Physics-Informed Neural Network Policy Iteration: Algorithms, Convergence, and Verification

Yiming Meng*†, Ruikun Zhou*‡, Amartya Mukherjee‡, Maxwell Fitzsimmons‡, Christopher Song‡, and Jun Liu‡

* Equal Contribution

† Coordinated Science Laboratory, University of Illinois Urbana-Champaign

‡ Department of Applied Mathematics, University of Waterloo

UNIVERSITY OF WATERLOO

ILLINOIS

ICML International Conference On Machine Learning

July 21st – 27th, 2024, Vienna, Austria

Abstract

Solving nonlinear optimal control problems is a challenging task, particularly for high-dimensional problems. We propose algorithms for model-based policy iterations to solve nonlinear optimal control problems, ensuring convergence through an iterative procedure that uses neural approximations for linear PDEs.

We present two variants of the algorithms.

- ELM-PI, which can handle low-dimensional problems with high accuracy.
- PINN-PI, which has the potential to address high-dimensional problems.

Background

- Consider a control-affine system:

$$\dot{x} = f(x) + g(x)u, \; x(0) = x_0. \qquad (1)$$

  - We assume f, g are continuously differentiable and $f(0) = 0$.
  - We work on a compact region of interest Ω.
- We aim to compute optimal value and control w.r.t. the associated cost

$$J(x_0, u) = \int_0^\infty Q(\phi(t, x_0; u)) + u^T R(\phi(t, x_0; u))udt,$$

  where $\phi(t, x_0, u)$ is the solution to (1), and $Q(\cdot)$ & $R(\cdot)$ are positive definite.

  The optimal control u^* is such that $V^*(x) = J(x, u^*) = \inf_{u \in \mathcal{U}} J(x, u)$, which should solve the (nonlinear) HJB equation.
- **Policy iteration** method approaches the optimal value by iteratively solving a simpler (linear) Lyapunov-type PDE, with $u = \kappa_0(x)$:

  1) [policy evaluation] Solve the (GHJB) PDE subject to $V_i(0) = 0$:

$$F(x, V_i(x), DV_i(x)) := DV_i(x) \cdot (f(x) + g(x)\kappa_i(x)) + Q(x) - \kappa_i(x)^T R(x)\kappa_i(x) = 0\,; \qquad (2)$$

  2) [policy improvement] Update the controller using

$$\kappa_{i+1}(x) = -\frac{1}{2} R^{-1}(x) g^T(x) DV_i^T(x). \qquad (3)$$

  The limit value function $V^*(x)$ is a *Lyapunov function*.

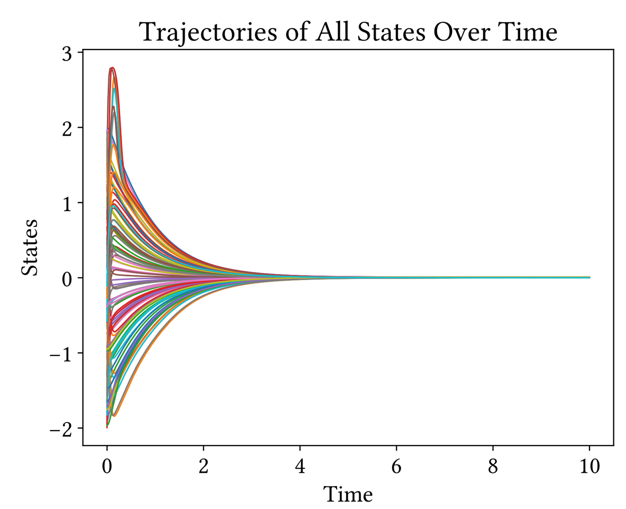
Motivation

- In previous works [1,2], V_i's and V^* were assumed to be C^1.
  - Conditions not verified.
  - $V_i \xrightarrow{\|\cdot\|_\infty} V^*$ may not be guaranteed.
- The solution may exhibit non-differentiability.
- The main goal is to answer the following questions:
  - Can neural approximations converge to the viscosity solution of the HJB?
  - Can neural approximations efficiently compute solutions of the HJB with high accuracy?
  - Can neural policy iteration overcome the curse of dimensionality?
  - Can neural approximations be guaranteed to lead to stabilizing controllers?

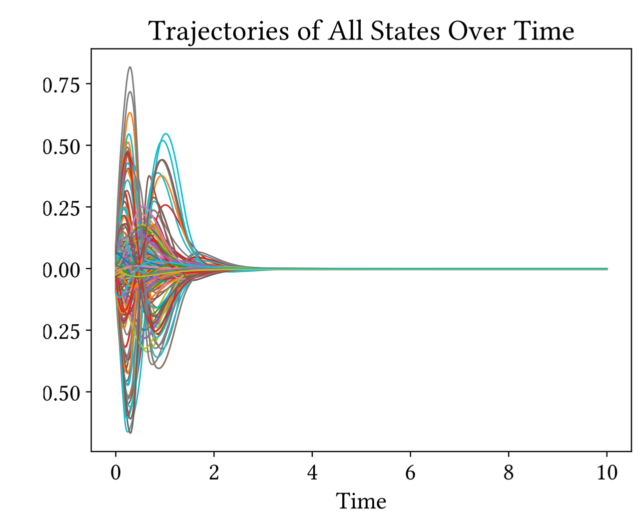
Numerical Experiments

- **Trajectories starting from different initial conditions under the optimal controller learned using PINN-PI**

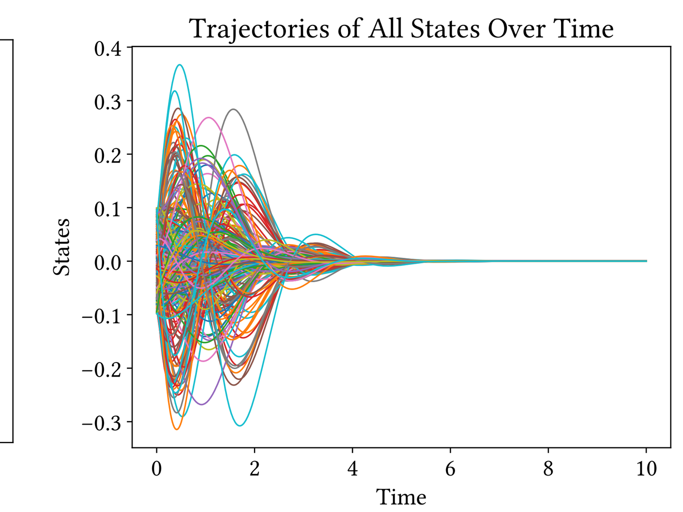
Inverted Pendulum



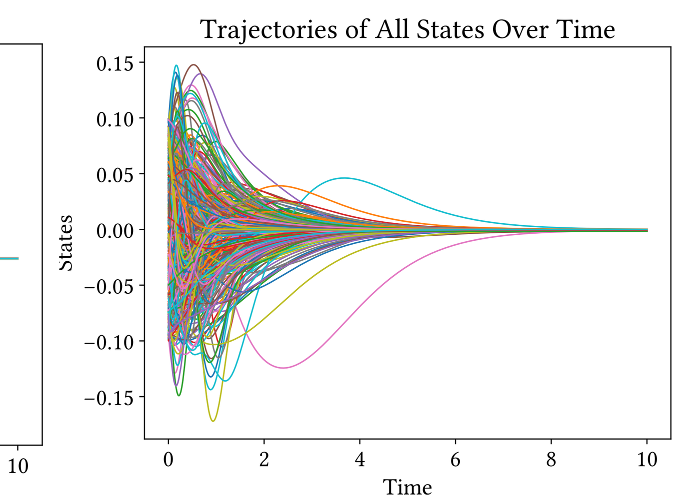
Cartpole



2D Quadrotor

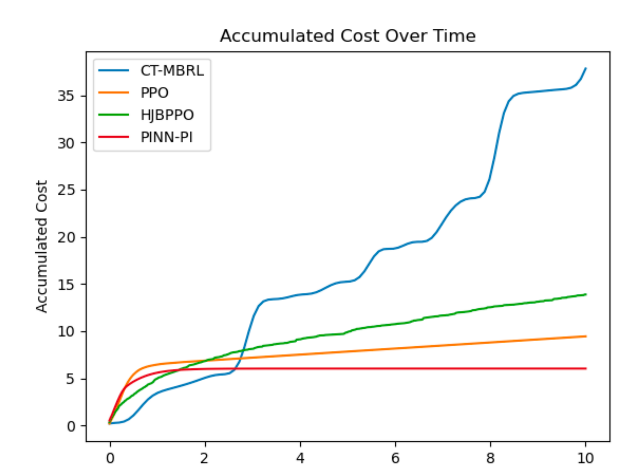


3D Quadrotor

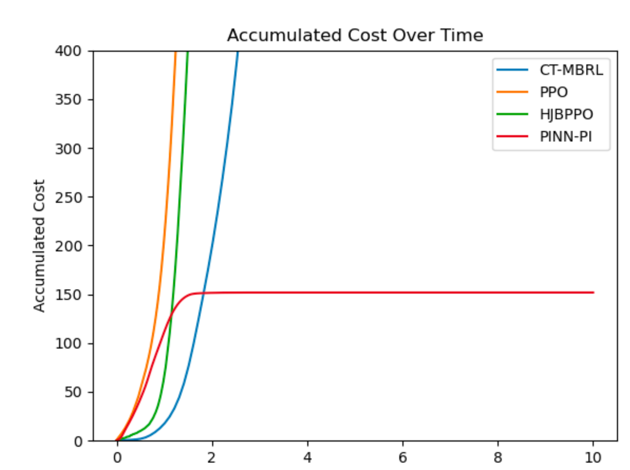


- **Comparison with reinforcement learning algorithms**

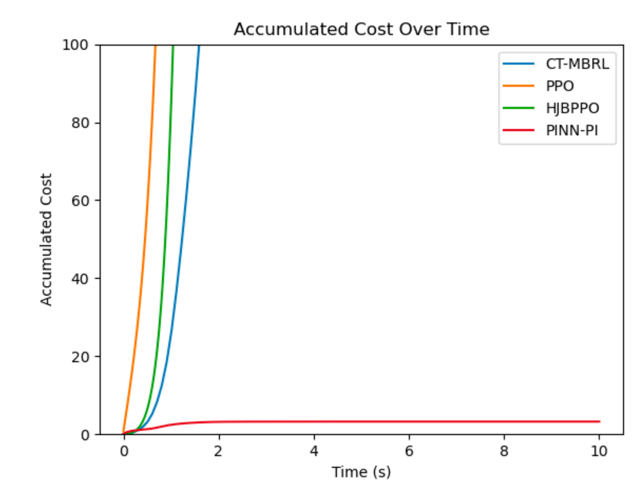
Inverted Pendulum



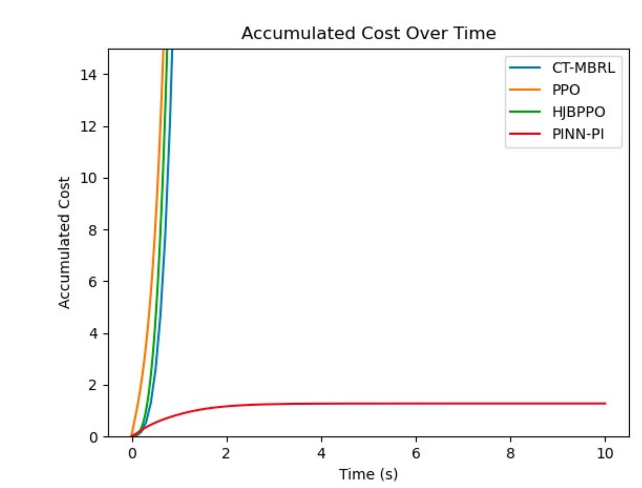
Cartpole



2D Quadrotor



3D Quadrotor



- **Comparison of ELM-PI, PINN-PI, and successive Galerkin algorithm (SGA) on the inverted pendulum example.**

SGA			ELM-PI			PINN-PI		
Order	Time (s)	Verified?	m	Time (s)	Verified?	m	Time (s)	Verified?
2	4.80	Yes	50	**0.11**	Yes	50	255.15	Yes
4	19.37	Yes	100	**0.24**	Yes	100	256.53	Yes
6	66.52	Yes	200	**0.71**	Yes	200	258.89	Yes
8	212.42	Yes	400	**2.92**	Yes	400	256.52	Yes

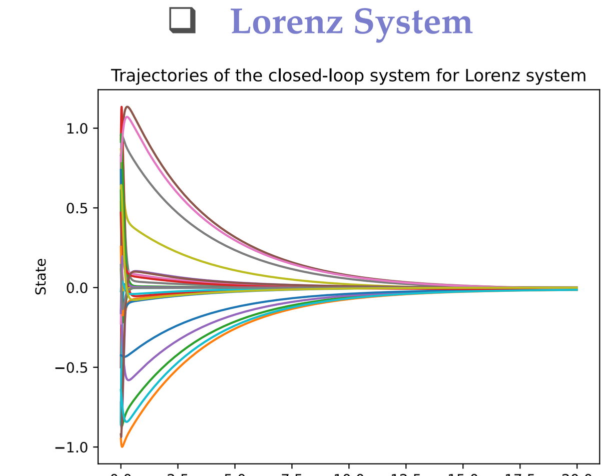
- **Performance of ELM-PI and PINN-PI on a synthetic n-dimensional nonlinear problem.**

Problem & model size			ELM-PI		PINN-PI	
n	m	N	Error	Time (s)	Error	Time (s)
5	800	4800	5.36E00	32.22	3.63E-02	770.54
5	3200	16000	3.08E-01	6303.20	5.20E-02	1204.22
5	6400	32000	6.37E-02	53479.10	9.10E-02	5906.69
6	800	4800	8.76E00	39.03	4.31E-02	800.76
6	6400	38400	2.33E00	63403.53	1.38E-01	6995.10
7	800	100000	–	–	3.74E-02	4414.28
8	800	100000	–	–	5.28E-02	4415.12
9	800	100000	–	–	4.88E-02	4422.54
10	800	100000	–	–	3.66E-02	4424.33
11	800	100000	–	–	8.29E-02	4424.00
12	800	100000	–	–	6.11E-02	4426.54

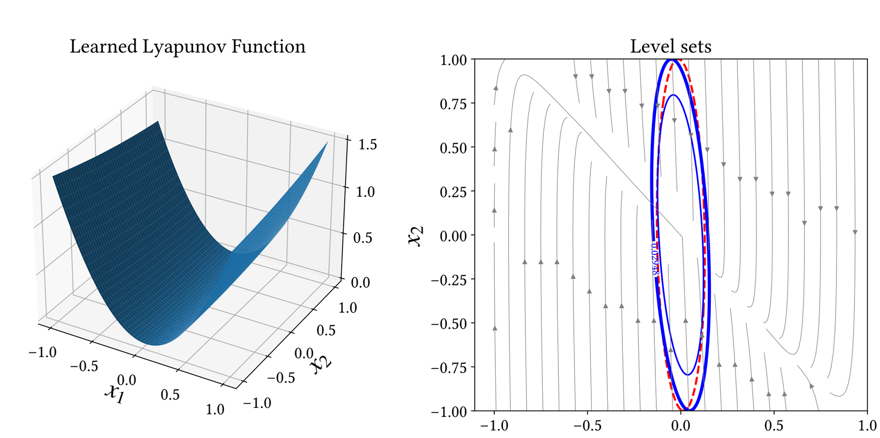
- ***n*** represents the dimension of the problem ;
- ***m*** denotes the number of hidden units ;
- ***N*** indicates the number of collocation points.

- **Miscellaneous**

Lorenz System



Inverted Pendulum



Algorithm

- **PINN-PI:**
  - We use a PINN to solve the first-order PDE (2) for each i:

$$F(x, V_i(x), DV_i(x)) = 0, \quad x \in \Omega,$$

    subject to the boundary condition $V_i(0) = 0$.

  Goal: find a physics-informed neural network solution $V_{i,N}(x;\theta)$ by minimizing a loss:

$$\text{Loss}(\theta) = \mathcal{L}_{\text{residual}}(\theta) + \mathcal{L}_{\text{boundary}}(\theta) + \mathcal{L}_{\text{data}}(\theta).$$

    - $\mathcal{L}_{\text{residual}}$: the residual error of the PDE;
    - $\mathcal{L}_{\text{boundary}}$: the boundary error;
    - $\mathcal{L}_{\text{data}}$: the data error or any side information desirable.
  - Update $\kappa_{i+1}(x)$ by (3) for $x \in \Omega \setminus \{0\}$ and set $\kappa_{i+1}(x) = 0$ elsewhere.
  - Iterate until the desired accuracy or the maximum number of iterations is reached.
- **ELM-PI:**
  - Consider $V_{i,N}(x;\theta) := \theta^T \sigma(Wx + b)$.
  - The other steps remain the same.
- **Formal verification:**

  We verify the Lyapunov conditions are satisfied (everywhere in Ω) with a satisfiability modulo theories (SMT) solver [3] within the tool LyZNet [5].

Theoretical Analysis

- **Exact-PI:**
  - For each i, Eq.(2) has a unique positive definite viscosity solution $V_i \in C(\Omega) \cap C^1(\Omega \setminus \{0\})$.
  - $V^* \le V_{i+1} \le V_i$ on Ω for all $i \in \{0, 1, \cdots\}$.
  - $V_i \xrightarrow{\|\cdot\|_\infty} V^*$.
- **PINN (ELM) - PI:**

  Let $\{V_i\}$ and $\{\kappa_{i+1}\}$ be updated by exact-PI.

  Let $\{\hat{V}_i\}$ and $\{\hat{\kappa}_{i+1}\}$ be updated by PINN-PI or ELM-PI with $\hat{\kappa}_0 = \kappa_0$. Then, for any i and $\theta > 0$, we can achieve

$$|\hat{V}_i(x) - V_i(x)| \le \theta, \; |\hat{\kappa}_{i+1}(x) - \kappa_{i+1}(x)| \le \theta, \; \forall x \in \Omega.$$

- **Remark:**
  - The proof shows the absolute continuity of the neural solutions to the training errors on $\Omega \setminus B_\varepsilon(0)$ for any $\varepsilon > 0$.
  - The convergence can be shown on $\Omega \setminus B_\varepsilon(0)$ using absolute continuity and on $B_\varepsilon(0) \setminus \{0\}$ given the boundedness.

Appendix: Lyapunov Stability Theorem

Suppose there exists a $V \in C^1(\Omega)$ that satisfies the conditions:

- $V(0) = 0$;
- $V(x) > 0$ and $DV(x) \cdot f(x) < 0, \forall x \in \Omega \setminus \{0\}$.

Then the origin is an asymptotically stable equilibrium point.

References

[1] Randal W. Beard, George N. Saridis, and John T. Wen. Galerkin Approximation of the Generalized Hamilton-Jacobi-Bellman Equation. *Automatica, 33.12 (1997), 2159-2177.*

[2] Yu Jiang and Zhong-Ping Jiang. Robust adaptive dynamic programming. *John Wiley & Sons, (2017).*

[3] Sicun Gao, Soonho Kong, and Edmund M Clarke. dReal: an SMT solver for nonlinear theories over the reals. In *Proceedings of International Conference on Automated Deduction, (2013): 208–214.*

[4] Jun Liu, Yiming Meng, Maxwell Fitzsimmons, and Ruikun Zhou. Physics-Informed Neural Network Lyapunov Functions: PDE Characterization, Learning, and Verification. *arXiv preprint arXiv:2312.09131* (2023).

[5] Jun Liu, Yiming Meng, Maxwell Fitzsimmons, and Ruikun Zhou. LyZNet: A Lightweight Python Tool for Learning and Verifying Neural Lyapunov Functions and Regions of Attraction. In *Proceedings of HSCC* (2024).