# Convex Controller Synthesis
for Evolution Equations

**Lauren Conger**
California Institute of Technology
Control and Dynamical Systems
Computing and Mathematical Sciences

# **Convex Controller Synthesis**
# for Evolution Equations

American Control Conference 2025

Franca Hoffmann
Caltech (co-advisor*)

Antoine Leeman
ETH Zurich

Institute for
Dynamic Systems and Control
IDSC
Institut für Dynamische Systeme
und Regelungstechnik

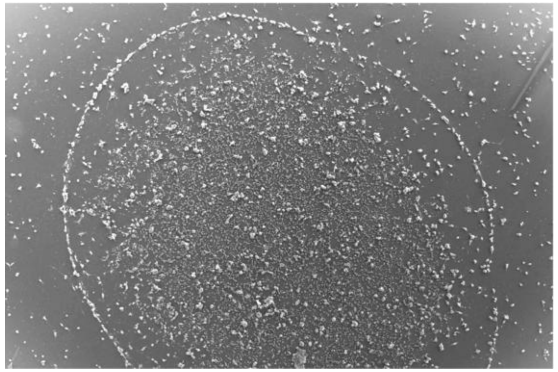
*Co-advisors: Franca Hoffmann, Eric Mazumdar, John Doyle.

Caltech

# Control and Analysis in Infinite Dimensions

Multi-particle system

Multi-cell system

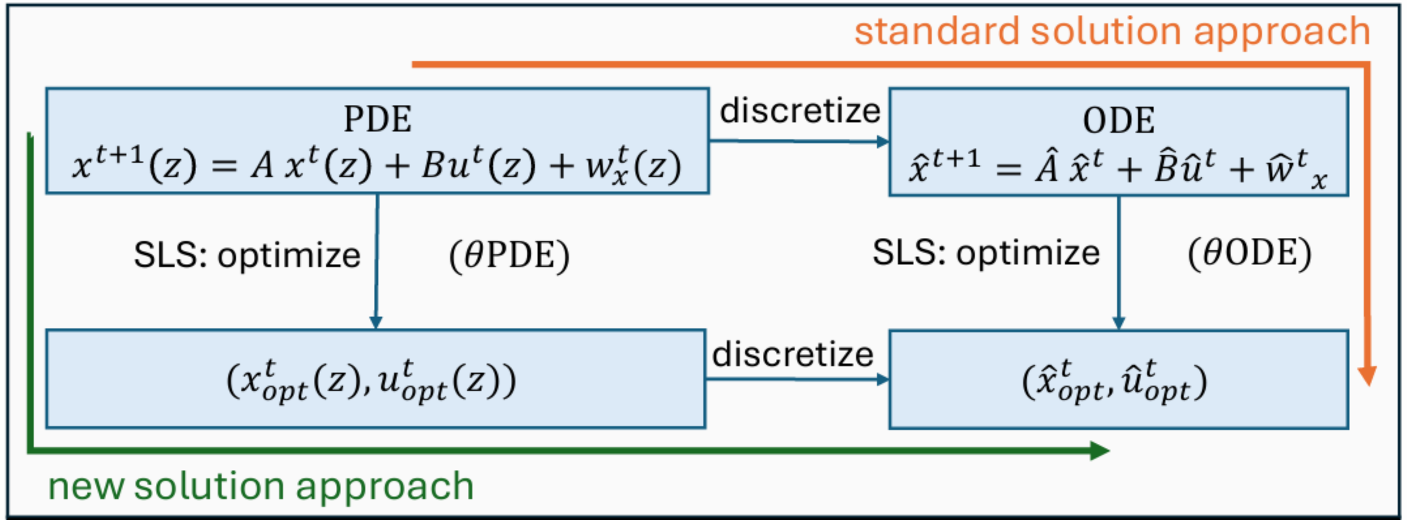


Multi-agent system

Modeling at the continuous level captures global structure. Not always feasible to model at particle level.
⇒ **Discretize PDEs, integral equations in function space.**

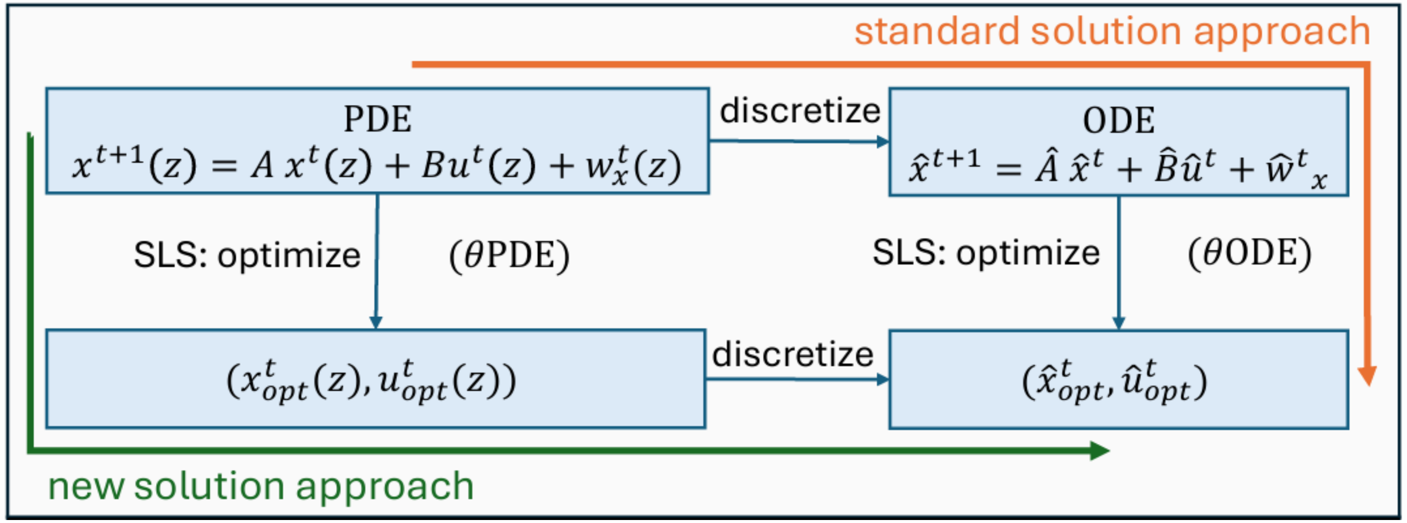
# Outline

1. Introduction
2. Discretize-then-optimize system level synthesis
3. Optimize-then-discretize controller synthesis
4. Implementation and numerical results



# Outline

1. Introduction
2. Discretize-then-optimize system level synthesis
3. Optimize-then-discretize controller synthesis
4. Implementation and numerical results



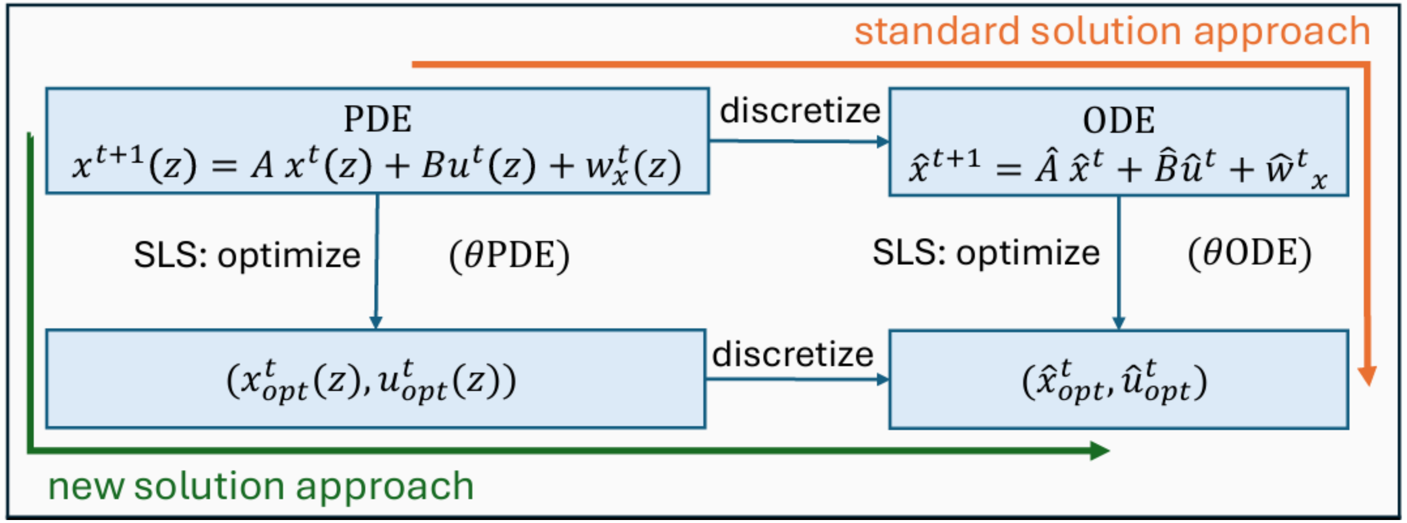
# Related Research

- Discretize-then-optimize does not commute [Liu and Wang, 2019]
- Continuous space and time with SLS proposed in [Jensen, 2020]
- Boundary control via Backstepping [Smyshlyaev and Krstic, 2010, Ascencio et al., 2018]
- System specifics [Tröltzsch, 2010], e.g., reaction-diffusion [Ayamou et al., 2024, Si et al.,2018, Vazquez and Krstic, 2019]
- Theoretical control in Hilbert space: adaptive control [Wen and Balas, 1989], robust control [Venkatesh et al., 2000], controllability [Slemrod, 1974], and stabilizing operators [Gibson, 1979]

Need practical and principled general-purpose tools for (uncertain) linear PDEs!

# Outline

1. Introduction
2. Discretize-then-optimize system level synthesis
3. Optimize-then-discretize controller synthesis
4. Implementation and numerical results



# Discretization Example

Consider the system[1] with full state measurements $x^t \in L^2(\Omega)$, $u^t \in \mathbb{R}^{n_u}$, parameterized by $a \in L^2(\Omega)$ and $b: \Omega \to \mathbb{R}^{n_u}$,

$$x^{t+1}(z) = \underbrace{\int a(z - z')x^t(z')dz'}_{Ax^t} + \underbrace{\sum_{l=1}^{n_u} b(l,z)u_l^t}_{Bu^t}.$$

After discretization, it yields

$$\hat{x}^{t+1} = \hat{A}\hat{x}^t + \hat{B}\hat{u}^t, \qquad \hat{x}^t \in \mathbb{R}^{n_x}, \qquad \hat{u}^t \in \mathbb{R}^{n_u}.$$

[1]The linear Boltzmann dynamics is used to model, e.g., bacterial movement [Perthame, 2007, Sec 5.6]

# Output feedback system level synthesis

We have a system

$$x^{t+1} = Ax^t + Bu^t + w^t$$
$$y^t = Cx^t + v^t$$

which we write compactly over the time horizon

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{w}$$
$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{v}.$$

$$\boldsymbol{x} = \begin{bmatrix} x^0 \\ \vdots \\ x^T \end{bmatrix}, \boldsymbol{u} = \begin{bmatrix} u^0 \\ \vdots \\ u^T \end{bmatrix}, \boldsymbol{w} = \begin{bmatrix} x^0 \\ w^0 \\ \vdots \\ w^{T-1} \end{bmatrix}, \boldsymbol{v} = \begin{bmatrix} v^0 \\ \vdots \\ v^T \end{bmatrix},$$

$$\boldsymbol{A} = \begin{bmatrix} 0 & & & \\ A & \ddots & & \\ & & \ddots & \\ & & A & 0 \end{bmatrix}, \text{ etc.}$$

We introduce an output feedback controller

$$\boldsymbol{u} = \boldsymbol{K}\boldsymbol{y}.$$

Rearranging the dynamics,

$$\boldsymbol{x} = (\boldsymbol{I} - \boldsymbol{Z}\boldsymbol{A} - \boldsymbol{Z}\boldsymbol{B}\boldsymbol{K}\boldsymbol{C})^{-1}\boldsymbol{w} + (\boldsymbol{I} - \boldsymbol{Z}\boldsymbol{A} - \boldsymbol{Z}\boldsymbol{B}\boldsymbol{K}\boldsymbol{C})^{-1}\boldsymbol{Z}\boldsymbol{B}\boldsymbol{K}\boldsymbol{v}$$
$$\boldsymbol{u} = \boldsymbol{K}\boldsymbol{C}\boldsymbol{x} + \boldsymbol{K}\boldsymbol{v}$$

Define closed-loop maps

$$\Phi_{xx} = (\boldsymbol{I} - \boldsymbol{Z}\boldsymbol{A} - \boldsymbol{Z}\boldsymbol{B}\boldsymbol{K}\boldsymbol{C})^{-1}$$
$$\Phi_{xy} = (\boldsymbol{I} - \boldsymbol{Z}\boldsymbol{A} - \boldsymbol{Z}\boldsymbol{B}\boldsymbol{K}\boldsymbol{C})^{-1}\boldsymbol{Z}\boldsymbol{B}\boldsymbol{K}$$
$$\Phi_{ux} = \mathbf{KC}\Phi_{xx}$$
$$\Phi_{uy} = \boldsymbol{K} + \boldsymbol{K}\boldsymbol{C}\Phi_{xx}\boldsymbol{Z}\boldsymbol{B}\boldsymbol{K}$$

Dynamics in terms of closed-loop maps:

$$\boldsymbol{x} = \Phi_{xx}\boldsymbol{w} + \Phi_{xy}\boldsymbol{v}$$
$$\boldsymbol{u} = \Phi_{ux}\boldsymbol{w} + \Phi_{uy}\boldsymbol{v}$$

Anderson, Doyle, Low, Matni. System Level Synthesis, 2019.

# Output feedback system level synthesis

We have a system

$$x^{t+1} = Ax^t + Bu^t + w^t$$
$$y^t = Cx^t + v^t$$

which we write compactly over the time horizon

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{w}$$
$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{v}.$$

$$\boldsymbol{x} = \begin{bmatrix} x^0 \\ \vdots \\ x^T \end{bmatrix}, \boldsymbol{u} = \begin{bmatrix} u^0 \\ \vdots \\ u^T \end{bmatrix}, \boldsymbol{w} = \begin{bmatrix} x^0 \\ w^0 \\ \vdots \\ w^{T-1} \end{bmatrix}, \boldsymbol{v} = \begin{bmatrix} v^0 \\ \vdots \\ v^T \end{bmatrix},$$

$$\boldsymbol{A} = \begin{bmatrix} 0 & & & \\ A & \ddots & & \\ & & \ddots & \\ & & A & 0 \end{bmatrix}, \text{ etc.}$$

We introduce an output feedback controller

$$\boldsymbol{u} = \boldsymbol{K}\boldsymbol{y}.$$

Rearranging the dynamics,

$$\boldsymbol{x} = (\boldsymbol{I} - \boldsymbol{Z}\boldsymbol{A} - \boldsymbol{Z}\boldsymbol{B}\boldsymbol{K}\boldsymbol{C})^{-1}\boldsymbol{w} + (\boldsymbol{I} - \boldsymbol{Z}\boldsymbol{A} - \boldsymbol{Z}\boldsymbol{B}\boldsymbol{K}\boldsymbol{C})^{-1}\boldsymbol{Z}\boldsymbol{B}\boldsymbol{K}\boldsymbol{v}$$
$$\boldsymbol{u} = \boldsymbol{K}\boldsymbol{C}\boldsymbol{x} + \boldsymbol{K}\boldsymbol{v}$$

Define closed-loop maps

$$\Phi_{xx} = (\boldsymbol{I} - \boldsymbol{Z}\boldsymbol{A} - \boldsymbol{Z}\boldsymbol{B}\boldsymbol{K}\boldsymbol{C})^{-1}$$
$$\Phi_{xy} = (\boldsymbol{I} - \boldsymbol{Z}\boldsymbol{A} - \boldsymbol{Z}\boldsymbol{B}\boldsymbol{K}\boldsymbol{C})^{-1}\boldsymbol{Z}\boldsymbol{B}\boldsymbol{K}$$
$$\Phi_{ux} = \mathbf{KC}\Phi_{xx}$$
$$\Phi_{uy} = \boldsymbol{K} + \boldsymbol{K}\boldsymbol{C}\Phi_{xx}\boldsymbol{Z}\boldsymbol{B}\boldsymbol{K}$$

Dynamics in terms of closed-loop maps:

$$\boldsymbol{x} = \Phi_{xx}\boldsymbol{w} + \Phi_{xy}\boldsymbol{v}$$
$$\boldsymbol{u} = \Phi_{ux}\boldsymbol{w} + \Phi_{uy}\boldsymbol{v}$$

Anderson, Doyle, Low, Matni. System Level Synthesis, 2019.

# Output feedback system level synthesis

We have a system

$$x^{t+1} = Ax^t + Bu^t + w^t$$
$$y^t = Cx^t + v^t$$

We introduce an output feedback controller

$$\boldsymbol{u} = \boldsymbol{K}\boldsymbol{y}.$$

Dynamics in terms of closed-loop maps:

$$\boldsymbol{x} = \Phi_{xx}\boldsymbol{w} + \Phi_{xy}\boldsymbol{v}$$
$$\boldsymbol{u} = \Phi_{ux}\boldsymbol{w} + \Phi_{uy}\boldsymbol{v}$$

***Theorem (informal) [Anderson et al, 2019]***

The trajectory $(\boldsymbol{x}, \boldsymbol{u})$ of the system in closed loop with $\boldsymbol{u} = \boldsymbol{K}\boldsymbol{y}$ can be expressed as above, *if and only if* $\Phi_{xx}, \Phi_{xy}, \Phi_{ux}, \Phi_{uy}$ lie on the affine subspace

$$[\boldsymbol{I} - \boldsymbol{Z}\boldsymbol{A} \quad -\boldsymbol{Z}\boldsymbol{B}] \begin{bmatrix} \Phi_{xx} & \Phi_{xy} \\ \Phi_{ux} & \Phi_{uy} \end{bmatrix} = [\boldsymbol{I} \quad \boldsymbol{0}],$$

$$\begin{bmatrix} \Phi_{xx} & \Phi_{xy} \\ \Phi_{ux} & \Phi_{uy} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} - \boldsymbol{Z}\boldsymbol{A} \\ -\boldsymbol{C} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{0} \end{bmatrix}.$$

The controller can be computed as $\boldsymbol{u} = \left(\Phi_{uy} - \Phi_{ux}\Phi_{xx}^{-1}\Phi_{xy}\right)\boldsymbol{y}$; implementation does not require inversion of $\Phi_{xx}$ .

# Constraints

Dynamics in terms of closed-loop maps:

$$\boldsymbol{x} = \Phi_{xx}\boldsymbol{w} + \Phi_{xy}\boldsymbol{v}$$
$$\boldsymbol{u} = \Phi_{ux}\boldsymbol{w} + \Phi_{uy}\boldsymbol{v}$$

Let the adjacency matrix $G$ be defined such that $G[i,j] = 1$ if $A[i,j] \neq 0$.

(locality $r$) $\quad \text{supp}\ \Phi_x(\cdot,\cdot) = \text{supp}\ G^r,\quad \text{supp}\ \Phi_u = \text{supp}(|\mathrm{B}^\top|\text{supp}\ \Phi_\mathrm{x})$

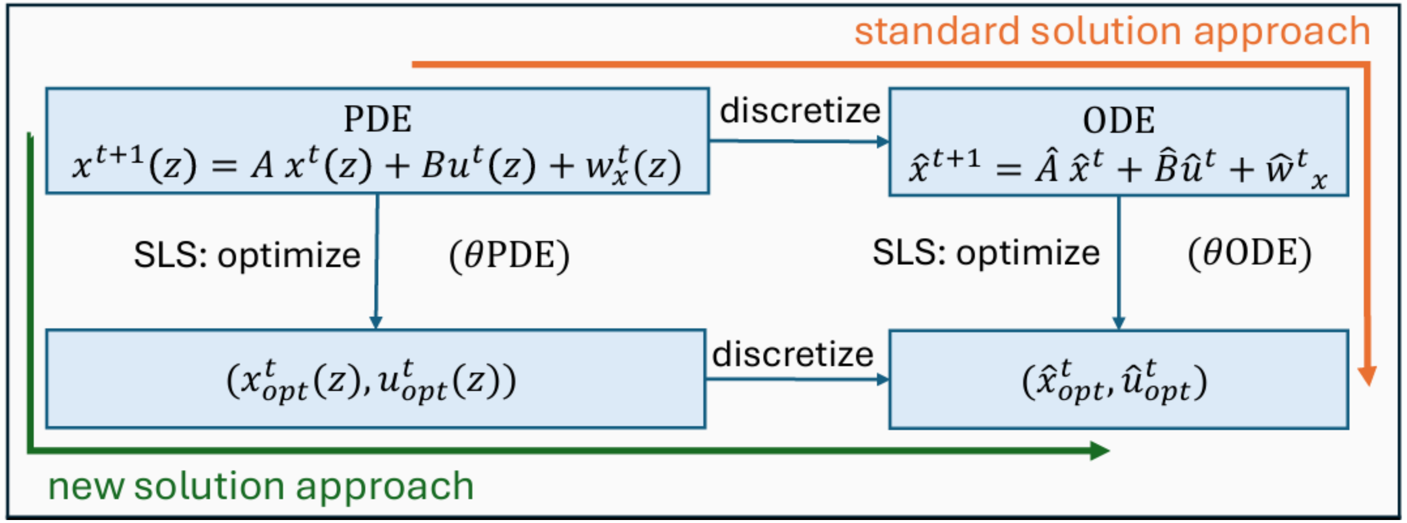
(communication speed $v$) $\quad \text{supp}\ \Phi_x(t,\cdot) = \text{supp}\ G^{\lfloor vt \rfloor},\quad \text{supp}\ \Phi_u = \text{supp}(|\mathrm{B}^\top|\text{supp}\ \Phi_\mathrm{x})$

(actuation delay $\tau$) $\quad \Phi_u(t,\cdot) = 0\ \ \forall\, t < \tau.$

⇒ **Convex constraints!**

# Outline

1. Introduction
2. Discretize-then-optimize system level synthesis
3. Optimize-then-discretize controller synthesis
4. Implementation and numerical results



# Natural Extension: *strong* form of dynamics

For Hilbert spaces $X, U, Y$, at time $t$ let

- state $x^t \in X$
- input $u^t \in U$
- output $y^t \in Y$,
- disturbance $w_x^t \in X$
- observation noise $w_y^t \in \mathrm{Y}$.

Consider the discrete-time dynamics

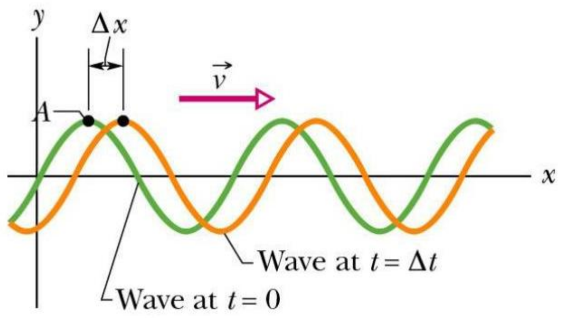
$$x^{t+1} = A^* x^t + B^* u^t + w_x^t$$
$$y^t = C^* x^t + w_y^t,$$

dynamics $A^*: D(A^*) \to X, \; D(A^*) \subseteq X,$
control $B^*: D(B^*) \to X, \; D(B^*) \subseteq U,$
observation $C^*: D(C^*) \to Y, \; D(C^*) \subseteq X.$

**Example**
Consider the advection operator

$$A^* x = I + \Delta t \left( v \cdot \nabla_z x(z) \right).$$

- Initial condition: $x^0(z) = \mathrm{sign}(z)$
- Desired solution: translation
- Bad news: step function not differentiable; physically relevant solutions not captured by this strong framework. **Need another approach!**



# Generalization: *weak* form of dynamics

Consider the dynamics

$$\langle x^{t+1}, f\rangle_X = \langle x^t, Af\rangle_X + \langle u^t, Bf\rangle_U + \langle w_x^t, f\rangle_X$$
$$\langle y^t, g\rangle_Y = \langle x^t, Cg\rangle_X + \left\langle w_y^t, g\right\rangle_Y$$

for all text functions $f \in D(A) \cap D(B) \subseteq X$ and $g \in D(C) \subseteq Y$.

Compare with strong form

$$x^{t+1} = A^* x^t + B^* u^t + w_x^t$$
$$y^t = C^* x^t + w_y^t,$$

**Test functions**
- Applying differential operators to test functions instead of solution itself allows for less regular solutions
- $(x^t, u^t)_{t=0}^T$ solves *weakly* if all test functions fulfilled.

**Finite-Dimensional Hilbert Space: Example**
- For $X = \mathbb{R}^{n_x}, \mathrm{U} = \mathbb{R}^{n_u}$ and $Y = \mathbb{R}^{n_y}$, the weak form reduces to the standard strong form.
- $X, U, Y$ are Hilbert spaces when equipped with the 2-norm.

# Linear Feedback in Hilbert Spaces

We define in the weak sense, for $t, \tau \in [0, \dots, T]$

$$\langle u^t, h \rangle_U = \sum_{\tau=0}^{t} \left\langle x^\tau, K_x^{t,t-\tau} h \right\rangle_X \text{ for state feedback}$$

$$\langle u^t, h \rangle_U = \sum_{\tau=0}^{t} \left\langle y^\tau, K_y^{t,t-\tau} h \right\rangle_Y \text{ for output feedback}$$

for all test functions $h \in \cap_{\tau=0}^{t} D\left(K_{x,y}^{t,t-\tau}\right) \subseteq U$, with family of operators

- $K_x^{t,\tau}: D\left(K_x^{t,\tau}\right) \to X$,
- $K_y^{t,\tau}: D(K_y^{t,\tau}) \to Y$.

Compare with strong form,

$$u^t = \sum_{\tau=0}^{t} (K_x^{t,t-\tau})^* \, x^\tau \text{ for state feedback}$$

$$u^t = \sum_{\tau=0}^{t} (K_y^{t,t-\tau})^* \, y^\tau \text{ for output feedback.}$$

# System Level Parameterization

**Infinite Dimension**
The maps $\theta_x$ and $\theta_u$ parameterize the trajectories $x \in \mathcal{X}$ and $u \in \mathcal{U}$ via

$$\langle x, f\rangle_{\mathcal{X}} = \langle w_x, \theta_x f\rangle_{\mathcal{X}} \quad \forall\, f \in D(\theta_x) \subseteq \mathcal{X}$$
$$\langle u, f\rangle_{\mathcal{U}} = \langle w_x, \theta_u h\rangle_{\mathcal{X}} \quad \forall\, h \in D(\theta_u) \subseteq \mathcal{U},$$

with the controller $K_x$ given by

$$\langle f, \theta_x^{-1}\theta_u h\rangle_{\mathcal{X}} = \langle f, K_x h\rangle_{\mathcal{X}}$$

for appropriate test functions $f, h$.

**Finite Dimension**
The maps $\Phi_{xx}$ and $\Phi_{ux}$ parameterize the trajectories $\boldsymbol{x}$ and $\boldsymbol{u}$ via

$$\boldsymbol{x} = \Phi_{xx}\boldsymbol{w}$$
$$\boldsymbol{u} = \Phi_{ux}\boldsymbol{w}$$

with the controller $\boldsymbol{K_x}$ given by

$$\boldsymbol{K_x} = \Phi_{xx}^{-1}\Phi_{ux}.$$

# Theorem 1: State Feedback

Consider the trajectories

$$\langle x, f\rangle_{\mathcal{X}} = \langle w_x, \theta_x f\rangle_{\mathcal{X}} \quad \forall\, f \in D(\theta_x) \subseteq \mathcal{X}$$
$$\langle u, f\rangle_{\mathcal{U}} = \langle w_x, \theta_u h\rangle_{\mathcal{X}} \quad \forall\, h \in D(\theta_u) \subseteq \mathcal{U}. \qquad (*)$$

**Theorem (SLP-SF)**
Fix disturbance function realization $w_x \in \mathcal{X}$ and operators $\mathcal{A}, \mathcal{B}$.

i. If $K_x$ is given, then any trajectory $(x, u) \in \mathcal{X} \times \mathcal{U}$ satisfying the closed-loop dynamics also satisfies (*) with some causal closed-loop maps satisfying

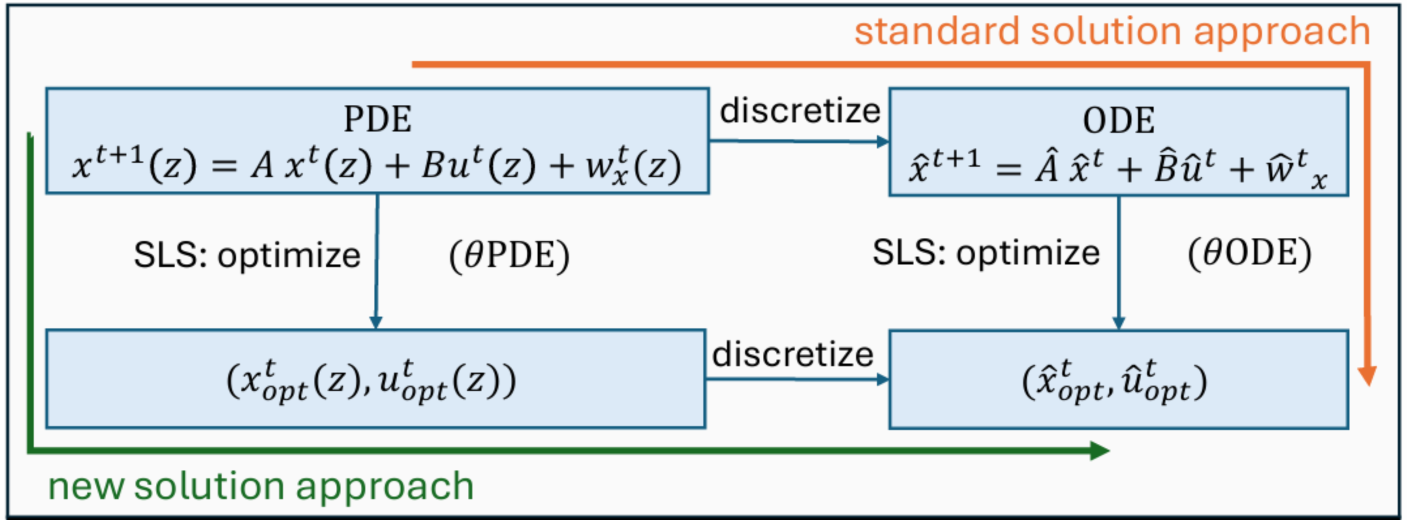
$$\left\langle f, \theta_x \hat{f}\right\rangle_{\mathcal{X}} = \left\langle f, \theta_x \mathcal{A} \hat{f}\right\rangle_{\mathcal{X}} + \left\langle f, \theta_u \mathcal{B} \hat{f}\right\rangle_{\mathcal{X}} + \left\langle f, \hat{f}\right\rangle_{\mathcal{X}} \text{ for all } f \in \mathcal{X}, \hat{f} \in D(\mathcal{A}) \cap D(\mathcal{B}). \qquad \text{(SLP-SF)}$$

ii. Let $\theta_x, \theta_u$ be arbitrary causal maps satisfying (SLP-SF). The trajectory $(x, u) \in \mathcal{X} \times \mathcal{U}$ computed with (*) also satisfies the closed-loop dynamics with the controller $K_x$ defined by $K_x \coloneqq \theta_x^{-1}\, \theta_u$ and $D(K_x) \coloneqq D(\theta_u)$.

⇒ Space of controllers parameterized by $K_x$ is equivalent to space parameterized by $(\theta_x, \theta_u)$.

# Outline

1. Introduction
2. Discretize-then-optimize system level synthesis
3. Optimize-then-discretize controller synthesis
4. Implementation and numerical results



# Implementation

**Structural Constraints**
As in finite-dimensional SLS, when restricting to a constraint set $S$, the optimization problem is

$$\min_{\theta \in S} J(\theta) \ \text{ such that (SLS} - \text{SF) holds,}$$

e.g. locality constraints, delayed measurements, communication delay.

**Integral Operators via kernels**
Express operators via kernels $f \in L^2(\Omega)$ and $z \in \Omega$

$$\left(\theta_x^{t,\tau}\right)^* f = \int_\Omega \vartheta_x^{t,\tau}(\tilde{z},\cdot) f(\tilde{z}) d\tilde{z}$$

$$\left(\theta_u^{t,\tau}\right)^* f = \int_\Omega \vartheta_u^{t,\tau}(\tilde{z},\cdot) f(\tilde{z}) d\tilde{z}\,.$$

**Implementation of Integral Operators via Kernels**

- Optimizing over $\theta_x, \theta_u$ is equivalent to optimizing over kernels $\vartheta \in M$.
- We use a real Fourier basis for $\vartheta$.
- *Apply constraints to basis functions.*

# Numerical Example

Consider the dynamics

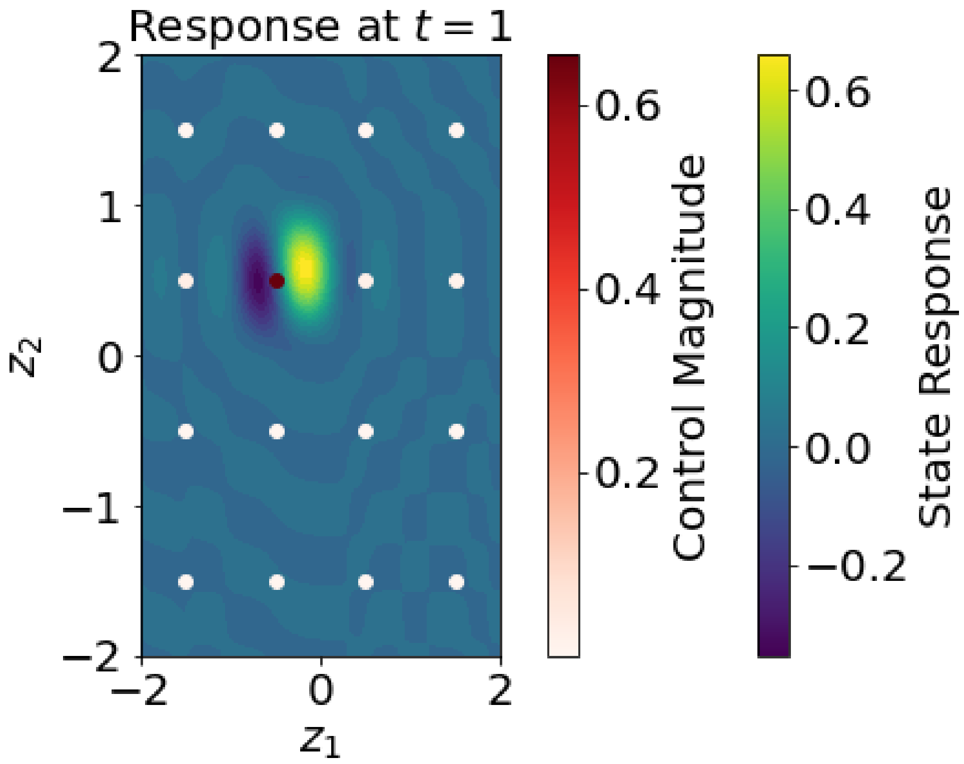
$$x^{t+1}(z) = \int a(z-z')x^t(z')dz' + \sum_{l=1}^{n_u} b(l,z)u_l^t\,.$$

Define the cost function

$$J(\vartheta) \coloneqq \sum_{t,\tau} Q \iint \left|\vartheta_x^{t,\tau}(\tilde{z},z)\right|^2 d\tilde{z}\,dz + R\int \left\|\vartheta_u^{t,\tau}(\tilde{z},\cdot)\right\|_2^2 d\tilde{z}\,,$$

For scalars $Q > 0$ and $R \geq 0$, *analogous to LQR* for finite-dimensional SLS.

- Time horizon $T = 5$
- Disturbance position (−0.26,0.56)
- Basis functions $k = 12$



# Simulation Results

1) Relative error, performance gain (relative to no control):

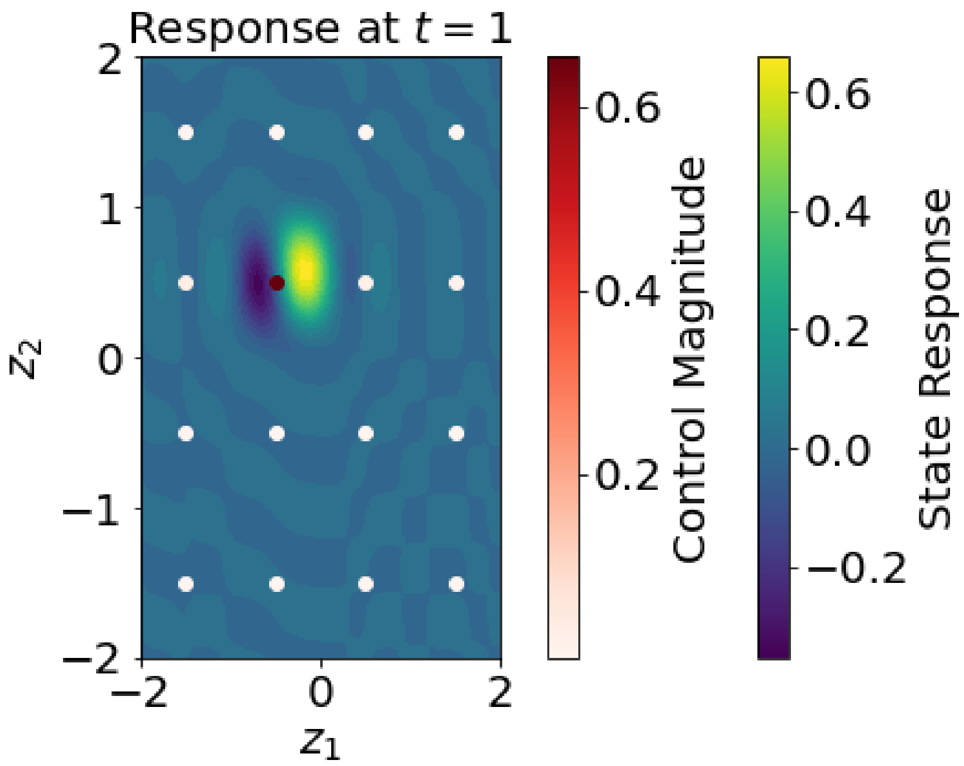
| time step | error (%) | perf. gain (%) |
|---|---|---|
| 1 | 0.16 | 42.23 |
| 2 | 0.11 | 61.23 |
| 3 | 0.17 | 70.93 |
| 4 | 0.21 | 77.39 |
| 5 | 0.23 | 82.16 |

⇒ small relative error
⇒ faster convergence to zeros relative to no control

2) Performance comparison with finite-dimensional SLS:

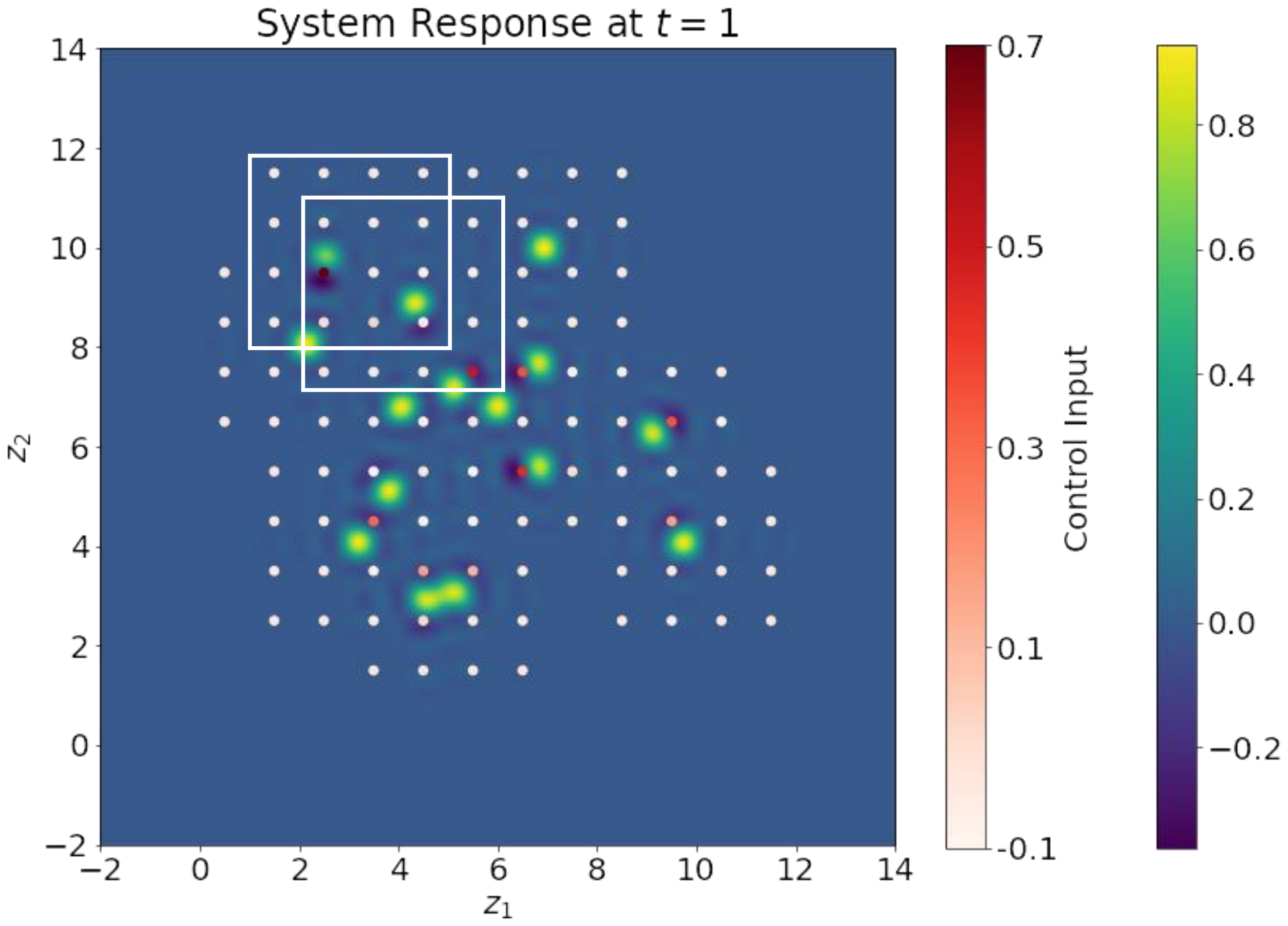
| discretization step | avg perf. gain (%) | state dimension $n_x$ |
|---|---|---|
| continuous (our approach) | 42.79 | / |
| $dx = 0.1$ | 32.26 | 1600 |
| $dx = 0.2$ | 30.54 | 400 |
| $dx = 0.25$ | 31.91 | 256 |
| $dx = 0.5$ | 37.36 | 64 |

⇒ scalability independent of discretization
⇒ higher performance compared to (finite-dimensional) SLS



# Parallel Computation and Constraints

Example: **Constrain** to allow only local controllers to respond to disturbances, compute responses in **parallel**.



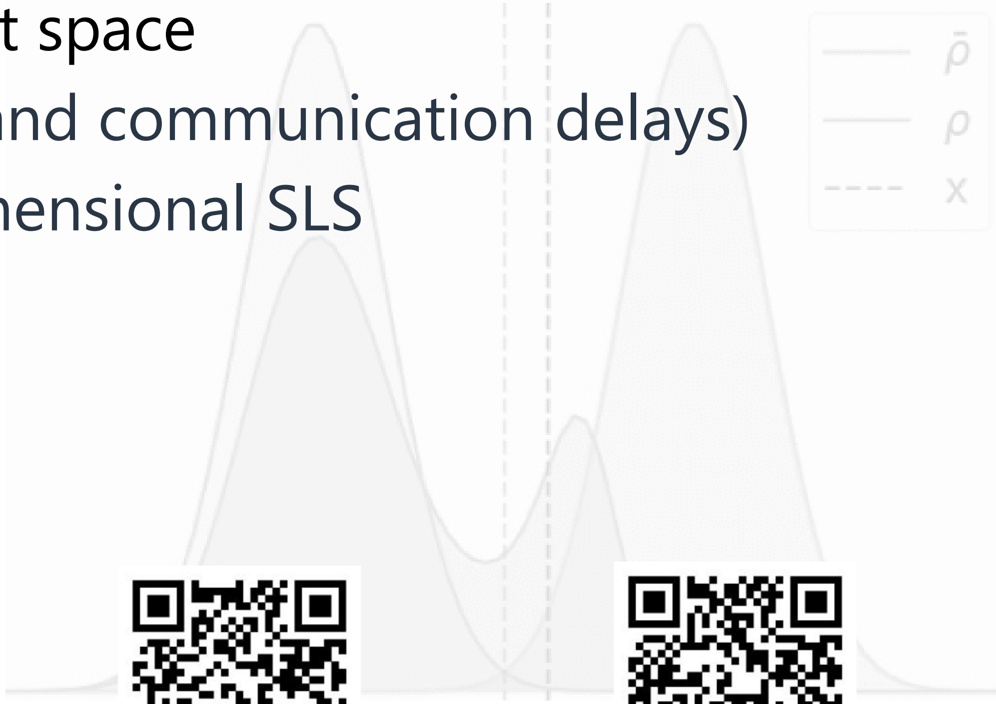
# Contributions, Future Work

✓Extension of SLS to infinite-dimensional Hilbert space

✓Convex structural constraints (locality, sensor and communication delays)

✓Improved performance compared to finite-dimensional SLS

- ❑ **Controllability and observability**
- ❑ Time-varying operators, continuous time
- ❑ Robustness guarantees




paper code