On Learning-based Control of Dynamical Systems

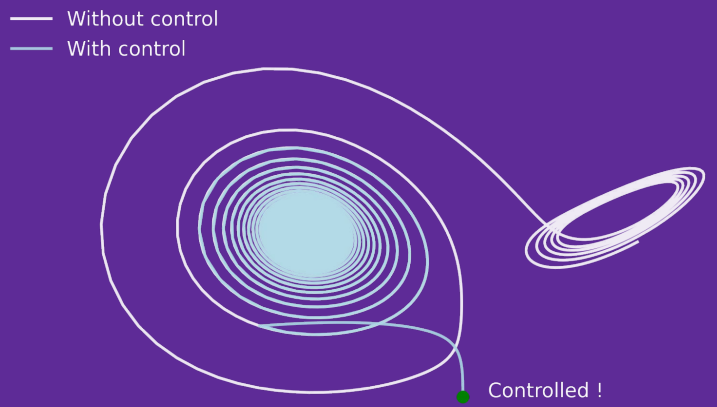
Application to Fluid Dynamics

Rémy Hosseinkhan-Boucher[1,2]

PhD Defense - April 10th 2025

Advisors:
Anne Vilnat[1,2]
Onofrio Semeraro[2]
Lionel Mathelin[2]

[1]Université Paris-Saclay
[2]LISN, CNRS



Dynamical Systems

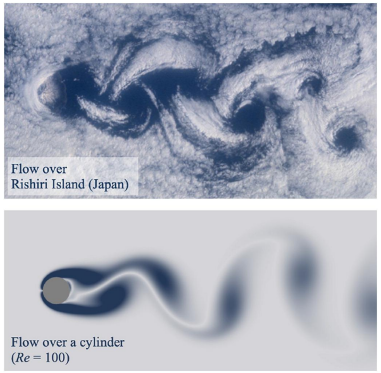
⟶ *Termed by Poincaré in "Les méthodes nouvelles de la mécanique céleste" (1892)*

Definition

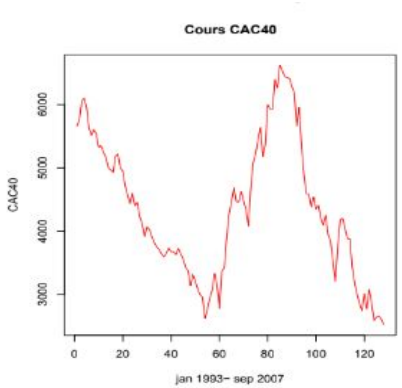
System + time-**evolution law**

Examples of fields

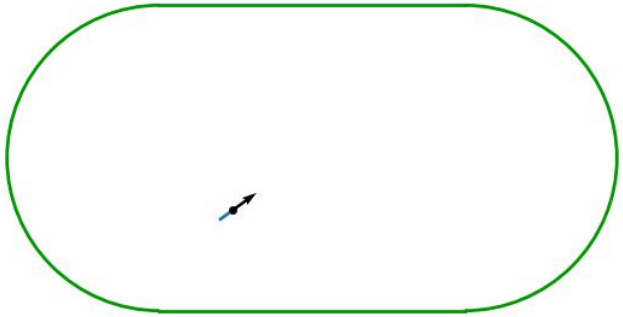
Fluids dynamics Financial markets Games Chemistry Astronomy Epidemiology Autonomous Driving Natural Language Processing



Von Kármán vortex street generated by an island in Japan
Taira *et al.* "Modal Analysis of Fluid Flows: Applications and Outlook", *AIAA Journal* (2019)



CAC40 Index between 1993 and 2007
I. Kharroubi - Gestion de Portefeuilles (2015)



Motion of a particle in the Bunimovich billiard
Credits: George Datseris, Wikipedia (2019)

(Optimal) Control

⟶ *Optimal control emerged post-WWII from automatic control needs in flight systems.*

Control

Force the system **initial state ⟶ final state**

Optimal Control

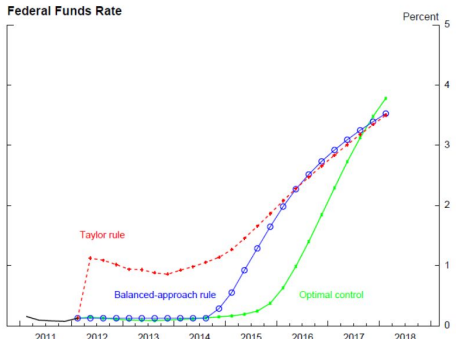
Control + **optimise** a **criterion** J

Control Policy π

Behaviour of the control input

Other Examples

$$J = \min[\textbf{Cash}], \min[\textbf{Time}], \min[\textbf{Energy}], \min[\textbf{Unemployment}], \max[\textbf{Score}]$$

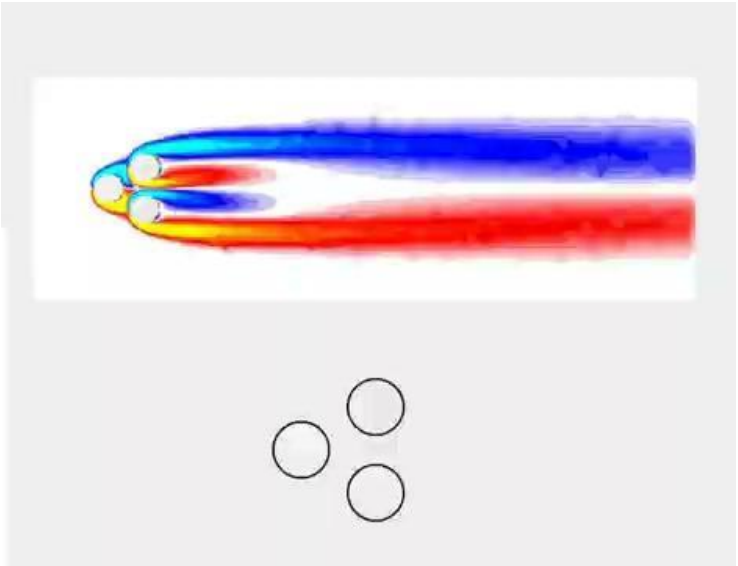


Optimal Control of Monetary Policy

J. L. Yellen
"Perspectives on Monetary Policy"
Boston Economic Club (2012)

Application in Fluids Dynamics

$$J = \max[\textbf{Lift}], \min[\textbf{Drag}], \min[\textbf{Noise}]$$



Fluidic Pinball under control laws - Credits: E. Kaiser

Physical Model-based vs. Learning-based Control

Physical Model

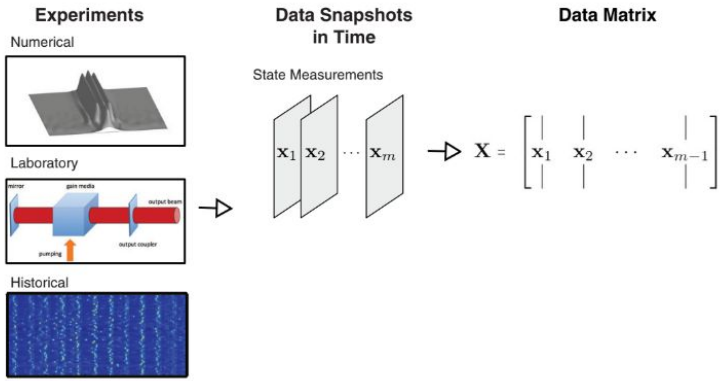
Explicit system representation (e.g. equation) ⟶ **Build** control policy π

Example (Navier-Stokes equation in Fluids Dynamics) ⟶

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \frac{1}{\mathrm{Re}}\Delta \mathbf{u},$$
$$\nabla \cdot \mathbf{u} = 0.$$

Learning-based Model

Implicit system representation from **data and statistics**⟶ **Build** control policy π



Data collection ⟶ **system interaction**

Data-driven methods ⟶ Machine Learning Control

Data can be collected from a number of differents sources
J. N. Kutz *et al.* - Dynamic Mode Decomposition: Data Driven Modeling Of Complex Systems, *SIAM* (2016)

Dynamical Systems Control: Challenges

Optimal Control Problem

Dynamics

$$\partial_t x(t) = f(x(t), u(t))$$

Goal

$$\min_u J(u) = \int_0^T c(x(t), u(t))dt$$

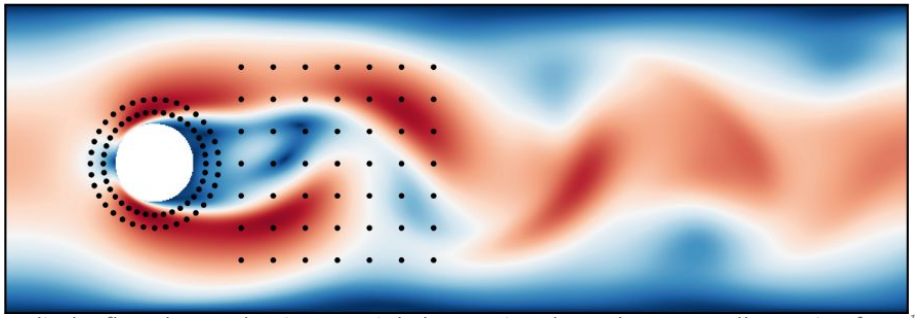
Challenges[1]

- **Robustness**
- **Sample complexity**
- **Partial observability (PO) and delays**
- Controllability
- High dimensional state space
- Very large degrees of freedom
  (*e.g.* sensors & actuators config.)

Example

$f \longrightarrow$ Navier-Stokes equation

Energy cost $\longrightarrow \quad c(x, u) = \|x\|^2 + \|u\|^2$



Cylinder flow drag reduction. Partial observation through sensors. Illustration from [1].

PhD thesis Goal

- Addressing the challenges
- Combine concepts from different fields

[1] J. Viquerat *et al.* "A review on deep reinforcement learning for fluid mechanics: An update", *AIP Publishing* (2022)

Presentation Outline

- ➔ Maximum Entropy: Noise Robustness
- ➔ Sampling Strategies with Semi-Markov Decision Process
- ➔ Towards Neural Controlled Delay Differential Equations
- ➔ *Academic and Scientific Involvement*
- ➔ Conclusion

Maximum Entropy: Noise Robustness

Entropy = Uncertainty Measure

How to quantify the **uncertainty** on the control law π ?

$du \rightarrow$ Infinitesimal volume (a.k.a. event) of $\mathcal{U}$

Uncertainty on *du*

$$I(du) = \log\left(\frac{1}{\pi(du)}\right)$$

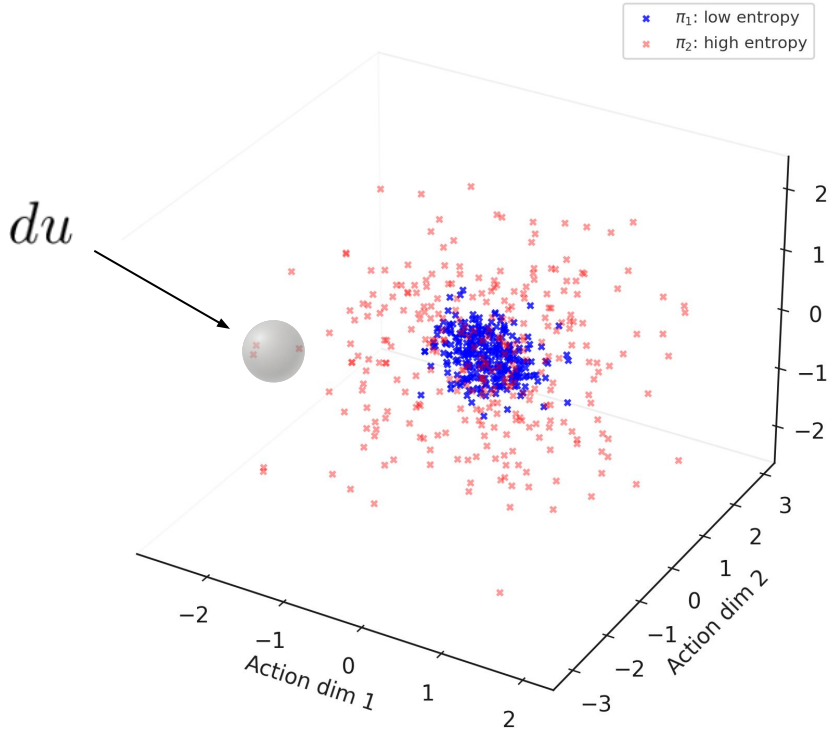
Entropy (average uncertainty)

$$\mathcal{H}(\pi) = \int_{\mathbb{R}} \log\left(\frac{1}{\pi(du)}\right) \pi(du)$$

Gaussian case

$$\pi(du) = f_{\mathcal{N}(\mu,\Sigma)}(u)\, du$$

$$\mathcal{H}(\pi) = \frac{1}{2}\log\left(2\pi e\, \Sigma\right)$$



Robustness: Maximum Entropy and Regularity

Maximum Entropy Reinforcement Learning

$$\arg\min_{\pi} \mathbb{E}^{\pi} \left[\sum_{k=0}^{T} \gamma^k c\left(X_k, U_k\right) - \alpha \mathcal{H}[\pi(du \mid X_k)] \right], \quad \alpha > 0.$$

Control objective

Entropy

Previous Results[1]

- Better exploration
- **Robustness**
- **Loss regularisation**

Questions

Why does entropy **improve robustness**?

Why does entropy **regularise** the optimisation landscape**?**

Hypothesis

Entropy $\longrightarrow$ Policy complexity

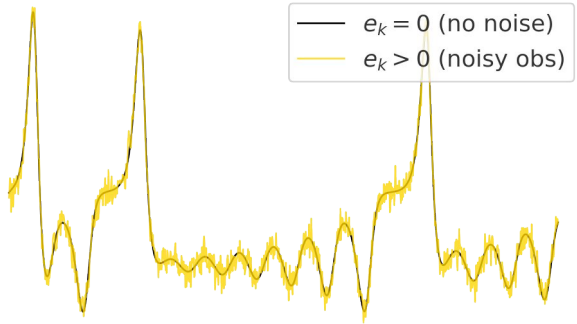
[1]A. Ahmed *et al.* "Understanding Flat Minima in Neural Networks", *ICLR* (2019)

Robustness Measure

Noisy Observable

$$X_{k+1} = F(X_k, U_k)$$
$$Y_{k+1} = G(X_k) + \epsilon_k \quad \epsilon_k \sim \mathcal{N}(0, \sigma_\epsilon^2 I_d)$$

$\longrightarrow$ Feedback observation policy $\pi : \mathcal{Y} \mapsto \pi(Y_{k+1}, du)$



Noisy observation of the 1st coordinate of the Lorenz system

Rate of Excess Risk Under Noise[1]

$\epsilon \equiv 0 \longrightarrow \mathbb{P}^\pi$
$\epsilon \not\equiv 0 \longrightarrow \mathbb{P}^{\pi,\epsilon}$

$$J^{\pi,\epsilon} = \mathbb{E}^{\pi,\epsilon}\left[\sum_{k=0}^{T} \gamma^k c(X_k, U_k)\right]$$

$$\mathcal{R}^\pi = \frac{J^{\pi,\epsilon} - J^\pi}{J^\pi}$$

$\longrightarrow$ How much the objective function deteriorates when $\epsilon \not\equiv 0$

Hypothesis

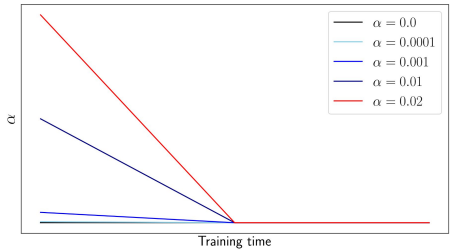
$$\begin{array}{ccc} \epsilon \nearrow & \longrightarrow & J^{\pi^*,\epsilon} \nearrow \\ \alpha > 0 & \longrightarrow & \mathcal{R}^{\pi^\alpha} \searrow \end{array}$$

[1] R. Hosseinkhan-Boucher *et al.* "Evidence on the Regularisation Properties of Maximum-Entropy Reinforcement Learning", *Optimization and Learning Conference* (2024)

Robustness Measure: Experiments

Experiment

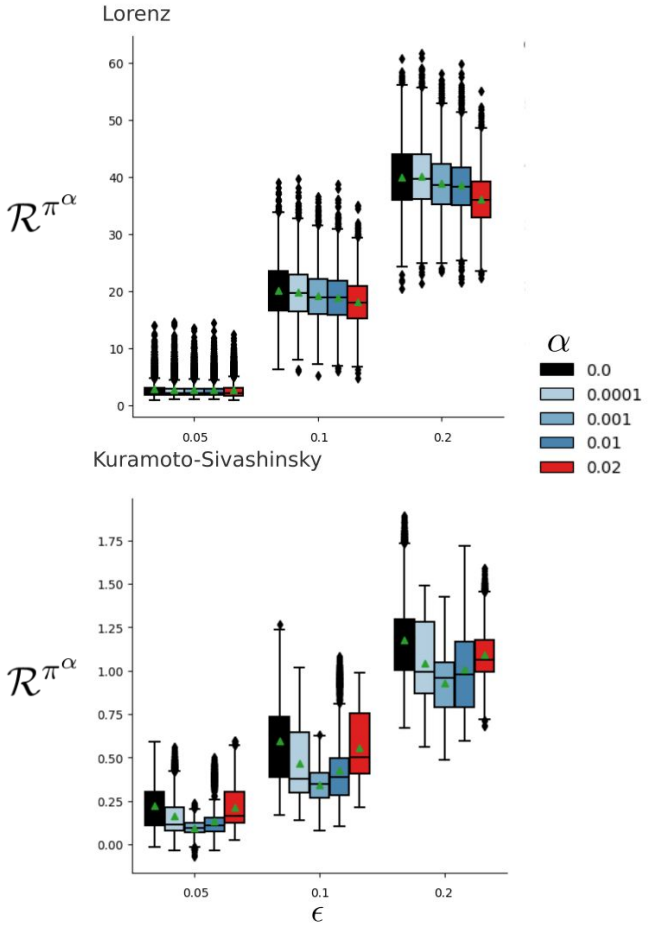
- Train 10 seeds x 5 entropy levels α



- Test over different noise intensities ϵ

Results

$$\epsilon \nearrow \longrightarrow J^{\pi^*,\epsilon} \nearrow$$
$$\alpha > 0 \longrightarrow \mathcal{R}^{\pi^\alpha} \searrow$$



Model Regularity

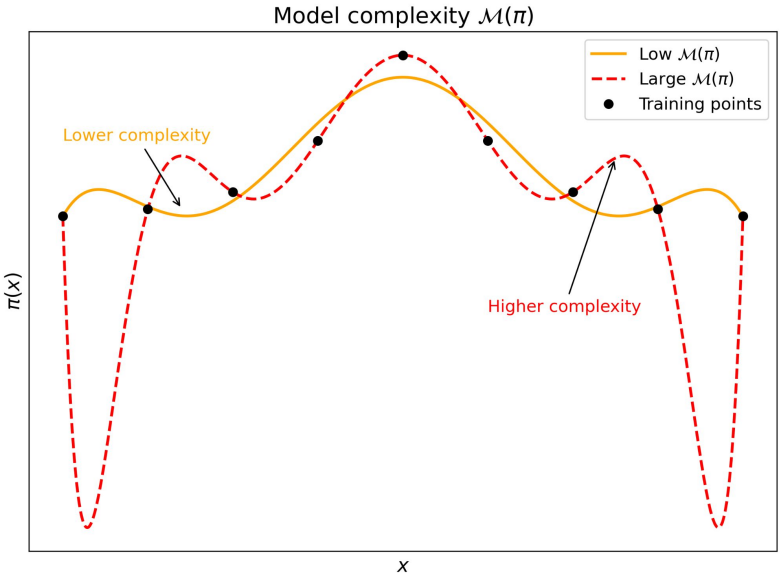
Complexity Measure[1]

$$\mathcal{M} : \pi \in \Pi \rightarrow \mathbb{R}_+$$

$\mathcal{M}(\pi)$ measures the **model complexity**

Robustness Measure

$$\mathcal{R}^{\pi} \leq \text{Bound}(\mathcal{M}(\pi))$$



Comparison of model complexity (polynomial class)

Question

Which complexity measures characterise robustness to noise?

[1]B. Neyshabur *et al.* "Exploring Generalization in Deep Learning", *NIPS* (2017)

Complexity Measure: Lipschitz Upper Bound

Policy $\longrightarrow$ $\pi_\theta(\cdot|x) \sim \mathcal{N}_{d_U}\left(\mu_\theta(x),\, \theta_{\sigma_\pi} I_{d_U}\right)$

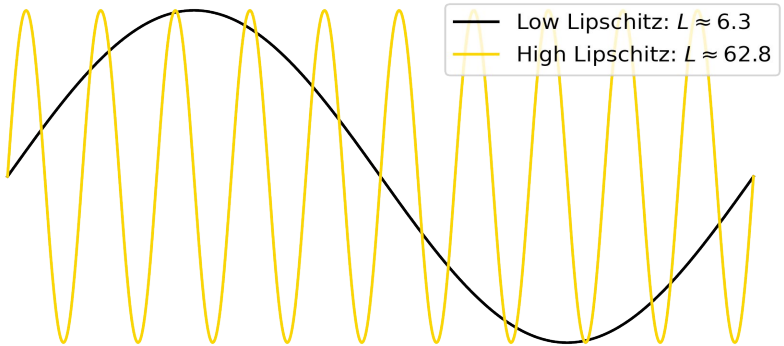
Neural Network $\longrightarrow$ $\mu_\theta\left(x\right) = \left(\sigma_l \circ \theta_l \circ \ldots \circ \sigma_1 \circ \theta_1\right)\left(x\right)$

Lipshitz Bound

$$\text{Lips}(\mu_\theta) \leq \prod_{i=1}^{l} \underbrace{\text{Lips}\left(\sigma_i\right)}_{=1} \prod_{i=1}^{l} \underbrace{\text{Lips}\left(\theta_i\right)}_{=\|\theta_i\|} = \prod_{i=1}^{l} \|\theta_i\|$$

Lipshitz-based Complexity Measure

$$\mathcal{M}(\pi_\theta) = \mathcal{M}(\mu_\theta) = \prod_{i=1}^{l} \|\theta_i\|$$



Comparison of Lipshitz constant for a trigonometric class of functions

Complexity Measure: Lipschitz Upper Bound

Results

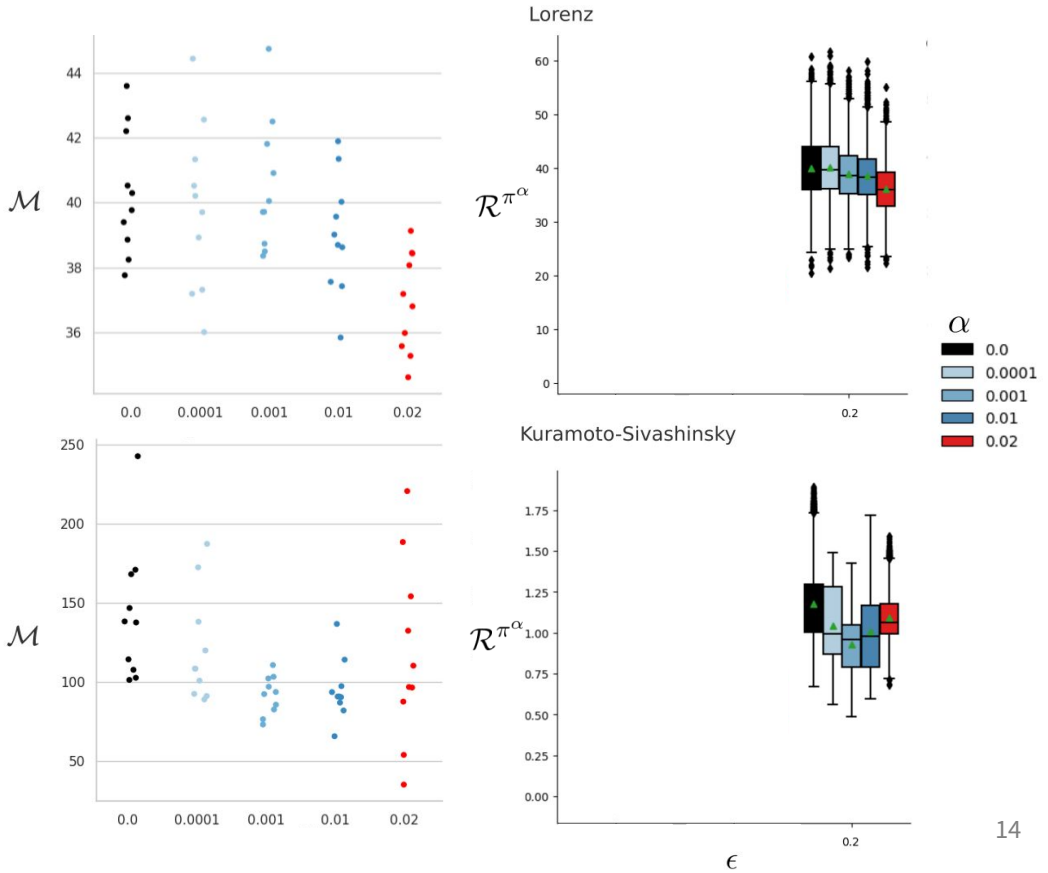
$\alpha > 0 \quad \longrightarrow \quad \mathcal{M}(\pi_\theta^\alpha) \searrow$

(Up to a threshold for KS)

$\mathcal{R}^{\pi^\alpha}$ and $\mathcal{M}(\pi_\theta)$ have **similar trend**

Lipshitz-based Complexity Measure

$$\mathcal{M}(\pi_\theta) = \mathcal{M}(\mu_\theta) = \prod_{i=1}^{l} \|\theta_i\|$$



Contributions and Perspectives

Contributions

- Robustness Metric: **Excess Risk Under Noise**
- Complexity Measure: **Model Regularity**
- Complexity Measure: **Optim. Landscape**

Results

Entropy $\longrightarrow$ Robustness to noise
Entropy $\longrightarrow$ **Regularity**

Evidence on the Regularisation Properties of Maximum-Entropy Reinforcement Learning

Rémy Hosseinkhan Boucher[1,2(✉)], Onofrio Semeraro[1,2], and Lionel Mathelin[1,2]

[1] Université Paris-Saclay, Orsay, France
{onofrio.semeraro,lionel.mathelin}@upsaclay.fr
[2] CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique, Orsay, France
remy.hosseinkhan@upsaclay.fr

Abstract. The generalisation and robustness properties of policies learnt through Maximum-Entropy Reinforcement Learning are investigated on chaotic dynamical systems with Gaussian noise on the observable. First, the robustness under noise contamination of the agent's observation of entropy regularised policies is observed. Second, notions of statistical learning theory, such as complexity measures on the learnt model, are borrowed to explain and predict the phenomenon. Results show the existence of a relationship between entropy-regularised policy optimisation and robustness to noise, which can be described by the chosen complexity measures.

Optimization and Learning: 7th International Conference, Revised Selected Papers (2024)

Perspectives

Mathematical analysis (theoretical Deep Reinforcement Learning)

Learning-Based Control: Sampling Strategies with Semi-Markov Decision Process

Model-based Control: Gaussian Process Modelling[1]

Controlled Markov Chain

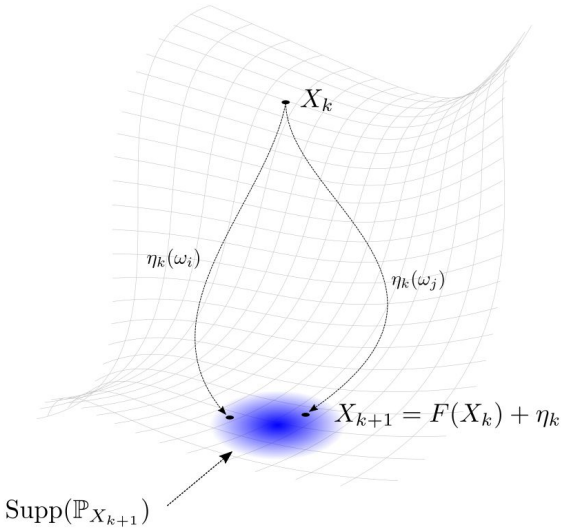
$$\mathbb{P}^{\pi}\left(dx_0 du_0 dx_1 du_1 \dots dx_T\right) = \mathbb{P}_{X_0}\left(dx_0\right)\pi\left(x_0, du_0\right)\mathcal{P}\left(dx_1 \mid x_0, u_0\right)$$
$$\pi\left(x_1 \mid du_1\right)\cdots\pi\left(x_{T-1} \mid du_{T-1}\right)\mathcal{P}\left(dx_T \mid x_{T-1}, u_{T-1}\right)$$

Transition Kernel $\mathcal{P}$

$(x,u) \in \mathcal{X}\times\mathcal{U} \quad \rightarrow \quad \mathcal{P}\left(dx' \mid x,u\right)$ Probability on the next state x'!

Learning Dynamics with Gaussian (Spatial) Process[1]

$$\hat{\mathcal{P}}_{\mathcal{D}}(\ \cdot\ , (x,u)) \sim \mathcal{N}\left(\mu_{(x,u)}, \Sigma_{(x,u),(x,u)} \mid \mathcal{D}\right)$$



[1] C. E. Rasmussen *et al.* "Gaussian Processes in Reinforcement Learning", *NIPS* (2003)

Learning-based Model Predictive Control

Model Predictive Control[1]

$$\pi^{\mathrm{MPC}}(x) = u_0^*$$

$$s.t. \quad (u_0^*, \ldots, u_{K^{\mathrm{MPC}}}^*) = \underset{(u_0, \ldots, u_{K^{\mathrm{MPC}}})}{\arg\min} \; \mathbb{E}^{(u_0, \ldots, u_{K^{\mathrm{MPC}}})} \left[\sum_{k=0}^{K^{\mathrm{MPC}}} c\left(X_k, u_k\right) \mid X_0 = x \right]$$

$$s.t. \quad X_{k+1} \sim \hat{\mathcal{P}}_{\mathcal{D}}\left(X_k, u_k\right)$$

Problem (Model Learning)

Fixed sampling budget $\longrightarrow \quad n$

$\mathcal{D}_n = \{(x_0, u_0, x_1), \ldots, (x_{n-1}, u_{n-1}, x_n)\}$

Learn $\hat{\mathcal{P}}_{\mathcal{D}} \simeq \mathcal{P}$

Question

Which **online sampling strategy**?

[1] L. Grune, J. Pannek - Nonlinear Model Predictive Control, *Springer* (2011)

Entropy Map

How to quantify the uncertainty on $X_{k+1} \sim \mathbb{P}_{X_{k+1}}$?

Infinitesimal volume element of $\mathcal{X} \longrightarrow dx$

Uncertainty on *dx*

$$I(dx) = \log(\frac{1}{\mathbb{P}_{X_{k+1}}(dx)})$$

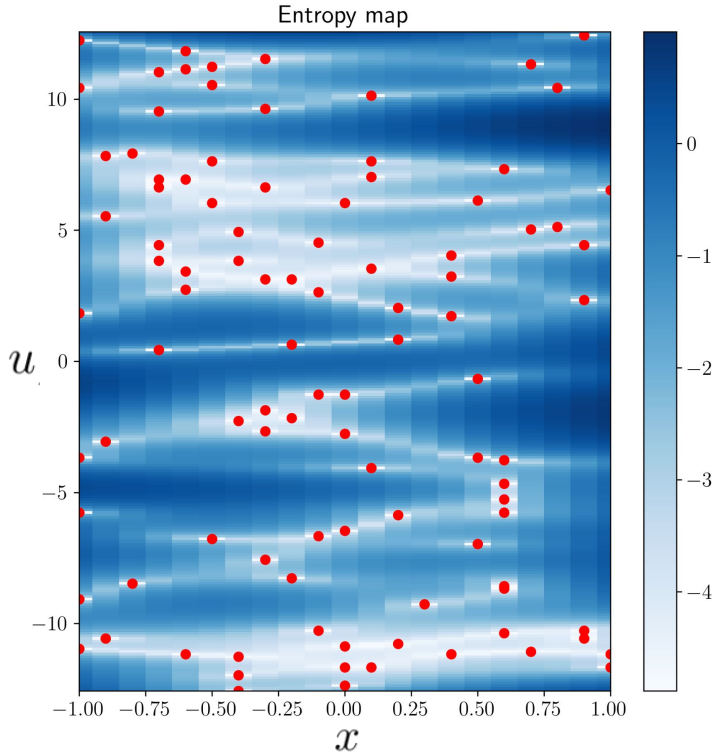
Entropy (average uncertainty)

$$\mathcal{H}(\mathbb{P}_{X_{k+1}}) = \int_{\mathbb{R}} \log \frac{1}{\mathbb{P}_{X_{k+1}}(dx)} \mathbb{P}_{X_{k+1}}(dx)$$

Gaussian case

$$\mathbb{P}_{X_{k+1}}(dx) = f_{\mathcal{N}(\mu,\Sigma)}(x)dx$$

$$\mathcal{H}(\mathbb{P}_{X_{k+1}}) = \tfrac{1}{2} \log\left(2\pi e \, \Sigma\right)$$



Red dots are observation contained in $\mathcal{D}_n$

$$\hat{\mathcal{P}}_{\mathcal{D}}(\ \cdot\ , (x,u)) \sim \mathcal{N}\left(\mu_{(x,u)}, \Sigma_{(x,u),(x,u)} \mid \mathcal{D}\right)$$

Expected Information Gain[1]

Dataset Construction (Sampling)

How to select the next data point (x, u) ?

$$\mathcal{D}_{n+1} = \mathcal{D}_n \cup (x, u, X_{n+1})$$

Process trajectory $\longrightarrow$ $H_T = (X_0, U_0, \ldots, U_T, X_T)$

Optimal trajectory under $\hat{\mathcal{P}}_{\mathcal{D}} \longrightarrow$ $\hat{H}_T^*$

Expected Information Gain (EIG) on the Optimal Trajectory[2]

$$\mathrm{EIG}(x, u) = \mathcal{H}[\hat{H}_T^* \mid \mathcal{D}_n] - \mathbb{E}_{\mathbb{P}_{X_{n+1}|\mathcal{D}_n,\, X_n = x,\, U_n = u}} \Big[\mathcal{H}[\hat{H}_T^* \mid \underbrace{\mathcal{D}_n, X_n = x, U_n = u, X_{n+1}}_{\mathcal{D}_{n+1}}]\Big]$$

By symmetry $\longrightarrow$ Uncertainty on X_{n+1}

$$\mathrm{EIG}_n(x, u) = \mathcal{H}\left[X_{n+1} \mid \mathcal{D}_n,\, X_n = x,\, U_n = u\right] - \mathbb{E}_{\mathbb{P}_{\hat{H}_T^*|\mathcal{D}_n}} \Big[\mathcal{H}[X_{n+1} \mid \mathcal{D}_n, X_n = x,\, U_n = u,\, \hat{H}_T^*]\Big]$$

[1] D. V. Lindley "On a measure of the Information Provided by an Experiment", *Chapel Hill and Berkley meetings of the Institute of Mathematical Statistics* (1955)

[2] V. Mehta *et al.* "An Experimental Design Perspective on Model-Based Reinforcement Learning", *ICLR* (2022)

Decision Epochs: Temporal Abstraction with Options[1]

Question

Data $\mathcal{D}_n = \{(x_0, u_0), \ldots, (x_n, u_n)\}$ is **collected** online **along the dynamics**

Which **online sampling strategy**?

Hypothesis

Exploit the **auto-correlation** of (X_{n+1}, U_{n+1}) from $\mathcal{D}_n$

Temporal Abstraction

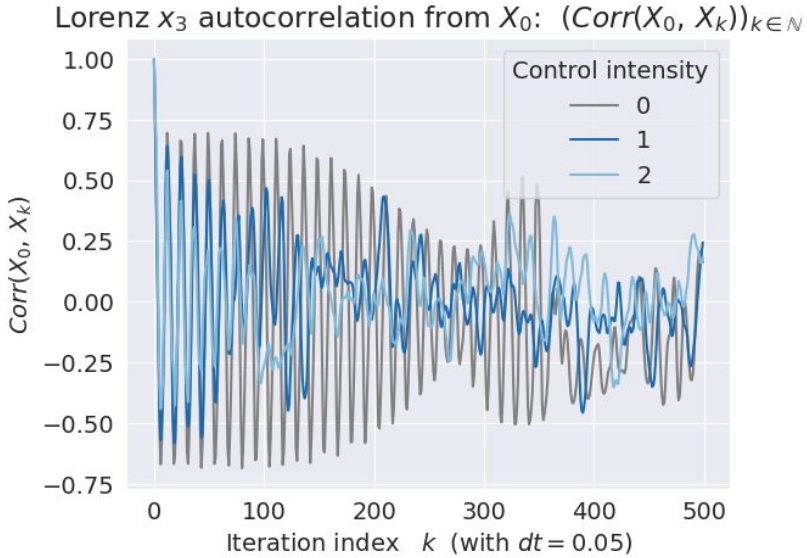
Irregular (Optional) Decision Epochs $\longrightarrow$ $(\kappa_j)_{j\in\mathbb{N}}$

Semi-Markov Decision Process $\longrightarrow$ $(X_{\kappa_j})_{j\in\mathbb{N}}$

Interdecision delay $\longrightarrow$ $\tau \in \mathcal{T}$

Constant control during $\tau \in \mathcal{T}$

New control space $\longrightarrow$ $\mathcal{U} \times \mathcal{T}$



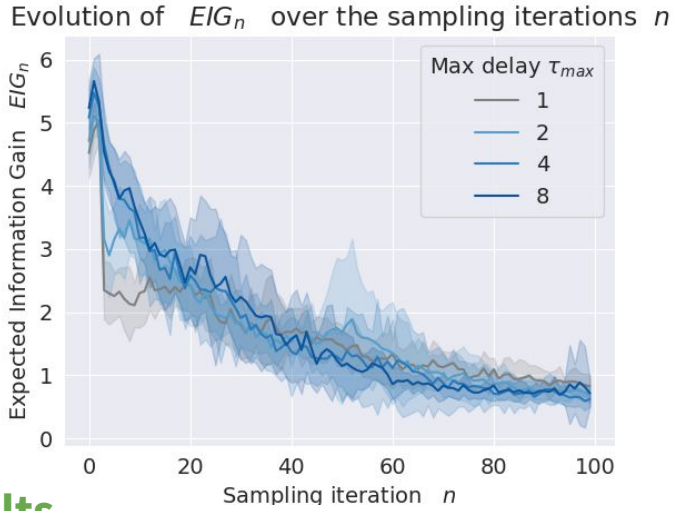
[1] R. S. Sutton *et al.* "Between MDPs and semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning", *NIPS* (1999)

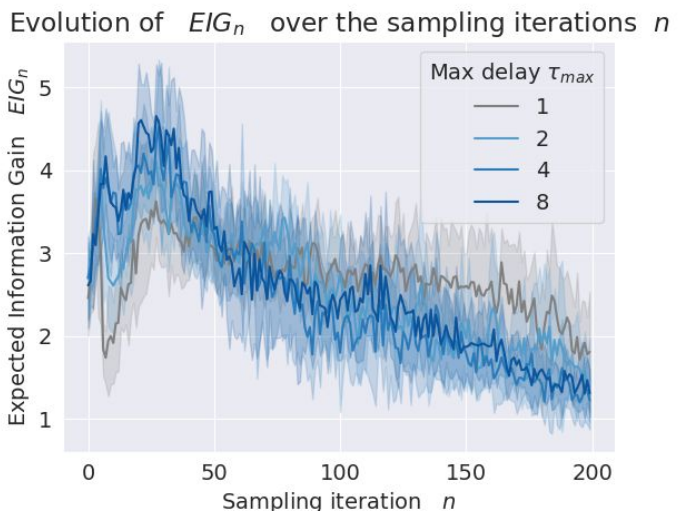
Semi-Markov Expected Information Gain

New criterion[1]

$$\mathrm{EIG}^{\mathrm{New}}(x,(u,\tau)) = \mathcal{H}\left[X_{\kappa_n+\tau+1} \mid \mathcal{D}_n,\, X_{\kappa_n+\tau}=x,\, U_{\kappa_n+\tau}=u,\, \kappa_n\right] - \mathbb{E}_{\mathbb{P}_{\hat{H}_T^*\mid\mathcal{D}_n}}\left[\mathcal{H}\left[X_{\kappa_n+\tau+1} \mid \mathcal{D}_n, X_{\kappa_n+\tau}=x,\, U_{\kappa_n+\tau}=u,\, \hat{H}_T^*,\, \kappa_n\right]\right]$$

Query the **interdecision time** to maximise information!





Results

Temporal Abstraction $\longrightarrow$ Information Gain

[1] R. Hosseinkhan-Boucher *et al.* "Increasing Information for Model Predictive Control with Semi-Markov Decision Processes", *L4DC* (2024)

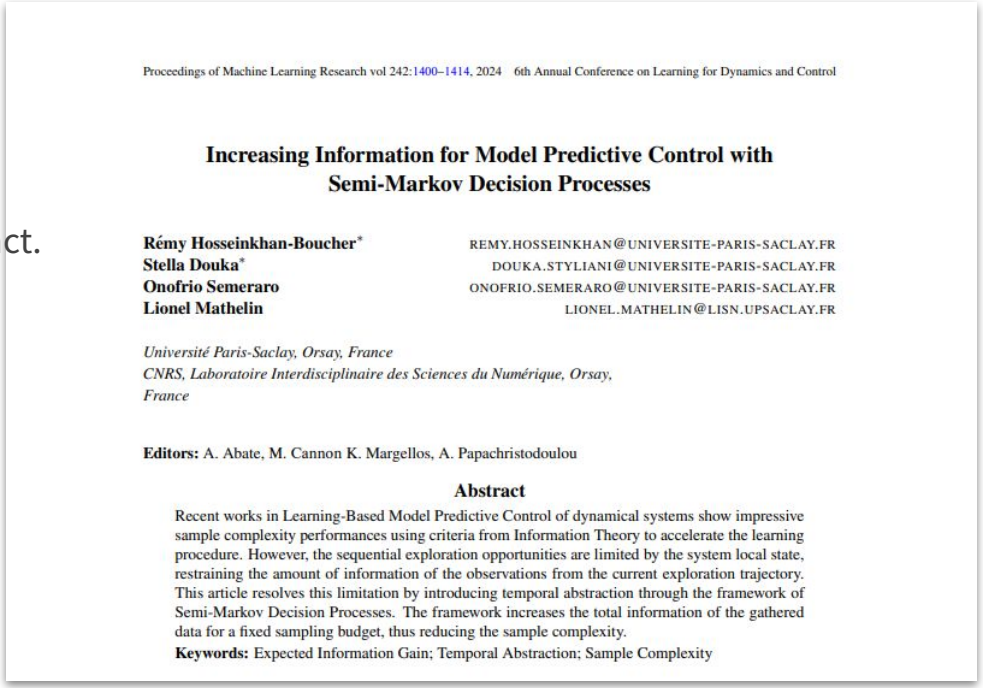
Contributions and Perspectives

Contributions

- Temporal Abstraction (Options framework)
- Extension of Information-based acquisition funct.

Results

Temporal Abstraction + EIG ⟶ Information gain
Information gain ⟶ Control performances



Proceedings of Machine Learning Research vol 242:1400–1414, 2024 6th Annual Conference on Learning for Dynamics and Control

Increasing Information for Model Predictive Control with Semi-Markov Decision Processes

Rémy Hosseinkhan-Boucher* REMY.HOSSEINKHAN@UNIVERSITE-PARIS-SACLAY.FR
Stella Douka* DOUKA.STYLIANI@UNIVERSITE-PARIS-SACLAY.FR
Onofrio Semeraro ONOFRIO.SEMERARO@UNIVERSITE-PARIS-SACLAY.FR
Lionel Mathelin LIONEL.MATHELIN@LISN.UPSACLAY.FR

Université Paris-Saclay, Orsay, France
CNRS, Laboratoire Interdisciplinaire des Sciences du Numérique, Orsay, France

Editors: A. Abate, M. Cannon K. Margellos, A. Papachristodoulou

Abstract

Recent works in Learning-Based Model Predictive Control of dynamical systems show impressive sample complexity performances using criteria from Information Theory to accelerate the learning procedure. However, the sequential exploration opportunities are limited by the system local state, restraining the amount of information of the observations from the current exploration trajectory. This article resolves this limitation by introducing temporal abstraction through the framework of Semi-Markov Decision Processes. The framework increases the total information of the gathered data for a fixed sampling budget, thus reducing the sample complexity.

Keywords: Expected Information Gain; Temporal Abstraction; Sample Complexity

Proceedings of the 6th Annual Learning for Dynamics & Control Conference, PMLR (2024)

Perspectives

Mathematical analysis (theoretical Gaussian Process MPC)

Towards Neural Controlled Delay Differential Equations for Model Based Control

Learning Model-based Continuous-time Control[1]

Model Learning

$$\partial_t x(t) = f(x(t), u(t)) \xrightarrow{\text{Neural net. } f_\theta} \partial_t x(t) = f_\theta(x(t), u(t)) \qquad \theta \in \Theta$$

Advantages

- Temporal abstraction
- Irregularly sampled data ⟶ **Robustness**
- Model-based ⟶ **Sample efficient**

Limitations

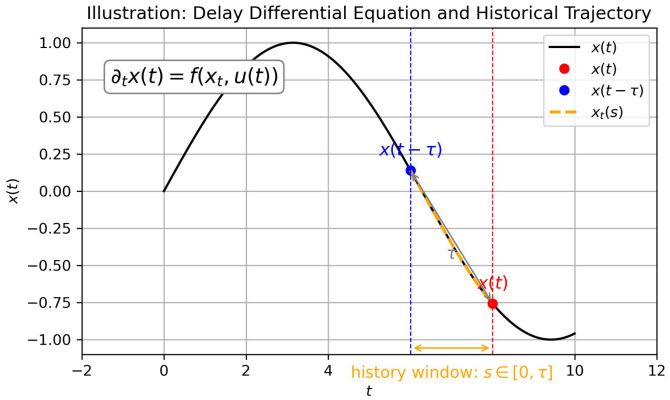
- **No partially-observed** system

$$\partial_t y(t) = g(x(t), u(t))$$

- No **delay** handling

$$\partial_t x(t) = f(x_t, u(t))$$

$$x_t : [0, \tau] \to \mathcal{X} \qquad s \mapsto x(t-s) \qquad \tau > 0$$



[1] C. Yildiz *et al.* "Continuous-Time Model-Based Reinforcement Learning", *ICML* (2021)

Learning Model-based Continuous-time Control[1]

Model Learning $\partial_t x(t) = f(x(t), u(t))$ $\xrightarrow{\text{Neural net. } f_\theta}$ $\partial_t x(t) = f_\theta(x(t), u(t))$ $\theta \in \Theta$

Advantages

- Temporal abstraction
- Irregularly sampled data ⟶ **Robustness**
- Model-based ⟶ **Sample efficient**

Limitations

- **No partially-observed** system

$$\partial_t y(t) = g(x(t), u(t))$$

- No **delay** handling

$$\partial_t x(t) = f(x_t, u(t))$$

$$x_t : [0, \tau] \to \mathcal{X} \qquad s \mapsto x(t-s) \qquad \tau > 0$$

Not Markovian

Dynamic Programming ⟶ Difficult

[1] C. Yildiz *et al.* "Continuous-Time Model-Based Reinforcement Learning", *ICML* (2021)

Delays and Partial Observability: Embedding

Solution ⟶ Markov Property in Larger Space

$H_{t+1} = (H_t, Y_{t+1}, U_{t+1})$ ⟶ Always Markov but **increasing dimension**

Information State[1,2]

$$I_{t+1} = \phi(I_t, Y_{t+1}, U_{t+1})$$

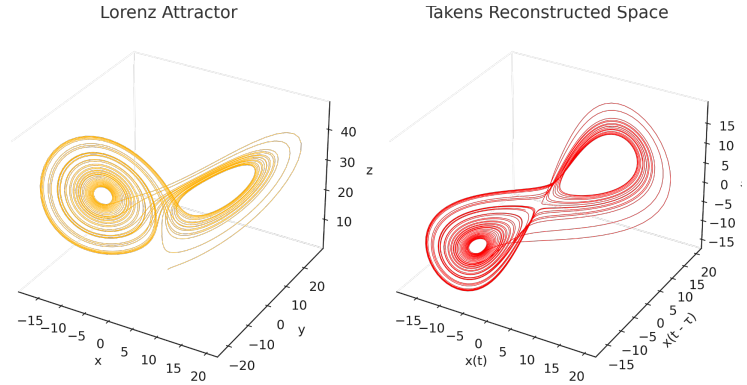
$\mathbb{P}(\cdot \mid H_t) = \mathbb{P}(\cdot \mid I_t)$ ⟶ Sufficient information

Takens theorem[3]

Theorem 1. Let M be a compact manifold of dimension m. For pairs (φ, y), $\varphi: M \to M$ a smooth diffeomorphism and $y: M \to \mathbb{R}$ a smooth function, it is a generic property that the map $\Phi_{(\varphi, y)}: M \to \mathbb{R}^{2m+1}$, defined by

$$\Phi_{(\varphi, y)}(x) = (y(x), y(\varphi(x)), \ldots, y(\varphi^{2m}(x))$$

is an embedding; by "smooth" we mean at least C^2.



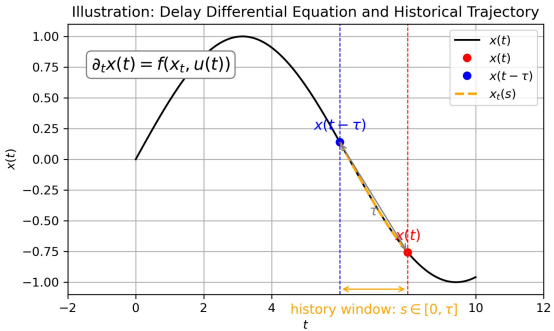
[1] O. Sigaud *et al.* - Processus décisionnels de Markov en intelligence artificielle, *Lavoisier* (2008)
[2] J. Subramanian *et al.* "Approximate Information State for Approximate Planning and Reinforcement Learning in Partially Observed Systems", *JMLR* (2022)
[3] F. Takens "Detecting Strange Attractors in Turbulence", *Dynamical Systems and Turbulence, Proceeding of a Warwick Symposium (1981)*

Learning Neural Delay Differential Equations (NDDE)

→ **implicit Information State**

Delay Differential Equations (DDE)[1]

$$\partial_t y(t) = g\big(y_t, u(t)\big) \quad \text{with} \quad y_t : [0, \tau] \to \mathcal{Y}, \quad y_t(s) = y(t-s)$$



Particular Case

$$\partial_t y(t) = g\big(y(t), y(t-\tau), u(t)\big)$$

Method → Neural Delay Differential Equations[2,3]

$$\partial_t y(t) = g_\theta\big(y(t), y(t-\tau), u(t)\big) \quad \theta \in \Theta \quad \tau \in \mathbb{R}_+ \quad (\tau \text{ is learnable})$$

Optimisation → **backpropagation** through **DDE solver**

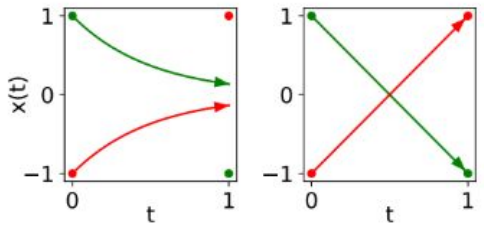


Figure 2: (Right) Two continuous trajectories generated by the DDEs are intersected, mapping -1 (resp., 1) to 1 (resp., -1), while (Left) the ODEs cannot represent such mapping.

[1] J. K. Hale - Functional Differential Equations, *Springer* (1971)
[2] Q. Zhu *et al.* "Neural Delay Differential Equation", *ICLR* (2021)
[3] T. Monsel *et al.* "Time and State Dependent Neural Delay Differential Equations", *PMLR - ECAI Workshop* (2024)

Learning NDDE: Ablation Study

Experiment

Compare 4 neural architectures

NODE $\longrightarrow$ $\partial_t y(t) = g_\theta\big(y(t)\big)$

NCDE $\longrightarrow$ $\partial_t y(t) = g_\theta\big(y(t), u(t)\big)$

NDDE $\longrightarrow$ $\partial_t y(t) = g_\theta\big(y(t), y(t-\tau)\big)$

NCDDE $\longrightarrow$ $\partial_t y(t) = g_\theta\big(y(t), y(t-\tau), u(t)\big)$

System: Van der Pol Oscillator

$$\partial_t x^1(t) = x^2(t) + u^1(t)$$
$$\partial_t x^2(t) = \epsilon_{\text{VDP}}(1 - (x^1(t))^2)x^2(t) - x^1(t) + u^2(t)$$

$$y(t) = x(t-\tau)$$

Collect dataset of trajectories $(y_t)_{t\in I}$
with **noisy control** $u(t)$
and $\tau \in \{0, 0.1\}$

Compare 4 neural architectures $\longrightarrow$

Generate predictive data (time series) $\longrightarrow$ $(y_t^\theta)_{t\in I}$

Minimise over $\theta \in \Theta$ $\longrightarrow$ $\mathcal{L} = \mathbb{E}^{\mathbb{P}_{(y_t)_{t\in I}}} \left\| (y_t)_{t\in I} - (y_t^\theta)_{t\in I} \right\|_{L^2}^2$

Hypothesis

$u(t)$-dependant models $\longrightarrow$ $\mathcal{L}$ $\searrow$

$y(t-\tau)$-dependant models fit better $\longrightarrow$ $\mathcal{L}$ $\searrow$

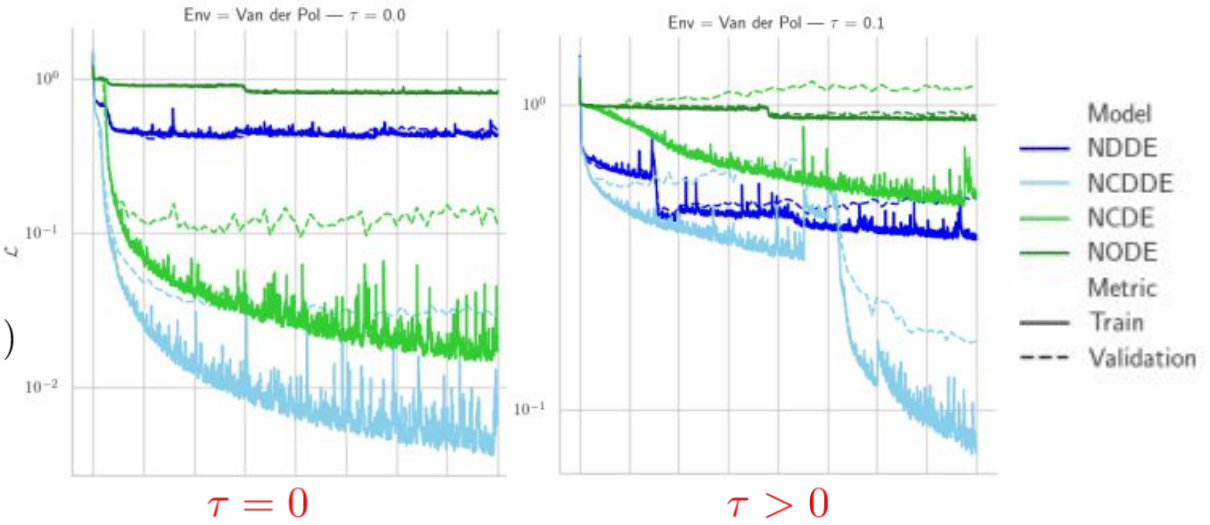
Learning NDDE: Ablation Study - Results

NODE $\longrightarrow$ $\partial_t y(t) = g_\theta(y(t))$

NCDE $\longrightarrow$ $\partial_t y(t) = g_\theta(y(t), u(t))$

NDDE $\longrightarrow$ $\partial_t y(t) = g_\theta(y(t), y(t-\tau))$

NCDDE $\longrightarrow$ $\partial_t y(t) = g_\theta(y(t), y(t-\tau), u(t))$



$\tau = 0$ $\tau > 0$

Results

$u(t)$-dependant models $\longrightarrow$ $\mathcal{L}$ $\searrow$

$y(t-\tau)$-dependant models fit better $\longrightarrow$ $\mathcal{L}$ $\searrow$

$y(t-\tau)$-dependant models also improve when $\tau = 0$

Drawback

Increased computational complexity

Contributions and Perspectives

Contributions

- Functional Differential Equation framework in Continuous-time model-based control
- Links with Information States and Dynamical Systems theory
- Controlled Neural Differential Equations

Results

Delay Differential Equation $\longrightarrow$ **Better regression performances**

Perspectives

Continuous-time Dynamic Programming in **Infinite Dimensional Spaces?**

Academic and Scientific Involvement

Academic and Scientific Involvement

Internship supervision 🕵🏻

- *Information-driven learning-based MPC* (*with* S. Douka)
- *Learning-based Functional Dynamic Programming* (*with* E. Pradeleix)

Teaching 👨🏻‍🏫

- *C++* (Université Paris-Saclay)
- *Data Science Project* (CentraleSupélec)
- *Advanced Deep Learning* (ENS Paris-Saclay MVA)

Reviews 👨🏻‍💻

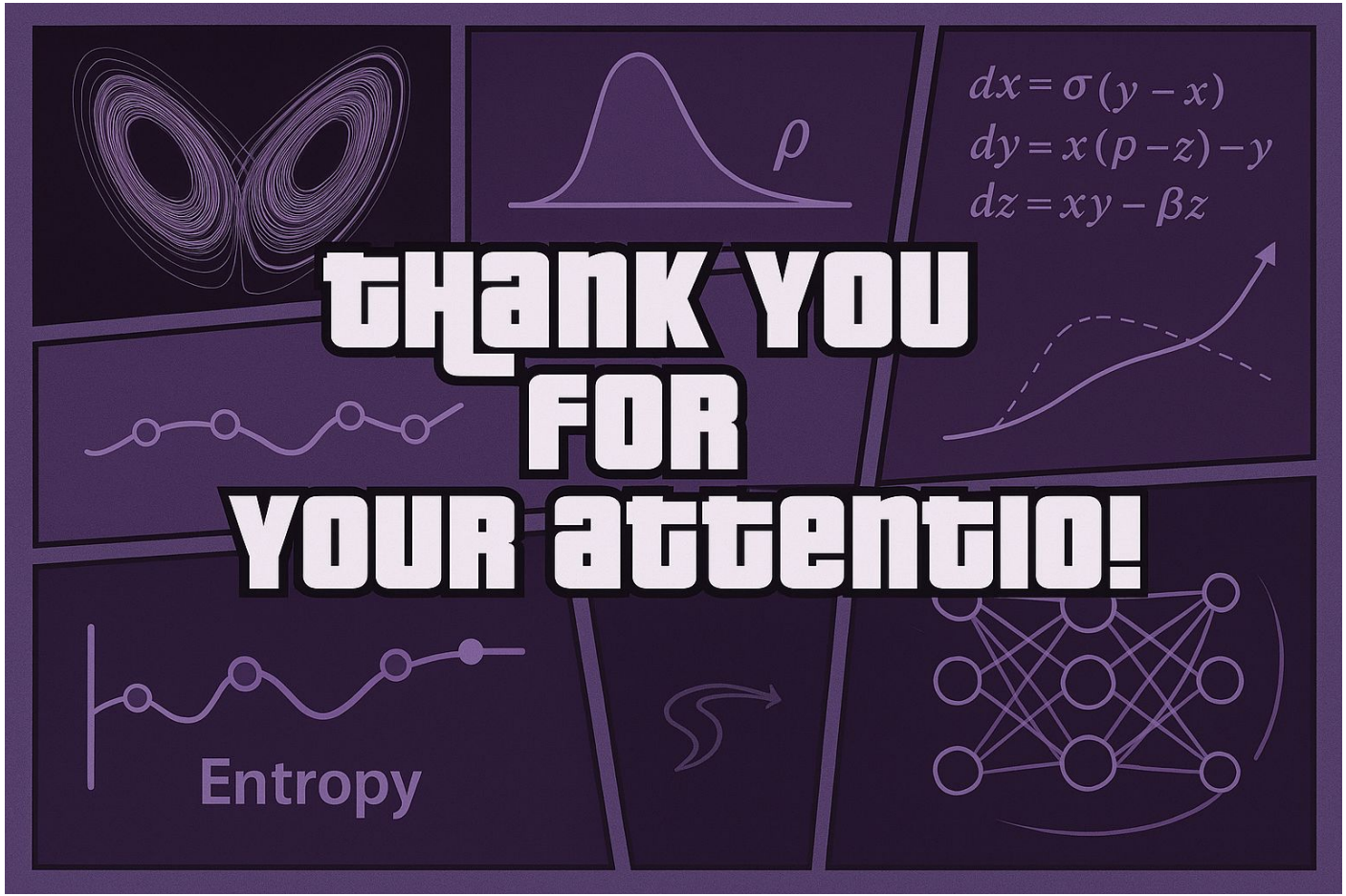
- *Transaction in Automatic and Control*
- *Journal of Fluid Mechanics*
- *European Workshop in Reinforcement Learning*

Contribution in open-source packages 👨🏻‍💻

- `control_dde` (*with* T. Monsel)
- `stable_baselines3`
- `hydrogym`
- `torchdde`

General Conclusion

- **Interdisciplinary Research**
- **Contribution to Learning-based Control Challenges**
- **Brings concepts from various fields and a novel viewpoint**



Generated with OpenAI Dall-E 3

Learning-based Control of Dynamical Systems: Challenges

Complexity Measure: Conditional Fisher Information

Hessian and Fisher Information

Objective Hessian $\longrightarrow$ $$\nabla_\theta^2 J^{\pi_\theta} = \mathbb{E}^{\pi_\theta}\left[\sum_{h,i,j=0}^{T} c(X_h, U_h)\left(\nabla_\theta \log \pi_\theta(U_i \mid X_j)\nabla_\theta \log \pi_\theta(U_j \mid X_j)^T + \nabla_\theta{}^2\left[\log \pi_\theta(U_i \mid X_j)\right]\right)\right]$$

Fisher Information $\longrightarrow$ $$\mathcal{I}(\theta) = -\mathbb{E}^{X\sim\rho, U\sim\pi_\theta(\cdot|X)}\left[\nabla_\theta^2 \log \pi_\theta(U|X)\right]$$

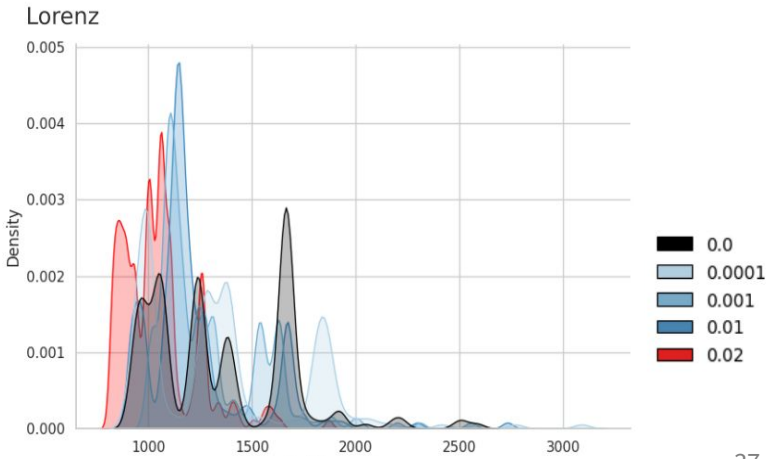
Fisher Information Complexity Measure

$$\mathcal{M}(\pi_\theta) = -\operatorname{Tr}\left(\mathbb{E}^{X\sim\rho^{\pi_\theta}, U\sim\pi_\theta(\cdot|X)}\left[\nabla_{\theta_\mu}^2 \log \pi_\theta(U|X)\right]\right)$$

Results

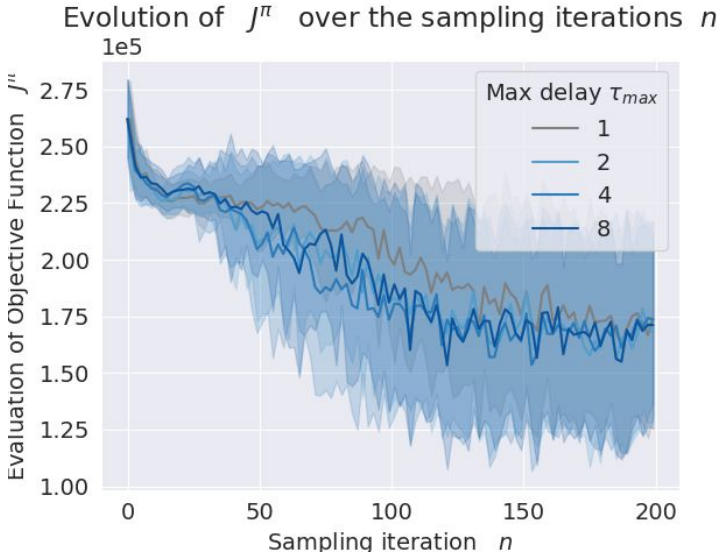
$\alpha = 0$ $\longrightarrow$ Right Leptokurtic distribution

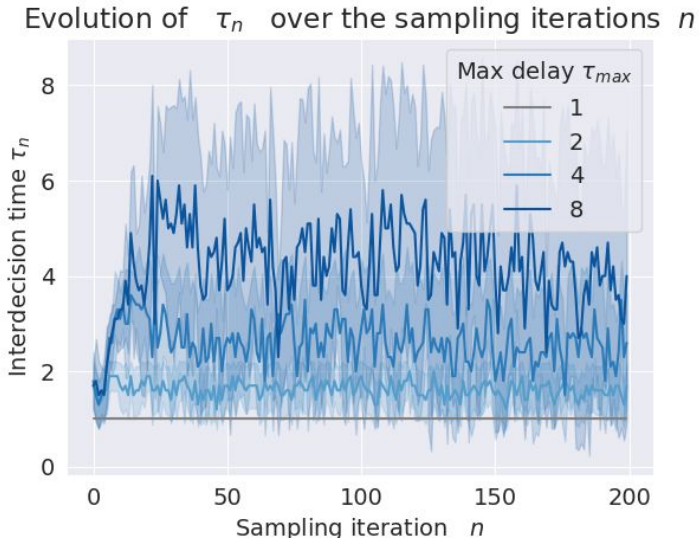
$\alpha > 0$ $\longrightarrow$ Relatively less extreme values



Density of $\mathcal{M}(\pi_\theta(\cdot|X)) = -\operatorname{Tr}\left(\mathbb{E}^{U\sim\pi_\theta(\cdot|X)}\left[\nabla_{\theta_\mu}^2 \log \pi_\theta(U|X)\right]\right)$ *where* $X \sim \rho$

Objective Function and Inter-decision time





Results

Temporal Abstraction ⟶ Information Gain ⟶ Control performances

Placeholder

Placeholder

Placeholder

[1]C. E. Rasmussen et al. "Gaussian Processes in Reinforcement Learning" *NIPS* (2003)