcal{S}: \quad h \quad \mapsto \quad \mathcal{L}^{-1} \circ \frac{1}{1 - \mathcal{L}(h)}$$

$$\mathcal{K}: \quad h \quad \mapsto \quad h - \mathcal{S}(h) \quad = \quad \mathcal{L}^{-1} \circ \frac{-\mathcal{L}(h)}{1 - \mathcal{L}(h)}, \qquad \boxed{\mathcal{K}^{-1} = \mathcal{K}} \quad \text{involution}$$

Are $\mathcal{S}$ and $\mathcal{K}$ continuous operators?

Lipschitz constants

$$\begin{aligned} L_{\mathcal{S}} &= \mathrm{e}^{2B} \\ L_{\mathcal{K}} &= \left(1 + B\mathrm{e}^{B}\right)\mathrm{e}^{B} \end{aligned}$$

for all impulse response inputs $h(t)$ that are L_∞-bounded by B over finite time

$\hat{\mathcal{S}} \approx \mathcal{S}$ and $\hat{\mathcal{K}} \approx \mathcal{K}$

Theorem. For all $B > 0$ and $\epsilon > 0$, there exists a DeepONet $\widehat{\mathcal{K}}$ satisfying

$$\boxed{\left|\mathcal{K}(h)(t) - \widehat{\mathcal{K}}(h)(t)\right| < \epsilon}$$

for all $\|h\|_\infty \leq B$ and $t \in [0, T]$.

$\hat{\mathcal{S}} \approx \mathcal{S}$ and $\hat{\mathcal{K}} \approx \mathcal{K}$

Theorem. For all $B > 0$ and $\epsilon > 0$, there exists a DeepONet $\widehat{\mathcal{K}}$ satisfying

$$\boxed{\left| \mathcal{K}(h)(t) - \widehat{\mathcal{K}}(h)(t) \right| < \epsilon}$$

for all $\|h\|_\infty \leq B$ and $t \in [0, T]$.

- $\widehat{\mathcal{K}}$ produces approximate closed-loop impulse response
- for *experimentally measured plant impulse response* (even if plant is ∞-dim.)

Outline

1. **Hyperbolic** PDE: operators on fcns of one variable
2. **Parabolic** PDE: operators on fcns of two variables

Outline

1. **Hyperbolic** PDE: operators on fcns of one variable
2. **Parabolic** PDE: operators on fcns of two variables
3. **Adaptive control** — unknown functional coefficients

Hyperbolic benchmark

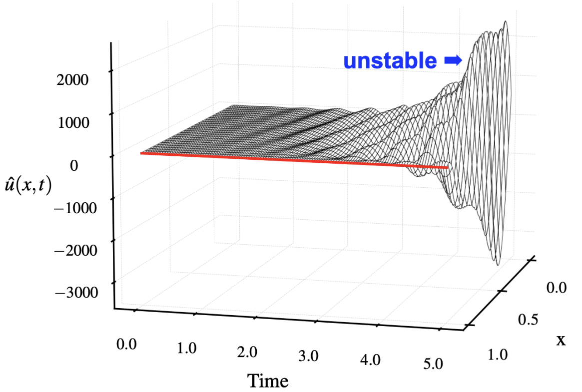
Hyperbolic PDE example: a pedagogical start

(mapping from/into fcns of one variable)

Simplest unstable PDE:

$$u_t(x,t) = u_x(x,t) \boxed{+ \beta(x) \; u(0,t)}$$

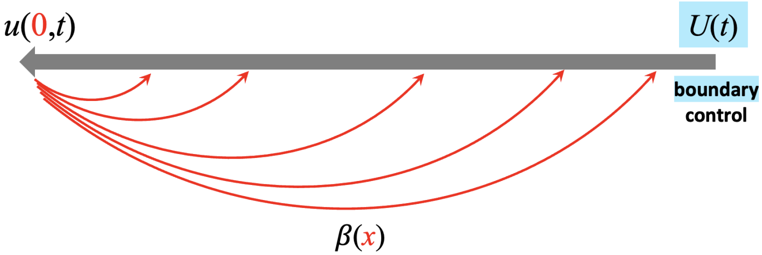
transport + recirculation = instability



Question #1 in *boundary* control:

$$u(1,t) = U(t)$$

boundary control



How to **UNLINK** domain-wide recirculation using only boundary actuation?

Answer: **PDE backstepping design**

Answer: PDE backstepping design

Backstepping operator, transforms $u(x, t)$ to $w(x, t)$:

$$\boxed{w = \mathcal{S}(\beta) * u}$$

inverse: $u = w - \beta * w$

Answer: PDE backstepping design

Backstepping operator, transforms $u(x, t)$ to $w(x, t)$:

$$\boxed{w = \mathcal{S}(\beta) * u} \qquad \text{inverse: } u = w - \beta * w$$

boundary control (t suppressed)

$$U = \mathcal{K}(\beta) * u\Big|_{x=1}$$

Answer: PDE backstepping design

Backstepping operator, transforms $u(x,t)$ to $w(x,t)$:

$$\boxed{w = \mathcal{S}(\beta) * u} \qquad \text{inverse: } u = w - \beta * w$$

boundary control (t suppressed)

$$U = \mathcal{K}(\beta) * u\Big|_{x=1}$$

target system (transport PDE/delay)

$$\begin{aligned} w_t &= w_x \cancel{+\beta(x)u(0,t)} \\ w(1,t) &= \mathbf{0} \end{aligned} \qquad \text{(recirculation **gone**!)}$$

Adaptive Control for
Unknown Functional Coefficients

offline-**on**line learning

combined

DeepONet with online parameter estimation

$\beta(x)$ = *unknown function*

DeepONet with online parameter estimation

$$
\begin{aligned}
\beta(x) &= \text{unknown function} \\
\hat{\beta}(x,t) &= \text{online-updated estimate, with projection to guarantee } \|\hat{\beta}(\cdot,t)\|_\infty \le B,\ \forall t \ge 0
\end{aligned}
$$

DeepONet with online parameter estimation

$\beta(x)$ = *unknown function*

$\hat{\beta}(x,t)$ = online-updated estimate, with projection to guarantee $\|\hat{\beta}(\cdot,t)\|_\infty \le B,\ \forall t \ge 0$

exact kernel	$\mathcal{K}(\beta)$
exact estimated kernel	$\mathcal{K}\left(\widehat{\beta}\right)$
approximate estimated kernel (adaptive kernel)	$\widehat{\mathcal{K}}\left(\widehat{\beta}\right)$

Adaptive Controller

$$U(t) = \int_0^1 \hat{\mathcal{K}}(\hat{\beta})(1-y,t)\, u(y,t)dy$$

Update law

(ensemble/∞-dim nonlinear ODE)

$$\frac{\partial}{\partial t}\hat{\beta}(x,t) = \underbrace{\frac{\gamma}{1+\|w(t)\|_c^2}}_{\text{normalization}} \underbrace{\left[e^{cx} w(x,t) - \int_x^1 e^{cy}\,\hat{\mathcal{K}}(\hat{\beta})(y-x,t)\, w(y,t)dy\right]}_{\text{regressor}} \underbrace{u(0,t)}_{\substack{\text{regulation}\\ \text{error}}}$$

Update law

(ensemble/∞-dim nonlinear ODE)

$$\frac{\partial}{\partial t}\hat{\beta}(x,t) = \underbrace{\frac{\gamma}{1+\|w(t)\|_c^2}}_{\text{normalization}} \underbrace{\left[e^{cx}w(x,t) - \int_x^1 e^{cy}\, \hat{\mathcal{K}}(\hat{\beta})(y-x,t)\; w(y,t)dy\right]}_{\text{regressor}} \underbrace{u(0,t)}_{\substack{\text{regulation}\\ \text{error}}}$$

where

$$\begin{aligned} w(x,t) &= u(x,t) - \int_0^x \hat{\mathcal{K}}(\hat{\beta})(x-y,t)\; u(y,t)dy \\ \|w(t)\|_c^2 &= \int_0^1 e^{cx} w^2(x,t)dx \end{aligned}$$

Global stabilization & pointwise-in-space regulation

Theorem.

$\exists R, \rho > 0$ s.t.

$$\boxed{\Gamma(t) \leq R\left(\mathrm{e}^{\rho\Gamma(0)} - 1\right)} \qquad \forall t \geq 0$$

$$\Gamma(t) = \int_0^1 \left[u^2(x,t) + \left(\beta(x) - \hat{\beta}(x,t)\right)^2\right] dx$$

Global stabilization & pointwise-in-space regulation

Theorem.

$\exists R, \rho > 0$ s.t.

$$\Gamma(t) \leq R\left(e^{\rho\Gamma(0)} - 1\right) \qquad \forall t \geq 0$$

$$\Gamma(t) = \int_0^1 \left[u^2(x,t) + \left(\beta(x) - \hat{\beta}(x,t)\right)^2\right] dx$$

and

$$\lim_{t\to\infty} u(x,t) = 0 \qquad \forall x \in [0,1]$$

Global stabilization & pointwise-in-space regulation

Theorem. For all systems with $\|\beta\|_\infty \leq B$ and

- all operators $\hat{\mathcal{K}}$ trained for any $\epsilon \in \left(0, \mathcal{O}\left(\frac{1}{1+B}\right)\right)$

$\exists R, \rho > 0$ s.t.

$$\Gamma(t) \leq R\left(e^{\rho\Gamma(0)} - 1\right) \qquad \forall t \geq 0$$

$$\Gamma(t) = \int_0^1 \left[u^2(x,t) + \left(\beta(x) - \hat{\beta}(x,t)\right)^2\right] dx$$

and

$$\lim_{t\to\infty} u(x,t) = 0 \qquad \forall x \in [0,1]$$

Global stabilization & pointwise-in-space regulation

Theorem. For all systems with $\|\beta\|_\infty \le B$ and

- all operators $\hat{\mathcal{K}}$ trained for any $\epsilon \in \left(0, \mathcal{O}\left(\frac{1}{1+B}\right)\right)$
- all adaptation gains $\gamma \in \left(0, \mathcal{O}\left(\frac{1}{1+B}\right)\right)$

$\exists R, \rho > 0$ s.t.

$$\Gamma(t) \le R\left(e^{\rho\Gamma(0)} - 1\right) \qquad \forall t \ge 0$$

$$\Gamma(t) = \int_0^1 \left[u^2(x,t) + \left(\beta(x) - \hat{\beta}(x,t)\right)^2\right] dx$$

and

$$\lim_{t\to\infty} u(x,t) = 0 \qquad \forall x \in [0,1]$$

Proof of Theorem

Perturbed target system

with $w(1,t) = 0$

$$
\begin{aligned}
w_t = w_x & - \left[\left(1 - \hat{\beta}*\right)\left(\mathcal{K}\left(\hat{\beta}\right) - \hat{\mathcal{K}}\left(\hat{\beta}\right)\right)\right] w(0,t) && \text{gain approximation error} \\
& + \left[\left(1 - \hat{k}*\right)\left(\beta - \hat{\beta}\right)\right] w(0,t) && \text{parameter estimation error} \\
& - \left[\left(1 - \mathcal{K} \circ \hat{\mathcal{K}}\left(\hat{\beta}\right) *\right)\right] \partial_t \left(\hat{\mathcal{K}}\left(\hat{\beta}\right)\right) * w && \text{parameter update rate perturbation}
\end{aligned}
$$

Proof of Theorem

Perturbed target system with $w(1,t)=0$

$$
\begin{aligned}
w_t \quad = \quad & w_x \\
& - \left[\left(1-\hat{\beta}*\right)\left(\mathcal{K}\left(\hat{\beta}\right) - \hat{\mathcal{K}}\left(\hat{\beta}\right)\right)\right] w(0,t) \\
& + \left[\left(1-\hat{k}*\right)\left(\beta - \hat{\beta}\right)\right] w(0,t) \\
& - \left[\left(1-\mathcal{K}\circ\hat{\mathcal{K}}\left(\hat{\beta}\right)*\right)\right] \partial_t\left(\hat{\mathcal{K}}\left(\hat{\beta}\right)\right) * w
\end{aligned}
$$

gain approximation error

parameter estimation error

parameter **update** **rate** perturbation

Proof of Theorem ★

Perturbed target system

with $w(1,t) = 0$

$$
\begin{aligned}
w_t \;=\; & w_x \\
& - \left[\left(1 - \hat{\beta} *\right) \left(\mathcal{K}\left(\hat{\beta}\right) - \hat{\mathcal{K}}\left(\hat{\beta}\right) \right) \right] w(0,t) \\
& + \left[\left(1 - \hat{k} *\right) \left(\beta - \hat{\beta}\right) \right] w(0,t) \\
& - \left[\left(1 - \mathcal{K} \circ \hat{\mathcal{K}}\left(\hat{\beta}\right) *\right) \right] \partial_t \left(\hat{\mathcal{K}}\left(\hat{\beta}\right) \right) * w
\end{aligned}
$$

gain approximation error

parameter estimation error

parameter **update** **rate** perturbation

Proof of Theorem

★

Perturbed target system

with $w(1, t) = 0$

$$
\begin{aligned}
w_t \quad = \quad & w_x \\
& - \left[\left(1 - \hat{\beta} *\right) \left(\mathcal{K}\left(\hat{\beta}\right) - \hat{\mathcal{K}}\left(\hat{\beta}\right) \right) \right] w(0, t) \\
& + \left[\left(1 - \hat{k} *\right) \left(\beta - \hat{\beta} \right) \right] w(0, t) \\
& - \left[\left(1 - \mathcal{K} \circ \hat{\mathcal{K}}\left(\hat{\beta}\right) *\right) \right] \partial_t \left(\hat{\mathcal{K}}\left(\hat{\beta}\right) \right) * w
\end{aligned}
$$

gain approximation error

parameter estimation error

parameter **update** **rate** perturbation

Proof

Key inequality (for handling update rate perturbation)

$$\left\| \partial_t \left(\hat{\mathcal{K}} \left(\hat{\beta} \right) \right) \right\| \leq \epsilon + M(B) \left\| \partial_t \hat{\beta} \right\|$$

Ensured by operator definition and training.

Proof

Lyapunov functional

$$V(t) \;=\; \ln\left(1 + \|w(t)\|_c^2\right) + \frac{1}{\gamma}\left\|\beta - \hat{\beta}(t)\right\|^2$$

Parabolic PDEs

(advancing to operators btw fcns of **two** variables)

Reaction-diffusion **PDE** (unstable)

$$u_t(x,t) \;=\; u_{xx}(x,t) + \lambda(x)u(x,t)$$

Reaction-diffusion **PDE** (unstable)

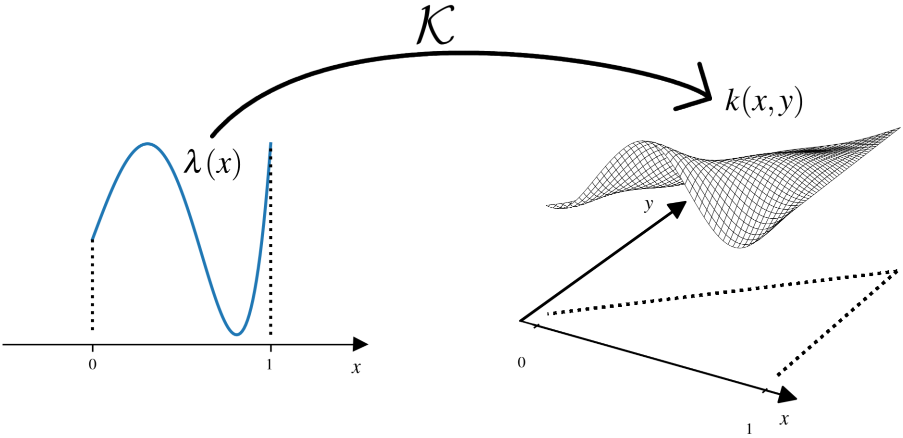
$$u_t(x,t) = u_{xx}(x,t) + \lambda(x)u(x,t)$$

Bkst transform

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

Kernel PDE: $k_{xx}(x,y) - k_{yy}(x,y) = \lambda(y)\, k(x,y)$

Bkst gain operator:



Adaptive Control of Parabolic PDEs

Adaptive Control with DeepONet implementation

$$
\begin{aligned}
\lambda(x) &= \text{\textit{unknown function}} \\
\hat{\lambda}(x,t) &= \text{online-updated estimate}
\end{aligned}
$$

Adaptive Control with DeepONet implementation

$$\lambda(x) = \text{unknown function}$$

$$\hat{\lambda}(x,t) = \text{online-updated estimate}$$

$$U(t) = \int_0^1 \hat{\mathcal{K}}(\hat{\lambda})(1,y,t)\, u(y,t)dy$$

$\hat{\mathcal{K}}(\hat{\lambda})$ takes ~ 1 millisecond to re-evaluate at each timestep in real time on an old laptop

Update law

$$\frac{\partial}{\partial t}\hat{\lambda}(x,t) = \underbrace{\frac{\gamma}{1+\|w(t)\|^2}}_{\text{normalization}} \underbrace{\left[w(x,t) - \int_x^1 \hat{\mathcal{K}}(\hat{\lambda})(y,x,t)\, w(y,t)dy\right]}_{\text{regressor}} \underbrace{u(x,t)}_{\text{regulation error}}$$

where

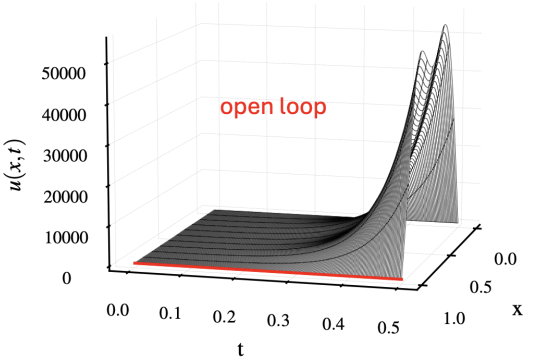
$$w(x,t) = u(x,t) - \int_0^x \hat{\mathcal{K}}(\hat{\lambda})(x,y,t)\, u(y,t)dy$$

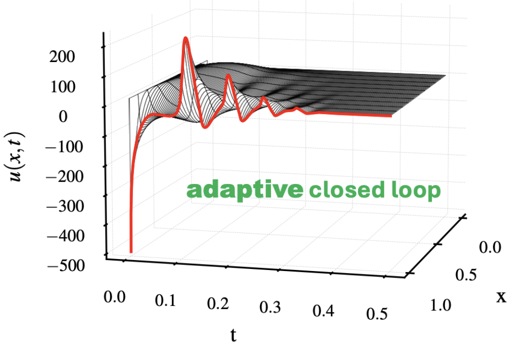
Update law

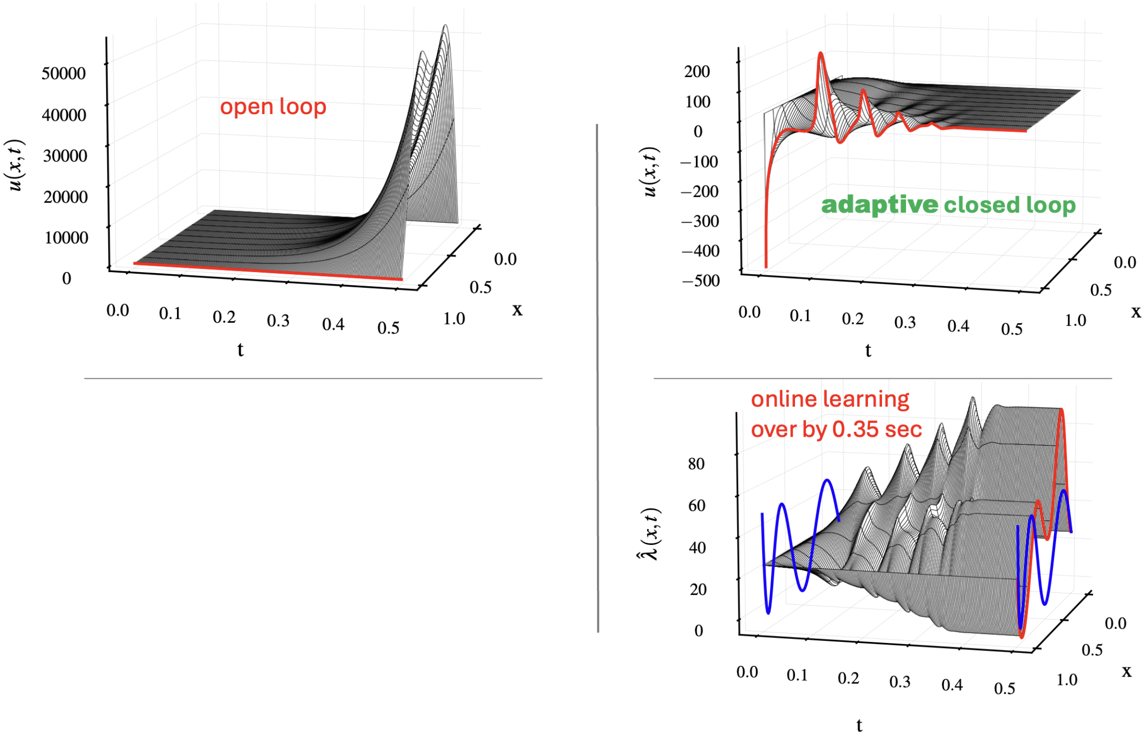
$$\frac{\partial}{\partial t}\hat{\lambda}(x,t) = \underbrace{\frac{\gamma}{1+\|w(t)\|^2}}_{\text{normalization}} \underbrace{\left[w(x,t) - \int_x^1 \hat{\mathcal{K}}(\hat{\lambda})(y,x,t)\, w(y,t)dy\right]}_{\text{regressor}} \underbrace{u(x,t)}_{\substack{\text{regulation}\\ \text{error}}}$$

where

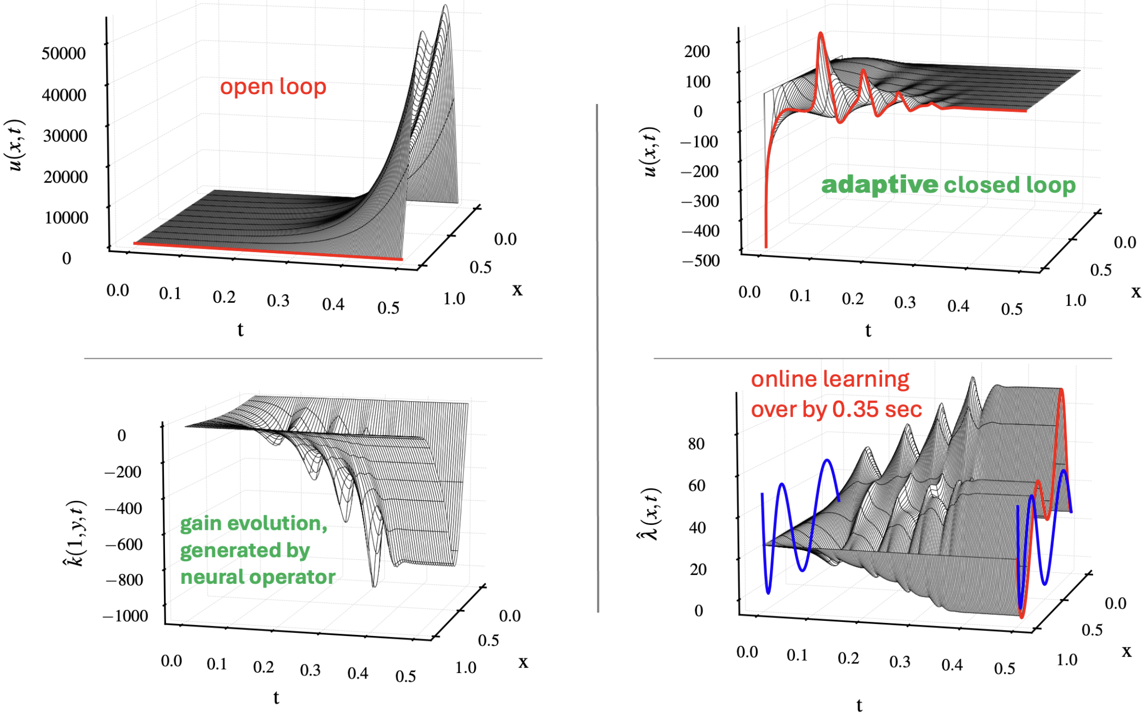
$$w(x,t) \quad = \quad u(x,t) - \int_0^x \hat{\mathcal{K}}(\hat{\lambda})(x,y,t)\, u(y,t)dy$$


open loop
50000
40000
30000
20000
10000
0
u(x,t)
0.0
0.1
0.2
0.3
0.4
0.5
t
0.0
0.5
1.0
x


adaptive closed loop
200
100
0
−100
−200
−300
−400
−500
u(x,t)
0.0
0.1
0.2
0.3
0.4
0.5
t
0.0
0.5
1.0
x



open loop
$u(x,t)$
50000
40000
30000
20000
10000
0
0.0
0.1
0.2
0.3
0.4
0.5
t
0.0
0.5
1.0
x
adaptive closed loop
$u(x,t)$
200
100
0
−100
−200
−300
−400
−500
0.0
0.1
0.2
0.3
0.4
0.5
t
0.0
0.5
1.0
x
online learning
over by 0.35 sec
$\hat{\lambda}(x,t)$
80
60
40
20
0
0.0
0.1
0.2
0.3
0.4
0.5
t
0.0
0.5
1.0
x


open loop
$u(x,t)$
50000
40000
30000
20000
10000
0
0.0
0.1
0.2
0.3
0.4
0.5
t
0.0
0.5
1.0
x
adaptive closed loop
200
100
0
−100
−200
−300
−400
−500
gain evolution,
generated by
neural operator
$\hat{k}(1,y,t)$
−1000
−800
−600
−400
online learning
over by 0.35 sec
$\hat{\lambda}(x,t)$
80
60
40
20

PERTURBED target system ★

$$
\begin{aligned}
w_t(x,t) &= w_x(x,t) \\
&\quad -2\frac{d}{dt}\left(\mathcal{K}\left(\hat\lambda\right)(x,x,t) - \hat{\mathcal{K}}\left(\hat\lambda\right)(x,x,t)\right)u(x,t) \\
&\quad -\int_0^x \left(\partial_{xx} - \partial_{yy} - \hat\lambda(y,t)\right)\left(\mathcal{K}\left(\hat\lambda\right)(x,y,t) - \hat{\mathcal{K}}\left(\hat\lambda\right)(x,y,t)\right)u(y,t)dy
\end{aligned}
$$

gain approximation error

$$
+\left(\lambda(x) - \hat\lambda(x,t)\right)u(x,t) - \int_0^x \left(\lambda(y) - \hat\lambda(y,t)\right)\hat{\mathcal{K}}\left(\hat\lambda\right)(x,y,t)\,u(y,t)dy
$$

param. estim. error

$$
-\int_0^x \partial_t\left(\hat{\mathcal{K}}\left(\hat\lambda\right)\right)(x,y,t)\,u(y,t)dy
$$

param. update rate perturbation

Global stabilization & pointwise-in-space regulation

Theorem. $\exists R, \rho > 0$ s.t.

$$\boxed{\Gamma(t) \leq R\left(e^{\rho\Gamma(0)} - 1\right)} \qquad \forall t \geq 0$$

$$\Gamma(t) = \int_0^1 \left[u^2(x,t) + \left(\lambda(x) - \hat{\lambda}(x,t)\right)^2\right] dx$$

and

$$\boxed{\lim_{t\to\infty} u(x,t) = 0} \qquad \forall x \in [0,1]$$

Recap

- Speedup in producing PDE gains $\sim 1000\times$
- Training price? Minutes.
- **Enabled:** adaptive control of PDEs with unknown parameters

Future?

Generalizations to:

- 2D, 3D
- coupled + ensemble PDEs
- **applications**

NEXT TOPIC — *Population* Dynamics

“Aging” **predator-prey**:

NEXT TOPIC — *Population* Dynamics

"Aging" **predator-prey**:

$$\frac{\partial x_1(a,t)}{\partial t} = -\frac{\partial x_1(a,t)}{\partial a} - \left(\int_0^A g_1(\alpha) x_2(\alpha,t)\,d\alpha + \boxed{u(t)}\right) x_1(a,t)$$

predator kills prey

$$\frac{\partial x_2(a,t)}{\partial t} = -\frac{\partial x_2(a,t)}{\partial a} - \left(\frac{1}{\int_0^A g_2(\alpha) x_1(\alpha,t)\,d\alpha} + \boxed{u(t)}\right) x_2(a,t)$$

prey nourishes predator

NEXT TOPIC — *Population* Dynamics

"Aging" **predator-prey**:

$$\frac{\partial x_1(a,t)}{\partial t} = -\frac{\partial x_1(a,t)}{\partial a} - \left(\int_0^A g_1(\alpha) x_2(\alpha,t)\, d\alpha + \boxed{u(t)}\right) x_1(a,t)$$

predator kills prey

$$\frac{\partial x_2(a,t)}{\partial t} = -\frac{\partial x_2(a,t)}{\partial a} - \left(\frac{1}{\int_0^A g_2(\alpha) x_1(\alpha,t)\, d\alpha} + \boxed{u(t)}\right) x_2(a,t)$$

prey nourishes predator

Birth boundary conditions:

$$x_1(0,t) = \int_0^A k_1(a) x_1(a,t)\, da$$

$$x_2(0,t) = \int_0^A k_2(a) x_2(a,t)\, da$$

NEXT TOPIC — *Population* Dynamics

"Aging" **predator-prey**:

$$\frac{\partial x_1(a,t)}{\partial t} = -\frac{\partial x_1(a,t)}{\partial a} - \left(\int_0^A g_1(\alpha) x_2(\alpha,t)\,d\alpha + \boxed{u(t)}\right) x_1(a,t)$$

predator kills prey

$$\frac{\partial x_2(a,t)}{\partial t} = -\frac{\partial x_2(a,t)}{\partial a} - \left(\frac{1}{\int_0^A g_2(\alpha) x_1(\alpha,t)\,d\alpha} + \boxed{u(t)}\right) x_2(a,t)$$

prey nourishes predator

Birth boundary conditions:

$$x_1(0,t) = \int_0^A k_1(a) x_1(a,t)\,da$$

$$x_2(0,t) = \int_0^A k_2(a) x_2(a,t)\,da$$

ecology • epidemiology • opinion dynamics • amortization of assets (in DoD)

NEXT TOPIC — *Population* Dynamics

"Aging" **predator-prey**:

$$\frac{\partial x_1(a,t)}{\partial t} = -\frac{\partial x_1(a,t)}{\partial a} - \left(\int_0^A g_1(\alpha) x_2(\alpha,t) d\alpha + \boxed{u(t)}\right) x_1(a,t)$$

predator <u>kills</u> prey

$$\frac{\partial x_2(a,t)}{\partial t} = -\frac{\partial x_2(a,t)}{\partial a} - \left(\frac{1}{\int_0^A g_2(\alpha) x_1(\alpha,t) d\alpha} + \boxed{u(t)}\right) x_2(a,t)$$

prey <u>nourishes</u> predator

Birth boundary conditions:

$$x_1(0,t) = \int_0^A k_1(a) x_1(a,t) da$$

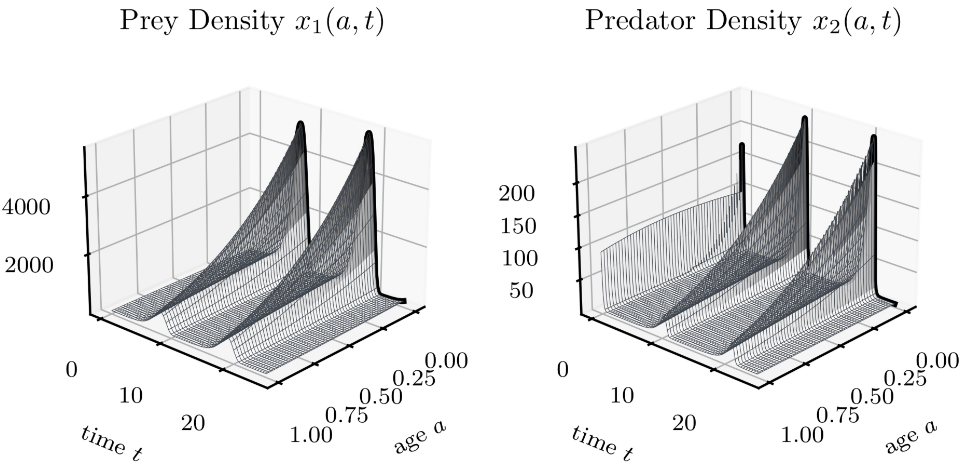
$$x_2(0,t) = \int_0^A k_2(a) x_2(a,t) da$$

ecology • epidemiology • opinion dynamics • amortization of assets (in DoD)

Hard: <u>common input</u>

Open loop — Oscillation

(an order of magnitude)



Closed loop — Settle to setpoint

(overpopulation transient necessitated by control positivity constraint)