Multiscale Feedback Control Synthesis for Interacting Particle Systems

Dante Kalise

Department of Mathematics
Imperial College London

Numerical Methods for Optimal Transport Problems
Mean Field Games, and Multi-agent Systems
UTFSM, Valparaíso, January 9, 2024

Engineering and Physical Sciences Research Council

Imperial College London

Multiscale Feedback Control Synthesis for Interacting Particle Systems

Dante Kalise

Department of Mathematics
Imperial College London



Giacomo Albi (Verona), Sara Bicego and Greg Pavliotis (Imperial).

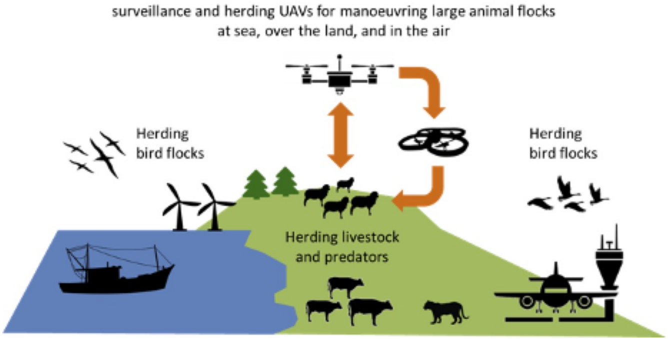
The Dynamics Universe according to Strogatz

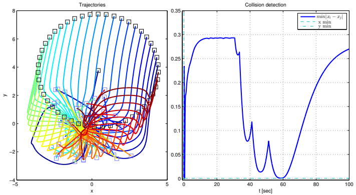
Number of variables →

Nonlinearity ↓	$n = 1$	$n = 2$	$n \geq 3$	$n >> 1$	Continuum
Linear	*Growth, decay, or equilibrium* Exponential growth RC circuit Radioactive decay	*Oscillations* Linear oscillator Mass and spring RLC circuit 2-body problem (Kepler, Newton)	Civil engineering, structures Electrical engineering	*Collective phenomena* Coupled harmonic oscillators Solid-state physics Molecular dynamics Equilibrium statistical mechanics	*Waves and patterns* Elasticity Wave equations Electromagnetism (Maxwell) Quantum mechanics (Schrödinger, Heisenberg, Dirac) Heat and diffusion Acoustics Viscous fluids
Nonlinear	Fixed points Bifurcations Overdamped systems, relaxational dynamics Logistic equation for single species	Pendulum Anharmonic oscillators Limit cycles Biological oscillators (neurons, heart cells) Predator-prey cycles Nonlinear electronics (van der Pol, Josephson)	*Chaos* Strange attractors (Lorenz) 3-body problem (Poincaré) Chemical kinetics Iterated maps (Feigenbaum) Fractals (Mandelbrot) Forced nonlinear oscillators (Levinson, Smale) Practical uses of chaos Quantum chaos ?	*The frontier* Coupled nonlinear oscillators Lasers, nonlinear optics Nonequilibrium statistical mechanics Nonlinear solid-state physics (semiconductors) Josephson arrays Heart cell synchronization Neural networks Immune system Ecosystems Economics	*Spatio-temporal complexity* Nonlinear waves (shocks, solitons) Plasmas Earthquakes General relativity (Einstein) Quantum field theory Reaction-diffusion, biological and chemical waves Fibrillation Epilepsy Turbulent fluids (Navier-Stokes) Life

The “frontier” characterizes problems with a rich behavior in both space and time.

Feedback control in collective animal behaviour





- Complex animal swarm-human technology interplays:
  - Bird flocks and aircrafts.
  - Bird flocks and wind turbines.
  - Seabirds and fish stocks.
- Learning animal collective behaviour from data.
- Identifying actions through inverse optimal control.
- Developing animal-robot optimized feedback laws.
- Swarm robotics for sensing.

A.J. King et al. *Biologically inspired herding of animal groups by robots*, Methods in Ecology and Evolution, 2023.
G. Albi, M. Bongini, E. Cristiani and D.K. *Invisible control of self-organizing agents leaving unknown environments*, SIAM J. Appl. Math., 2016.
R. Escobedo, A. Ibáñez and E. Zuazua *Optimal strategies for driving a mobile agent in a "guidance by repulsion" model*, Commun. Nonlinear Sci., 2016.
Y.P. Choi, D.K., A. Peters and J. Peszek. *A collisionless singular Cucker-Smale model with decentralized formation control* , SIAM J. Appl. Dyn. Sys. 2019.

Multiscale control of agent-based dynamics

- Microscopic dynamics:

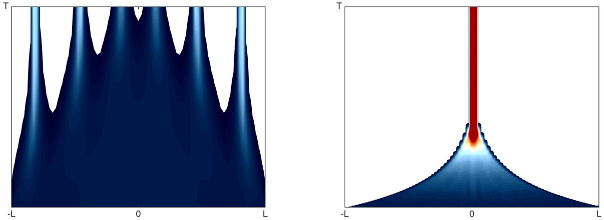
$$\dot{x}_i(t) = \frac{1}{N}\sum_{j=1}^{N} P(x_i, x_j)(x_j - x_i) + u_i(t), \qquad i = 1, \dots, N.$$

- Mean field approximation: $\{x_i(t)\}_{i=1}^N \underset{N\to\infty}{\approx} \mu = \mu(x,t)$, $\{u_i(t)\}_{i=1}^N \approx u = u(x,t)$ satisfying:

$$\partial_t m = \nabla \cdot ((\mathcal{P}[m] + u)\, m)\,, \quad \mathcal{P}[m](x) := \int_{\mathbb{R}^d} P(x,y)(y-x)m(y,t)\, dy\,.$$

Bounded confidence model:

$$x_i(t+1) = \frac{\sum\limits_{j:|x_i(t)-x_j(t)|<1} x_j(t)}{\sum\limits_{j:|x_i(t)-x_j(t)|<1} 1}$$

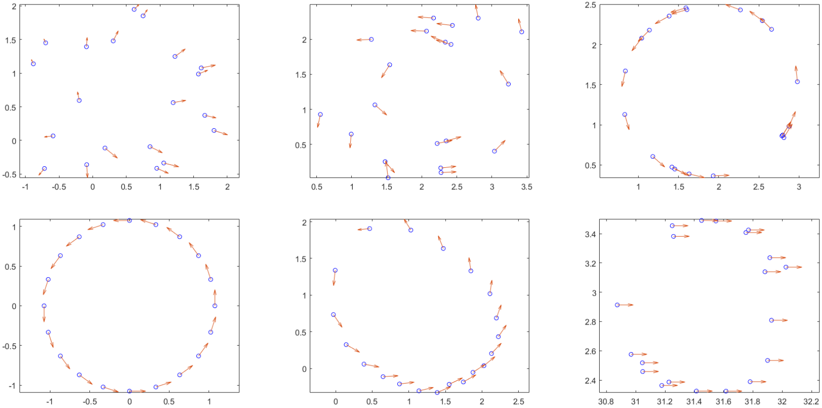


G. Albi, Y.P. Choi, M. Fornasier, and D. K. Mean field control hierarchy, Appl. Math. Optim. 2017.
Acemoglu, D., *Opinion fluctuations and disagreement in social networks*, Mathematics of Operations Research, 2013.

PART I: OPTIMAL FEEDBACK CONTROL FOR MICROSCOPIC DYNAMICS

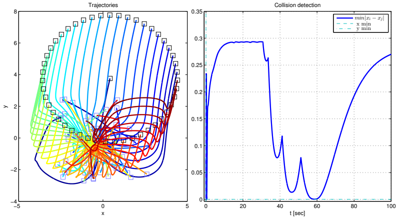
Nonlinear interacting particle systems: flocking, milling, consensus

$$\frac{d}{dt}x_i(t) = v_i(t), \qquad \frac{d}{dt}v_i(t) = -\frac{1}{N}\sum_{j\neq i}^{N} \nabla W(x_i(t) - x_j(t)) + {\color{red}u_i(t)}, \qquad W(r) = -C_a e^{-r/l_a} + C_r e^{-r/l_r}.$$



M. Caponigro, M. Fornasier, B. Piccoli and E. Trélat. *Sparse stabilization and control of alignment models*, M3AS, 2015.
A. Borzi and S. Wongkaew. *Modeling and control through leadership of a refined flocking system*, M3AS, 2015.
J.A. Carrillo, D.K., F. Rossi and E. Trélat. *Controlling Swarms Toward Flocks and Mills*, SIAM J. Control Optim., 2022.

The Hamilton-Jacobi-Bellman PDE in control



$$\underset{\mathbf{u}(\cdot)\in\mathcal{U}}{\text{minimize}} \quad \mathcal{J}(\mathbf{u}(\cdot);\mathbf{x}) := \int_0^{\infty} \ell(\mathbf{y}(t)) + \|\mathbf{u}(t)\|_R^2 \, dt$$

$$\text{subject to} \qquad \dot{\mathbf{y}}(t) = \mathbf{f}(\mathbf{y}(t)) + \mathbf{g}(\mathbf{y}(t))\mathbf{u}(t), \quad \mathbf{y}(0) = \mathbf{x} \in \mathbb{R}^d .$$

Dynamic Programming (Bellman 1950s): the value function

$$V(\mathbf{x}) := \inf_{\mathbf{u}(\cdot)\in\mathcal{U}} \mathcal{J}(\mathbf{u};\mathbf{x}), \quad \mathcal{U} \equiv L^{\infty}([0,+\infty); U),$$

satisfies the Hamilton-Jacobi-Bellman equation

$$\inf_{\mathbf{u}\in U} \left[(\mathbf{f}(\mathbf{x}) + \mathbf{g}(x)\mathbf{u})^{\top} \nabla V(\mathbf{x}) + \ell(\mathbf{x}) + \|\mathbf{u}\|_R^2 \right] = 0, \qquad \mathbf{x} \in \mathbb{R}^d . \tag{HJB}$$

The optimal control is a feedback map:

$$\mathbf{u}^*(\mathbf{x}(t)) := \underset{\mathbf{u}\in U}{\operatorname{argmin}} \left[(\mathbf{f}(\mathbf{x}) + \mathbf{g}(x)\mathbf{u})^{\top} \nabla V(\mathbf{x}) + \ell(\mathbf{x}) + \|\mathbf{u}\|_R^2 \right]$$

W. E. The Dawning of a New Era in Applied Mathematics. Notices of the AMS, April 2021.

High-dimensional HJB and Machine Learning

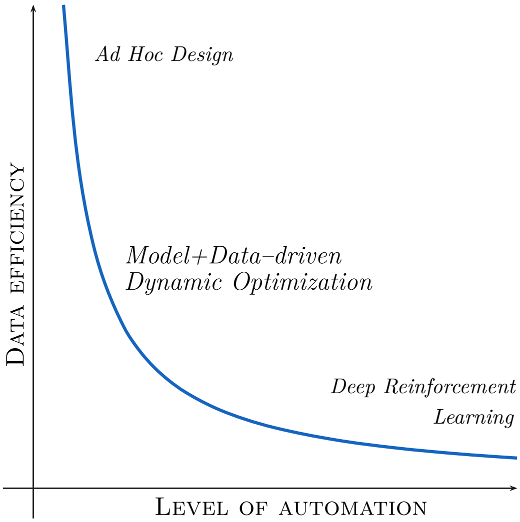
- Reinforcement Learning (Bertsekas-Tsitsiklis Neuro-Dynamic Programming in the 90’s),
- Deep BSDE solver (E-Han-Jentzen 17’) -stochastic control-,
- Deep Galerkin Method (Sirignano-Spiliopoulos 18’) -stochastic control-,
- Polynomial regression + supervised learning (Azmi et al. 21’) - deterministic-.
- Deep neural networks + supervised learning (Nakamura-Zimmerer et al. 19’, Bicego et al. 21’) .
- DNN for computing Lyapunov Functions (Gruene 20’).
- ML for MFG and MFOC (Ruthotto et al. 21’).
- Data-driven tensor decompositions (Dolgov et. al 23’).

Abstract—The deep learning boom motivates researchers and practitioners of computational fluid dynamics eager to integrate the two areas. The PINN (physics-informed neural network) method is one such attempt. While most reports in the literature show positive outcomes of applying the PINN method, our experiments with it stifled such optimism. This work presents our not-so-successful story of using PINN to solve two fundamental flow problems: 2D Taylor-Green vortex at $Re = 100$ and 2D cylinder flow at $Re = 200$. The PINN method solved the 2D Taylor-Green vortex problem with acceptable results, and we used this flow as an accuracy and performance benchmark. About 32 hours of training were required for the PINN method’s accuracy to match the accuracy of a 16×16 finite-difference simulation, which took less than 20 seconds. The 2D cylinder flow, on the other hand, did not even result in a physical solution. The PINN method behaved like a steady-flow solver and did not capture the vortex shedding phenomenon. By sharing our experience, we would like to emphasize that the PINN method is still a work-in-progress. More work is needed to make PINN feasible for real-world problems. (Reproducibility package: [Chu22].)

learning. These partial differential equations include the well-known Navier-Stokes equations—one of the Millennium Prize Problems. The universal approximation theorem ([Hor]) implies that neural networks can model the solution to the Navier-Stokes equations with high fidelity and capture complicated flow details as long as networks are big enough. This deep learning application is sometimes branded as unsupervised learning—it does not rely on human-provided data, making it sounds very *"AI."* It is unsurprising to see headlines like *"AI has cracked the Navier-Stokes equations"* in recent popular science articles ([Hao]).

The PINN method promises several advantages over traditional numerical methods (such as finite volume methods). First, it is a mesh-free scheme. A mesh-free scheme benefits engineering problems in which fluid flows interact with objects of complicated

Model+data-driven dynamic optimization



A concrete example:
Lax-Hopf formula (Osher-Darbon 16')

$$\frac{\partial V(\mathbf{x}, t)}{\partial t} + \frac{1}{2}\|\nabla_x V(\mathbf{x}, t)\|^2 = 0\,,$$
$$V(\mathbf{x}, 0) = \mathcal{J}(\mathbf{x})$$
$$V(\mathbf{x}, t) = -\min_{\mathbf{y}\in\mathbb{R}^d} \left\{\mathcal{J}^*(\mathbf{y}) + \frac{t}{2}\|\mathbf{y}\|^2 - \langle \mathbf{x}, \mathbf{y}\rangle\right\}\,.$$

- Convex optimization problem.
- Solvable in real-time.
- ∇V can be computed for free.
- What is the counterpart for relevant control problems?

J. Darbon and S. Osher, Algorithms for Overcoming the Curse of Dimensionality for Certain Hamilton-Jacobi Equations Arising in Control Theory and Elsewhere, Res. Math. Sci. 2016

Finite horizon optimal control

$$\min_{u(\cdot)\in L^2(t_0,T;\mathbb{R}^m)} \mathcal{J}(\mathbf{u};t_0,\mathbf{x}) := \int_{t_0}^{T} \ell(\mathbf{y}(t)) + \beta\|\mathbf{u}(t))\|_2^2\,dt\,, \qquad \beta > 0\,,$$

$$\text{subject to} \qquad \frac{d}{dt}\mathbf{y}(t) = \mathbf{f}(\mathbf{y}(t)) + \mathbf{g}(\mathbf{y}(t))\mathbf{u}(t)\,, \qquad \mathbf{y}(t_0) = \mathbf{x} \in \mathbb{R}^d\,.$$

- Optimal feedback law: $\mathbf{u}^*(t,\mathbf{x}) = -\frac{1}{2\beta}\mathbf{g}^\top(\mathbf{x})\nabla V(t,\mathbf{x})$, where $V(t,\mathbf{x}) : [0,T]\times\mathbb{R}^d \Rightarrow \mathbb{R}$ solves

$$\partial_t V(t,\mathbf{x}) - \frac{1}{4\beta}\nabla V(t,\mathbf{x})^\top \mathbf{g}(\mathbf{x})\mathbf{g}^\top(\mathbf{x})\nabla V(t,\mathbf{x}) + \nabla V(t,\mathbf{x})^\top \mathbf{f}(\mathbf{x}) + \ell(\mathbf{x}) = 0\,, \quad V(T,\mathbf{x}) = 0\,.$$

- Pontryagin's Maximum Principle for a single trajectory departing from $\mathbf{y}(t_0) = \mathbf{x}$:

$$\begin{cases} \dot{\mathbf{y}}(t) = \mathbf{f}(\mathbf{y}(t)) + \mathbf{g}(\mathbf{y}(t))\mathbf{u}(t)\,, & \mathbf{y}(t_0) = \mathbf{x}\,, \\ -\dot{\mathbf{p}}(t) = \nabla_{\mathbf{y}}(\mathbf{f}(\mathbf{y}(t)) + \mathbf{g}(\mathbf{y}(t))\mathbf{u}(t))^\top \mathbf{p}(t) + \nabla_{\mathbf{y}}\ell(\mathbf{y}(t))\,, & \mathbf{p}(T) = 0\,, \\ \mathbf{u}(t) = -\dfrac{1}{2\beta}\mathbf{g}^\top(\mathbf{y}(t))\mathbf{p}(t)\,, & \forall t \in (t_0,T)\,. \end{cases} \tag{TPBVP}$$

The link between HJB and PMP

Theorem (Mirică 85', Subbotina 06', Yegorov and Dower 17')

Let $\mathbf{f}, \mathbf{g}$ *and* ℓ *be* $C^1(\mathbb{R}^d)$*. Then, the characteristic curves of the HJB PDE*

$$\partial_t V(t,\mathbf{x}) - \frac{1}{4\beta}\nabla V(t,\mathbf{x})^\top \mathbf{g}(\mathbf{x})\mathbf{g}^\top(\mathbf{x})\nabla V(t,\mathbf{x}) + \nabla V(t,\mathbf{x})^\top \mathbf{f}(\mathbf{x}) + \ell(\mathbf{x}) = 0\,, \qquad V(T,\mathbf{x}) = 0\,.$$

correspond to the solution of the TPBVP departing from $\mathbf{y}(t_0) = \mathbf{x}$ *:*

$$\begin{cases} \dot{\mathbf{y}}(t) = \mathbf{f}(\mathbf{y}(t)) + \mathbf{g}(\mathbf{y}(t))\mathbf{u}(t)\,, & \mathbf{y}(t_0) = \mathbf{x}\,, \\ -\dot{\mathbf{p}}(t) = \nabla_{\mathbf{y}}(\mathbf{f}(\mathbf{y}(t)) + \mathbf{g}(\mathbf{y}(t))\mathbf{u}(t))^\top \mathbf{p}(t) + \nabla_{\mathbf{y}}\ell(\mathbf{y}(t))\,, & \mathbf{p}(T) = 0\,, \\ \mathbf{u}(t) = -\dfrac{1}{2\beta}\mathbf{g}^\top(\mathbf{y}(t))\mathbf{p}(t)\,, & \forall t \in (t_0, T)\,. \end{cases} \tag{TPBVP}$$

Moreover, along an optimal trajectory $(\mathbf{y}^*(t), \mathbf{p}^*(t), \mathbf{u}^*(t); \mathbf{x})$*, the value function and its gradient satisfy*

$$\boxed{V(t,\mathbf{y}^*(t)) = \int_t^T \ell(\mathbf{y}^*(s)) + \beta\|\mathbf{u}^*(s))\|_2^2\, ds\,, \quad \nabla V(t,\mathbf{y}^*(t)) = \mathbf{p}^*(t)\,, \quad \forall t \in (t_0, T)}$$

N. N. Subbotina. The method of characteristics for the Hamilton-Jacobi equations and its applications in dynamic optimization. J. Math. Sci., 2006.

Supervised learning - architectures for training

$$\text{Loss:} \quad \mathcal{L}(V_\theta, \nabla V_\theta) := \frac{1}{N_s}\sum_{i=1}^{N_s} \|V(\mathbf{x}^{(i)}) - V_\theta(\mathbf{x}^{(i)})\|^2 + \mu\|\nabla V(\mathbf{x}^{(i)}) - \nabla V_\theta(\mathbf{x}^{(i)})\|^2 \,.$$

- Gradient-augmented supervised learning (Nakamura-Zimmerer/Gong/Kang 2019):

$$\mathbf{u}_V(\mathbf{x}) := -\frac{1}{2\beta}\mathbf{g}^\top(\mathbf{x})\nabla V_\theta(\mathbf{x})\,, \quad V_\theta(\mathbf{x}) = l_M \circ \ldots \circ l_2 \circ l_1(\mathbf{x}), \quad l_m(\mathbf{y}) = \sigma_m(\mathbf{A}_m\mathbf{y} + \mathbf{b}_m)\,.$$

- Using recurrent neural networks (Albi/Bicego/DK 2022):

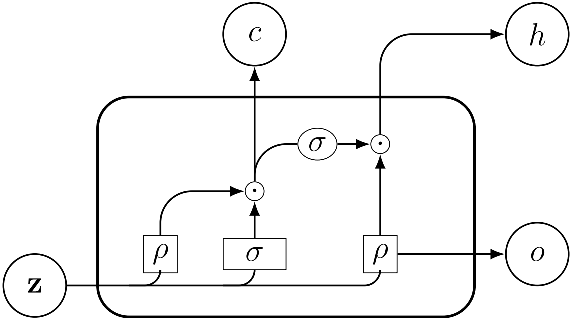


Figure: One-to-one LSTM cell

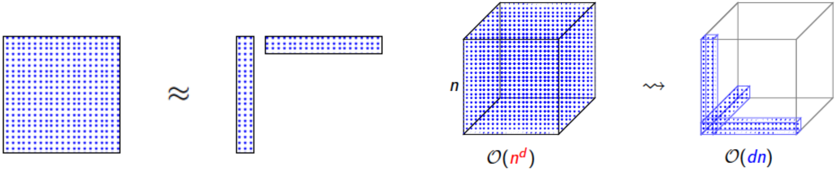
$i = \rho\big(W_i\,\mathbf{z} + b_i\big)$	input gate
$\tilde{c} = \sigma\big(W_c\,\mathbf{z} + b_c\big)$	candidate value
$c = i \odot \tilde{c}$	cell value
$o = \rho\big(W_o\,\mathbf{z} + b_o\big)$	output gate
$h = o \odot \sigma\big(c\big)$	final output

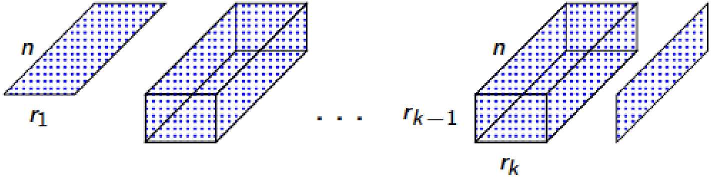
Supervised learning - further architectures for training

$$\text{Loss:} \quad \mathcal{L}(V, V_\theta) := \frac{1}{N_s}\sum_{i=1}^{N_s} \|V(\mathbf{x}^{(i)}) - V_\theta(\mathbf{x}^{(i)})\|^2 + \mu\|\nabla V(\mathbf{x}^{(i)}) - \nabla V_\theta(\mathbf{x}^{(i)})\|^2 \,.$$

- Gradient-augmented TT approximation (Dolgov/DK/Saluzzi 2021-2023):

$$V_\theta(\mathbf{x}) := \sum_{\alpha_0,\ldots,\alpha_d=1}^{r_0,\ldots,r_d} \mathbf{v}^{(1)}_{\alpha_0,\alpha_1}(x_1)\mathbf{v}^{(2)}_{\alpha_1,\alpha_2}(x_2)\cdots\mathbf{v}^{(d)}_{\alpha_{d-1},\alpha_d}(x_d)$$





Supervised learning - further architectures for training

$$\text{Loss:} \quad \mathcal{L}(V, V_\theta) := \frac{1}{N_s}\sum_{i=1}^{N_s} \|V(\mathbf{x}^{(i)}) - V_\theta(\mathbf{x}^{(i)})\|^2 + \mu\|\nabla V(\mathbf{x}^{(i)}) - \nabla V_\theta(\mathbf{x}^{(i)})\|^2 .$$

- Gradient-augmented TT approximation (Dolgov/DK/Saluzzi 2021-2023):

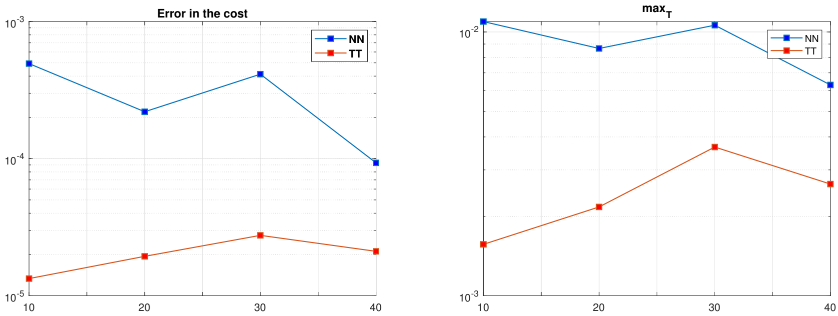
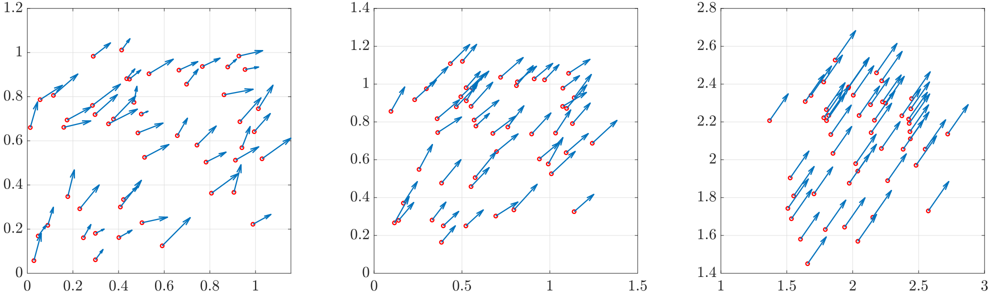


Figure: Error in the cost functional (left) and $\tilde{y}_{\max}(T)$ (right) for TT and NN for different dimensions.

S. Dolgov, D.K., and L. Saluzzi , Data-driven Tensor Train Gradient Cross Approximation for Hamilton-Jacobi-Bellman Equations, SISC (2023).

Supervised learning - feedback control map for 50 agents (200 states!)

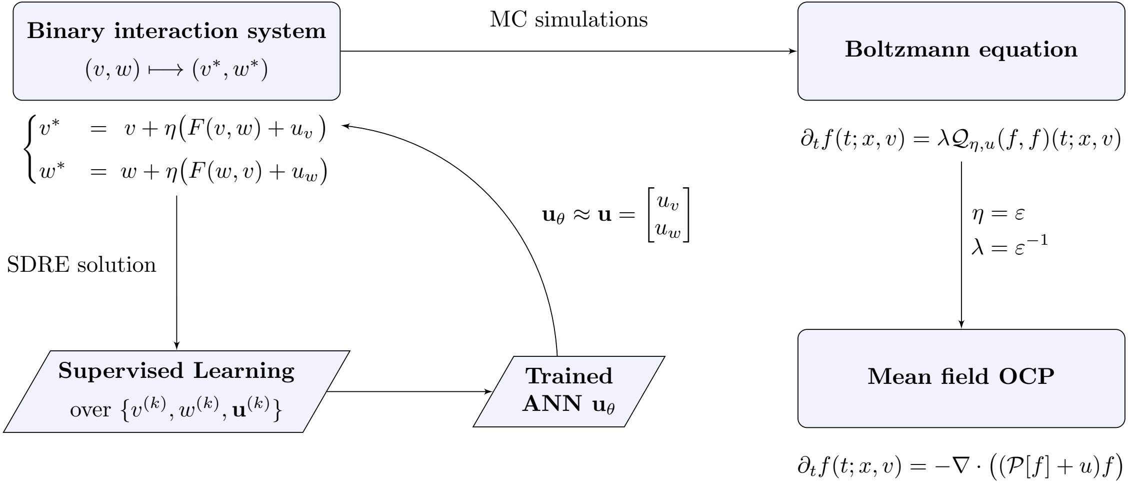
$$\text{Loss:}\quad \mathcal{L}(V, V_\theta) := \frac{1}{N_s}\sum_{i=1}^{N_s} \|V(\mathbf{x}^{(i)}) - V_\theta(\mathbf{x}^{(i)})\|^2 + \mu\|\nabla V(\mathbf{x}^{(i)}) - \nabla V_\theta(\mathbf{x}^{(i)})\|^2 \,.$$



System configuration at time $t = 0, 1, 10$ seconds respectively, under the feedback map $\mathbf{u}_V^{PMP}$.

PART II: EMBEDDING MICROSCOPIC LAWS INTO MESOSCOPIC DYNAMICS

Embedding high-dimensional feedback laws into kinetic models



Boltzmann description of the multi-agent control problem

Controlled binary post-interaction dynamics with strength parameter η

$$\begin{cases} x \mapsto x^* & = x + \eta\left[P(x,y)(y-x) + u(x,y)\right] \\ y \mapsto y^* & = y + \eta\left[P(y,x)(x-y) + u(y,x)\right] \end{cases}$$

$$Q_{\eta,u}(f,f)(t;x) = \underbrace{\int_\Omega \frac{1}{\mathcal{J}_\eta} f(t,{}^*x) f(t,{}^*y)dy}_{\text{gain}Q^+_{\eta,u}(f,f)} + \underbrace{f(t,x)\int_\Omega f(t,y)dy}_{\text{loss}Q^-_{\eta,u}(f,f)}$$

with $({}^*x, {}^*y) \longmapsto (x,y)$ pre-interaction states, $\mathcal{J}_\eta$ Jacobian of binary interactions.

The evolution of $f(t;x)$ is driven by Boltzmann-type dynamics with frequency λ

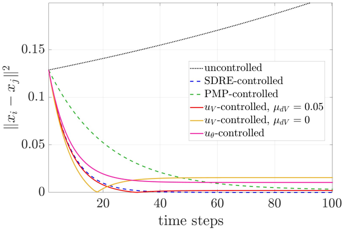
$$\partial_t f(t,x) = \lambda Q_{\eta,u}(f,f)(t,x)$$

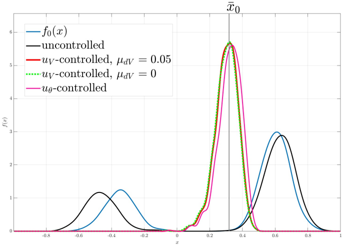
Under quasi-invariant scaling, i.e. $\eta = \varepsilon, \lambda = \varepsilon^{-1}$, the kinetic model is consistent with the mean field dynamics.

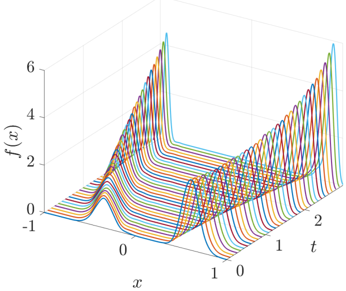
G. Albi, Y.P. Choi, M. Fornasier, and D. K. Mean field control hierarchy, Appl. Math. Optim. 2017
G. Albi, M. Herty, D. K. and C. Segala Moment-Driven Predictive Control of Mean-Field Collective Dynamics, SICON 2022

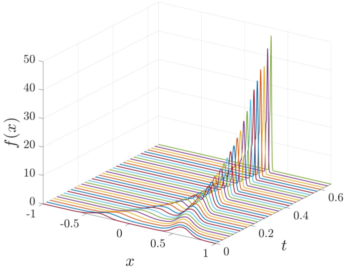
Controlling opinion dynamics: Sznajd's model

Interaction kernel: $P(x_i, x_j) = -(1 - x_i^2)$, $\Omega = [-1, 1]$, $N_s = 10^5$.





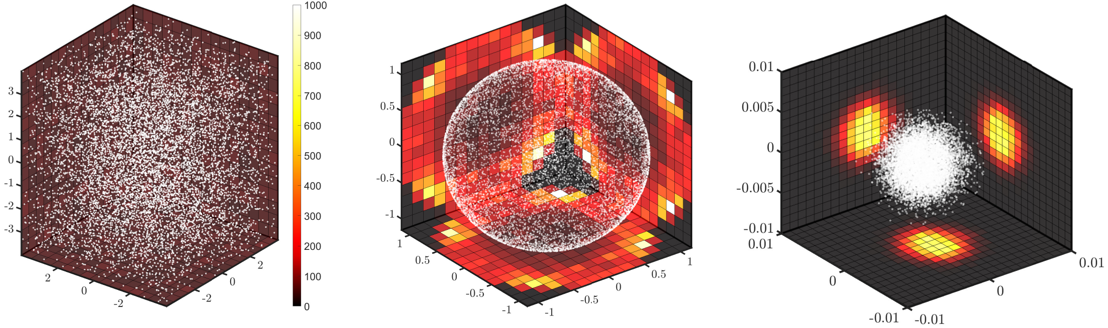




Embedding high-dimensional feedback laws into kinetic models

Consensus control for attraction-repulsion dynamics:

$$\frac{d}{dt}x_i(t) = v_i(t), \quad \frac{d}{dt}v_i(t) = -\frac{1}{N}\sum_{j\neq i}^{N}\nabla W(x_i(t)-x_j(t)) + u_i(t), \qquad W(r) = -C_a e^{-r/l_a} + C_r e^{-r/l_r}.$$



(a) $t = 0$ (b) $t = 2$, uncontrolled (c) $t = 2$, controlled

Embedding high-dimensional feedback laws into kinetic models

Consensus control for attraction-repulsion dynamics:

$$\frac{d}{dt}x_i(t) = v_i(t), \quad \frac{d}{dt}v_i(t) = -\frac{1}{N}\sum_{j\neq i}^{N} \nabla W(x_i(t) - x_j(t)) + u_i(t), \qquad W(r) = -C_a e^{-r/l_a} + C_r e^{-r/l_r}.$$

$N_s = 10^4$	$d = 3$	$d = 7$	$d = 10$	$d = 15$	$d = 30$
s_θ^{*FNN}	1.048226	1.212633	1.390813	2.142041	2.617840
s_θ^{*RNN}	2.033726	2.243084	2.493256	3.210856	3.893368
u_θ^{FNN}	7.712628	11.006977	15.041731	21.754862	70.172311
u_θ^{RNN}	7.734224	11.224325	15.991421	22.486736	70.564372
u	1.1979×10^3	5.2136×10^3	–	–	–

Table: CPU times (seconds) when considering 10^4 MC samples of coupled agents in $\mathbb{R}^{4d}$, with varying d. The omitted records exceeded a time threshold $t_{max} = 24h$.

PART III: CONTROLLING THE FOKKER-PLANCK EQUATION

Mean field control of agent-based dynamics

- Microscopic dynamics:

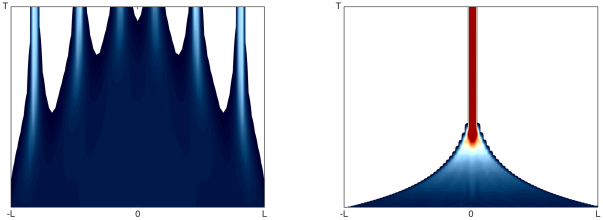
$$\dot{x}_i(t) = \frac{1}{N}\sum_{j=1}^{N} P(x_i, x_j)(x_j - x_i) + u_i(t), \qquad i = 1, \dots, N\,.$$

- Mean field approximation: $\{x_i(t)\}_{i=1}^N \underset{N\to\infty}{\approx} m = m(x,t)\,,\ \{u_i(t)\}_{i=1}^N \approx u = u(x,t)$ satisfying:

$$\partial_t m = \nabla\cdot((\mathcal{P}[m] + u - \nabla V(x))\, m) + \beta^{-1}\Delta m, \quad \mathcal{P}[m](x) := \int_{\mathbb{R}^d} P(x,y)(y-x)m(y,t)\, dy\,.$$

Bounded confidence model:

$$x_i(t+1) = \frac{\sum\limits_{j:|x_i(t)-x_j(t)|<1} x_j(t)}{\sum\limits_{j:|x_i(t)-x_j(t)|<1} 1}$$



G. Albi, Y.P. Choi, M. Fornasier, and D. K. Mean field control hierarchy, Appl. Math. Optim. 2017
J. Garnier, G. Papanicolau, and T-W. Zhang. Consensus Convergence with Stochastic Effects, Vietnam J. Math. 2017
Acemoglu, D., *Opinion fluctuations and disagreement in social networks*, Mathematics of Operations Research, 2013.

Finding steady states of the Fokker-Planck Equation

Opinion dynamics, Markov chain Monte Carlo, and global optimisation require knowledge about m_∞:

$$0 = \nabla \cdot \left(\left(\mathcal{P}[m_\infty] - \nabla V(x) \right) m_\infty \right) + \beta^{-1} \Delta m_\infty \, .$$

We look for multiple solutions using spectral approximation and deflation:

$$g(x) = \frac{f(x)}{\prod_{i=1}^{n} (x - x_i)} \qquad \Longrightarrow \mathcal{G}(c) = \mathcal{F}(c) \left(\frac{\mathcal{I}}{\eta(c)} + \mathcal{I}\, \xi \right), \qquad \eta(c) = \|c - r\|^p \, .$$

Deflated Newton iteration:

$$c^{n+1} = c^n - \frac{\mathcal{G}(c^n)}{\mathcal{G}'(c^n)} \, , \qquad \mathcal{G}' = \frac{\mathcal{F}'}{\eta_n} - \frac{\mathcal{F}}{\eta_n^2} \otimes \left(\eta'_{n-1}\, \tau + \eta_{n-1}\, \tau' \right) .$$

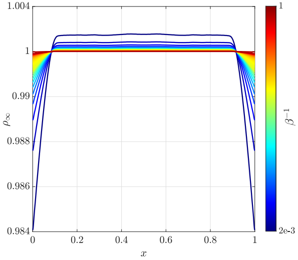
P. E. Farrell, Á. Birkisson, and S. W. Funke. Deflation Techniques for Finding Distinct Solutions of Nonlinear Partial Differential Equations, SISC 2015.

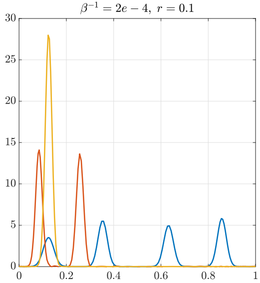
Finding steady states of the Fokker-Planck Equation

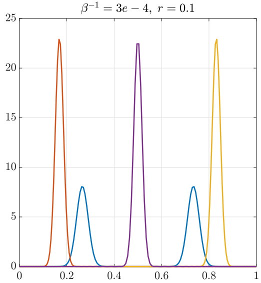
Opinion dynamics, Markov chain Monte Carlo, and global optimisation require knowledge about m_∞:

$$0 = \nabla \cdot \left(\left(\mathcal{P}[m_\infty] - \nabla V(x) \right) m_\infty \right) + \beta^{-1} \Delta m_\infty \, .$$

We look for multiple solutions using spectral approximation and deflation:







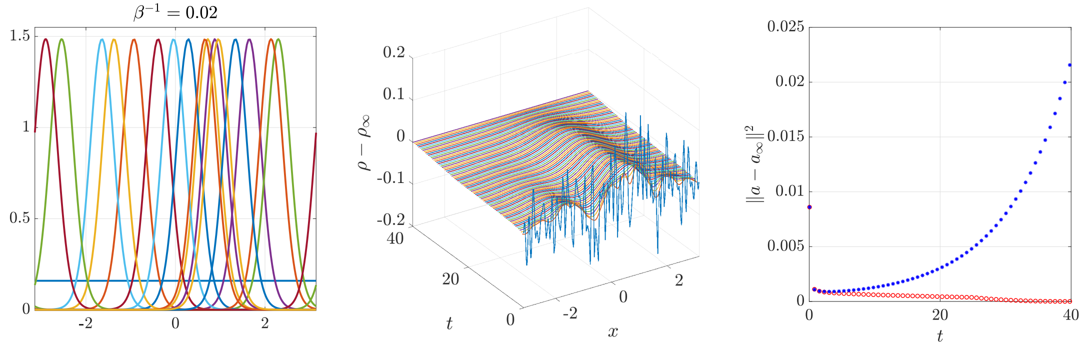
A. Barbaro, J. Cañizo, J.A. Carrillo and P. Degond Phase transitions in a kinetic flocking model of Cucker-Smale type, Multiscale Model Simul. 2016
J. Garnier, G. Papanicolau, and T-W. Zhang. Consensus Convergence with Stochastic Effects, Vietnam J. Math. 2017
M.G. Delgadino , R.S. Gvalani and G.A.Pavliotis On the Diffusive-Mean Field Limit for Weakly Interacting Diffusions Exhibiting Phase Transitions, ARMA 2021

Finding steady states of the Fokker-Planck Equation

Opinion dynamics, Markov chain Monte Carlo, and global optimisation require knowledge about m_∞:

$$0 = \nabla \cdot ((\mathcal{P}[m_\infty] - \nabla V(x))\, m_\infty) + \beta^{-1} \Delta m_\infty \,.$$

We control towards unstable solutions with model predictive control:



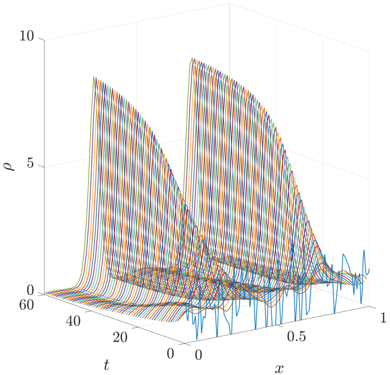
S. Bicego, D. K. and G. Pavliotis. Deflation and control in the Fokker-Planck equation, in preparation.

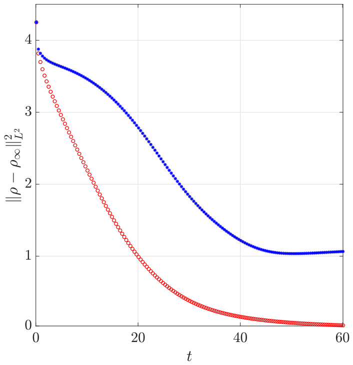
Finding steady states of the Fokker-Planck Equation

Opinion dynamics, Markov chain Monte Carlo, and global optimisation require knowledge about m_∞:

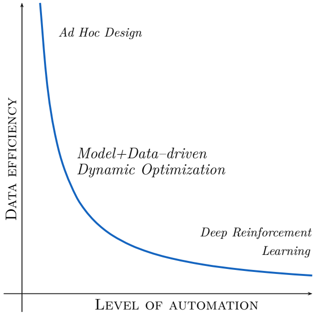
$$0 = \nabla \cdot ((\mathcal{P}[m_\infty] - \nabla V(x))\, m_\infty) + \beta^{-1} \Delta m_\infty \,.$$

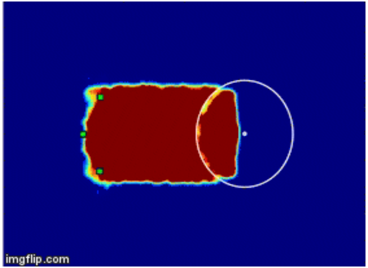
We control towards unstable solutions with model predictive control:





Outlook





Some final thoughts:

- Computing high-dimensional optimal feedback laws is feasible.
- Causality-free, data-driven schemes can be built using optimal control theory.
- Collective dynamics can be controlled at the kinetic/binary and mean-field level.
- Some classical problems remain open:

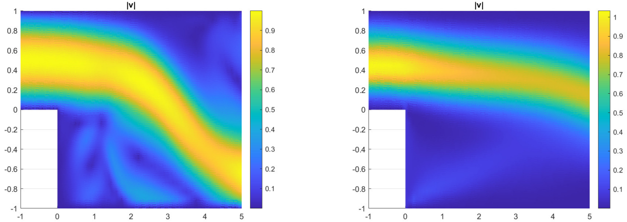


Figure: Optimal feedback control for Navier-Stokes

References

High-dimensional TT-HJB solver: S. Dolgov, D.K. and K. Kunisch, Tensor Decompositions Methods for High-dimensional Hamilton-Jacobi-Bellman Equations, SIAM J. Sci. Comput., 2021.

Data-driven tensor train: S. Dolgov, D.K., and L. Saluzzi , Data-driven Tensor Train Gradient Cross Approximation for Hamilton-Jacobi-Bellman Equations, to appear in SIAM J. Sci. Comput., 2023.

Gradient-augmented regression: B. Azmi, D.K. and K. Kunisch. Optimal Feedback Law Recovery by Gradient-Augmented Sparse Polynomial Regression, J. Mach. Learn. Res., 2021.

NN-SDRE solver: G. Albi, S. Bicego, and D. K. Gradient-augmented Supervised Learning of Optimal Feedback Laws Using State-dependent Riccati Equations, IEEE Control Systems Letters, 2022.

UKRI Engineering and Physical Sciences Research Council

Preprints and more: www.dkalise.net
Contact me 🙂: dkaliseb@ic.ac.uk

Questions?

If you'd like to avoid the awkward silence that usually follows after this talk, you may want to ask:

- What are the limitations of TT and NN architectures in high-dimensional feedback control?
- What about other NN-based approaches (e.g. PINNs) for high-dimensional HJB?
- Is there a data-driven framework for infinite-horizon HJB?
- Can you say something about the performance of the binary feedback law compared to mean-field optimal control?
- What's the link with OT?
- What about optimal feedback control of the Fokker-Plank equation?

Imperial College London

Preprints and more: www.dkalise.net
Contact me 🙂: dkaliseb@ic.ac.uk

Multiscale Feedback Control Synthesis for Interacting Particle Systems

Dante Kalise

Department of Mathematics
Imperial College London

joint work with G. Albi (Verona) and Sara Bicego (Imperial)
Numerical Methods for Optimal Transport Problems, Mean Field Games, and Multi-agent Systems,
UTFSM, Valparaíso, January 9, 2024

Engineering and Physical Sciences Research Council

Imperial College London

Infinite horizon feedback control

$$\min_{\mathbf{u}(\cdot)\in\mathcal{U}} J(\mathbf{u}(\cdot),\mathbf{x}_0) := \int_0^\infty \mathbf{x}^\top(s)\mathbf{Q}\mathbf{x}(s) \;+\; \mathbf{u}^\top(s)\mathbf{R}\mathbf{u}(s)\,ds$$

$$\text{subject to:}\quad \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{B}(\mathbf{x}(t))\mathbf{u}(t)\,,\qquad \mathbf{x}(0) = \mathbf{x}_0\,.$$

Unconstrained Dynamic Programming: with $\mathcal{U} \equiv L^\infty([0,+\infty);\mathbb{R}^m)$, $V(\mathbf{x})$ solves

$$\nabla V(\mathbf{x})^\top\mathbf{f}(\mathbf{x}) - \frac{1}{4}\nabla V(\mathbf{x})^\top\mathbf{B}(\mathbf{x})\mathbf{R}^{-1}\mathbf{B}(\mathbf{x})^\top\nabla V(\mathbf{x}) + \mathbf{x}^\top\mathbf{Q}\mathbf{x} = 0 \quad\Rightarrow\quad \mathbf{u}(\mathbf{x}) = -\frac{1}{2}\mathbf{R}^{-1}\mathbf{B}(\mathbf{x})^\top\nabla V(\mathbf{x})\,.$$

No representation formula!

Linear-quadratic (LQ) case: $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}$, $\mathbf{B}(x) = \mathbf{B}$, and $V(\mathbf{x}) = \mathbf{x}^\top\Pi\mathbf{x}$ with $\Pi \in \mathbb{R}^{d\times d}$ leads to

$$\text{(ARE):}\quad \mathbf{A}^\top\Pi + \Pi\mathbf{A} - \Pi\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top\Pi + \mathbf{Q} = 0\,.$$

We assume $V(\mathbf{x}) \approx \mathbf{x}^\top\Pi(\mathbf{x})\mathbf{x}$ close to the origin.
Writing $\mathbf{f}(\mathbf{x}) = \mathbf{A}(\mathbf{x})\mathbf{x}$, then $V(\mathbf{x}) \approx \mathbf{x}^\top\Pi(\mathbf{x})\mathbf{x}$ solves:

$$\text{(SDRE):}\quad \mathbf{A}^\top(\mathbf{x})\Pi(\mathbf{x}) + \Pi(\mathbf{x})\mathbf{A}(\mathbf{x}) - \Pi(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{R}^{-1}\mathbf{B}(\mathbf{x})^\top\Pi(\mathbf{x}) + \mathbf{Q} = 0\,.$$

Towards a solver for infinite horizon HJB

We want to solve:

$$\nabla V(\mathbf{x})^\top \mathbf{f}(\mathbf{x}) - \frac{1}{4}\nabla V(\mathbf{x})^\top \mathbf{B}(\mathbf{x})\mathbf{R}^{-1}\mathbf{B}(\mathbf{x})^\top \nabla V(\mathbf{x}) + \mathbf{x}^\top \mathbf{Q}\mathbf{x} = 0\,.$$

Our SDRE supervised learning is based on suboptimal feedback law:

$$\text{(SDRE):} \quad \mathbf{A}^\top(\mathbf{x})\Pi(\mathbf{x}) + \Pi(\mathbf{x})\mathbf{A}(\mathbf{x}) - \Pi(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{R}^{-1}\mathbf{B}(\mathbf{x})^\top \Pi(\mathbf{x}) + \mathbf{Q} = 0\,,$$

assuming $V(\mathbf{x}) = \mathbf{x}^\top \Pi(\mathbf{x})\mathbf{x}\,, \nabla V(\mathbf{x}) = 2\Pi(\mathbf{x})\mathbf{x}$, while in reality

$$\nabla V(\mathbf{x}) = 2\Pi(\mathbf{x})\mathbf{x} + \sum_{i,j=1}^{d} \mathbf{x}_i \mathbf{x}_j \frac{\partial \Pi(\mathbf{x})_{i,j}}{\partial \mathbf{x}_k}\,.$$

$$\text{HJB} = \text{SDRE} + \text{terms arising from } \nabla V(x)\,.$$

SDRE is not sufficient for data generation for infinite horizon HJB.

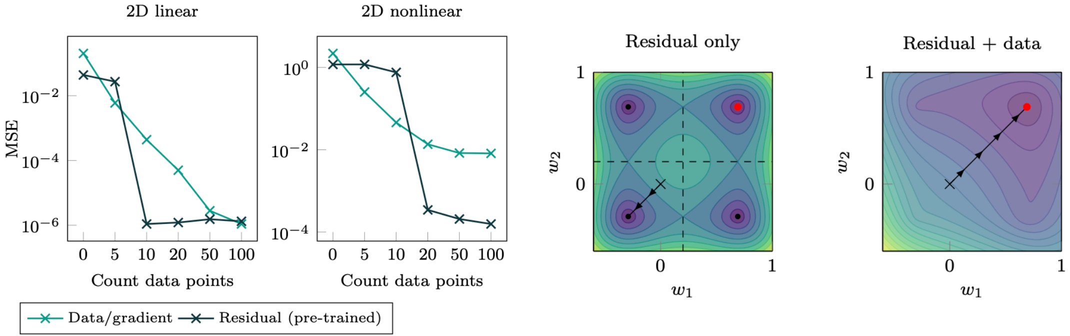
Pre-trained PINNs

Supervised phase:

$$\mathcal{L}^{\text{dat}}(\theta) = \frac{1}{N_1}\sum_{i=1}^{N_1} ||\bar{V}(\mathbf{x}^i) - V_\theta(\mathbf{x}^i)||_2^2 + \frac{\lambda_1}{N_1}\sum_{i=1}^{N_1} ||\nabla \bar{V}(\mathbf{x}^i) - \nabla V_\theta(\mathbf{x}^i)||_2^2 .$$

Residual phase initialized with supervised phase parameters:

$$\mathcal{L}^{\text{res}}(\theta) = \frac{\lambda_2}{N_2}\sum_{i=1}^{N_2} ||\mathcal{N}(\mathbf{x}^i, V_\theta(\mathbf{x}^i))||_2^2 , \qquad \mathcal{N}(\mathbf{x}, V) := \nabla V(\mathbf{x})^\top \mathbf{f}(\mathbf{x}) - \frac{1}{4}\nabla V(\mathbf{x})^\top \mathbf{B}(\mathbf{x})\mathbf{R}^{-1}\mathbf{B}(\mathbf{x})^\top \nabla V(\mathbf{x}) + \mathbf{x}^\top \mathbf{Q}\mathbf{x} .$$



A. Borovykh, D. Kalise, A. Laignelet and P. Parpas. Data-driven initialization of deep learning solvers for Hamilton-Jacobi PDEs, IFAC-PapersOnline, 2022.