Initial Value Problem Enhanced Sampling for Closed-Loop Optimal Control Design with Deep Neural Networks

Jiequn Han

Flatiron Institute, Simons Foundation
collaboration with Xuanxi Zhang, Jihao Long, Wei Hu, Weinan E

4th AFOSR Monterey Workshop on
Computational Issues in Nonlinear Control
May 2023

Open-Loop and Closed-Loop Optimal Control

Consider the following optimal control problem:

$$\min_{\boldsymbol{x}(t),\boldsymbol{u}(t)} \int_0^T L(t,\boldsymbol{x}(t),\boldsymbol{u}(t))\mathrm{d}t + M(\boldsymbol{x}(T))$$
$$\text{s.t. } \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t,\boldsymbol{x}(t),\boldsymbol{u}(t)),\ \boldsymbol{x}(0) = \boldsymbol{x},$$

$\boldsymbol{x}(t)$ = state, $\boldsymbol{u}(t)$ = control.

- Open-loop optimal control: find the optimal path $(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t))$ for a specific initial point.
- Closed-loop optimal control: find the optimal policy function $\boldsymbol{u}^*(t,\boldsymbol{x})$, applicable for a set of initial points $x(0) \in \mathcal{X}$. More powerful, but more difficult to solve.
- Traditional methods by solving the assoicated Hamilton-Jacobi-Bellman equation suffers from the curse of dimensionality.

Neural Network-Based Closed-Loop Control

Neural networks have demonstrated astonishing capability in dealing with high-dimensional functions.

How to use neural networks to design closed-loop optimal control for high-dimensional problems?

Neural Network-Based Closed-Loop Control

Neural networks have demonstrated astonishing capability in dealing with high-dimensional functions.

How to use neural networks to design closed-loop optimal control for high-dimensional problems?

Approach 1: direct policy search (Han and E (2016), Böttcher, Antulov-Fantulin and Asikis (2022))

$$\min_{\theta} \mathbb{E} \int_0^T L(t, \boldsymbol{x}(t), \boldsymbol{u}^{\mathbf{NN}}(t, \boldsymbol{x}(t); \theta)) \mathrm{d}t + M(\boldsymbol{x}(T))$$
$$\text{s.t. } \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}^{\mathbf{NN}}(t, \boldsymbol{x}(t); \theta)), \ \boldsymbol{x}(0) \sim \mu_0,$$

Neural Network-Based Closed-Loop Control

Neural networks have demonstrated astonishing capability in dealing with high-dimensional functions.

How to use neural networks to design closed-loop optimal control for high-dimensional problems?

Approach 1: direct policy search (Han and E (2016), Böttcher, Antulov-Fantulin and Asikis (2022))

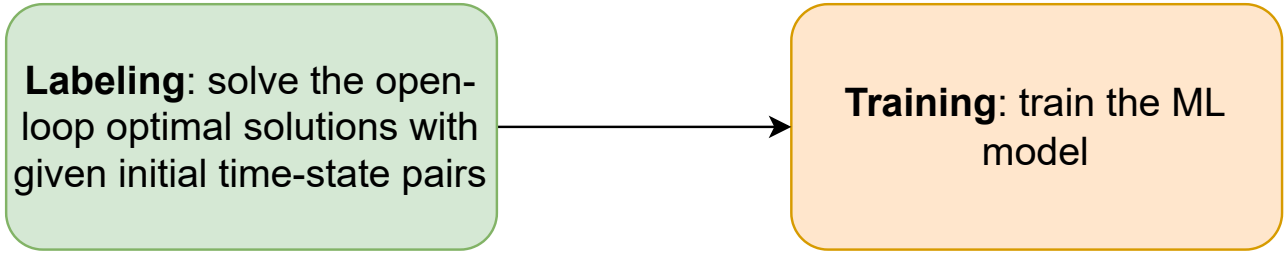
$$\min_{\theta} \mathbb{E} \int_0^T L(t, \boldsymbol{x}(t), \boldsymbol{u}^{\mathbf{NN}}(t, \boldsymbol{x}(t); \theta)) \mathrm{d}t + M(\boldsymbol{x}(T))$$
$$\text{s.t. } \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}^{\mathbf{NN}}(t, \boldsymbol{x}(t); \theta)), \ \boldsymbol{x}(0) \sim \mu_0,$$

Approach 2: supervised learning (Nakamura-Zimmerer, Gong, and Kang (2021))

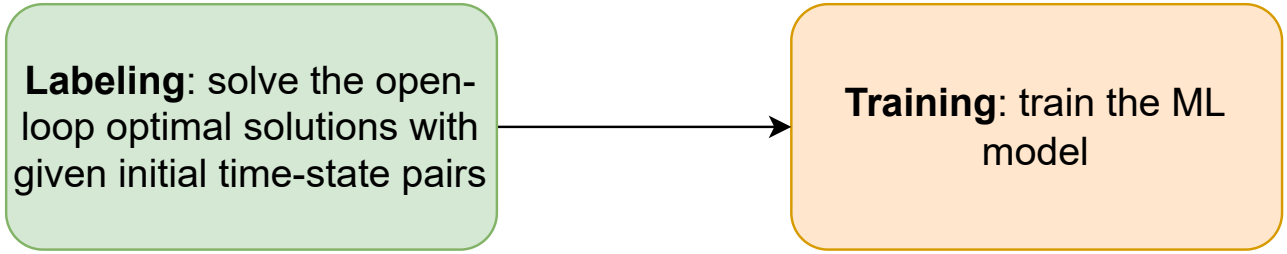
- Sample initial points and solve the corresponding open-loop optimal control problems.
- Choose the time-state-control tuples along every paths to collect the training data: $\mathcal{D} = \{(t^i, \boldsymbol{x}^i), \boldsymbol{u}^i\}_{1 \le i \le M}$.
- Train a neural network to approximate the closed-loop optimal control $\boldsymbol{u}^{\mathbf{NN}}$:

$$\min_{\theta} \sum_{i=1}^{M} \|\boldsymbol{u}^i - \boldsymbol{u}^{\mathbf{NN}}(t^i, \boldsymbol{x}^i; \theta)\|^2.$$

Supervised-Learning-Based Approach

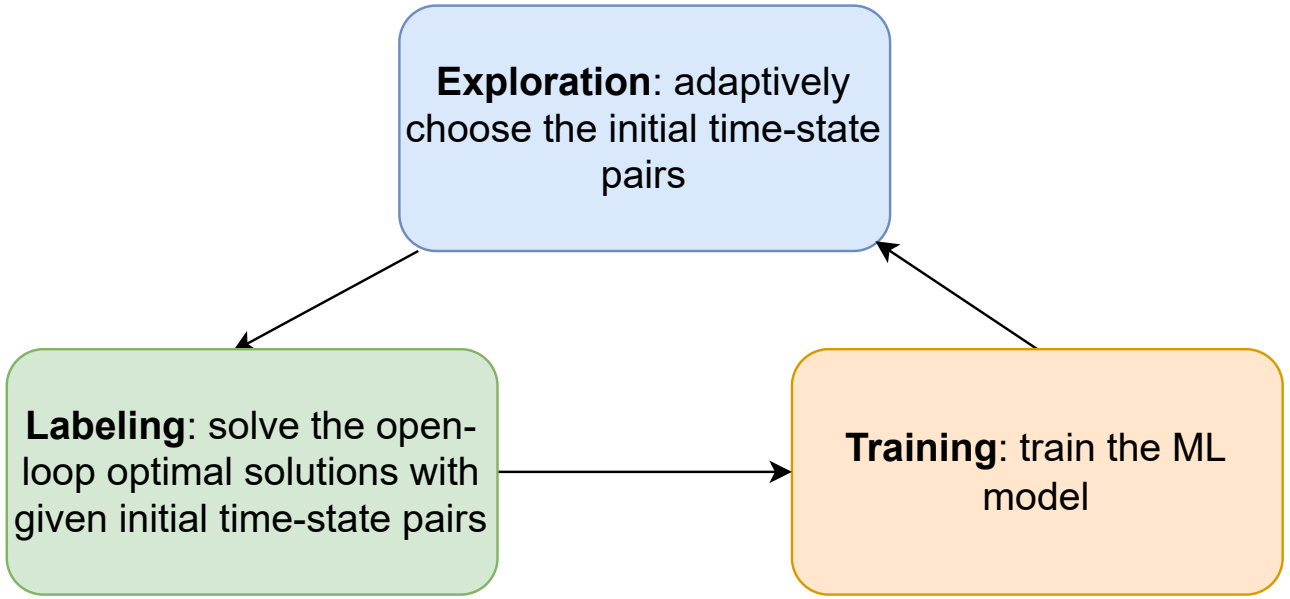


Supervised-Learning-Based Approach



Unlike a lot of classical ML tasks, we can choose data to label by our own choice.

Supervised-Learning-Based Approach



Unlike a lot of classical ML tasks, we can choose data to label by our own choice.

This work: focus on the exploration part to sample data that are more valuable to training

Adaptive Sampling Methods

- It is observed that the NN controller trained by the SL-based approach does not perform well even when both the training error and testing error are fairly small.

- Adaptive sampling methods aim to find the difficult points the NN controller suffers and add these points into the training data to improve the performance.

- Most existing approaches focus on choosing the initial points such as choosing the initial points with large gradients or bad performance.

Adaptive Sampling Methods

- It is observed that the NN controller trained by the SL-based approach does not perform well even when both the training error and testing error are fairly small.
- Adaptive sampling methods aim to find the difficult points the NN controller suffers and add these points into the training data to improve the performance.
- Most existing approaches focus on choosing the initial points such as choosing the initial points with large gradients or bad performance.
- However, these methods can not mitigate the distribution mismatch phenomenon brought by controlled dynamics.

Distribution Mismatch Phenomenon

Define $\mu_{\boldsymbol{u}}(t)$ the distribution of $\boldsymbol{x}(t)$:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t, \boldsymbol{x}(t))), \quad \boldsymbol{x}(0) \sim \mu_0,$$

- $\mu_{\boldsymbol{u}^*}(t)$: the distribution of the state at time t in the training data.
- $\mu_{\boldsymbol{u}^{\mathsf{NN}}}(t)$: the distribution of the input state of $\boldsymbol{u}^{\mathsf{NN}}$ at time t when applying the learned NN controller in the dynamics.

Distribution Mismatch Phenomenon

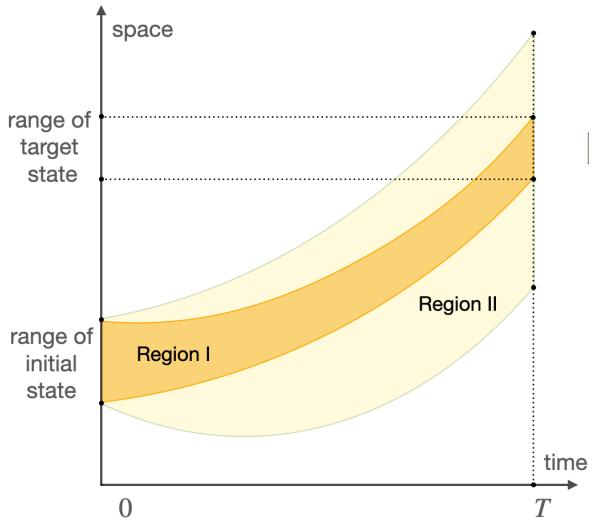
Define $\mu_{\boldsymbol{u}}(t)$ the distribution of $\boldsymbol{x}(t)$:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t, \boldsymbol{x}(t))), \quad \boldsymbol{x}(0) \sim \mu_0,$$

- $\mu_{\boldsymbol{u}^*}(t)$: the distribution of the state at time t in the training data.
- $\mu_{\boldsymbol{u}^{\mathbf{NN}}}(t)$: the distribution of the input state of $\boldsymbol{u}^{\mathbf{NN}}$ at time t when applying the learned NN controller in the dynamics.

Due to the learning error, $\mu_{\boldsymbol{u}^*}(t) \neq \mu_{\boldsymbol{u}^{\mathbf{NN}}}(t)$, and its discrepancy increases over time due to compounding error.

When t is large, the training data fails to represent the states encountered when keeping applying $\boldsymbol{u}^{\mathbf{NN}}$, and the error between $\boldsymbol{u}^*$ and $\boldsymbol{u}^{\mathbf{NN}}$ dramatically increases.



Region I: optimal paths

Region II: Paths controlled by the NN controller trained on data in Region I

Distribution Mismatch Phenomenon

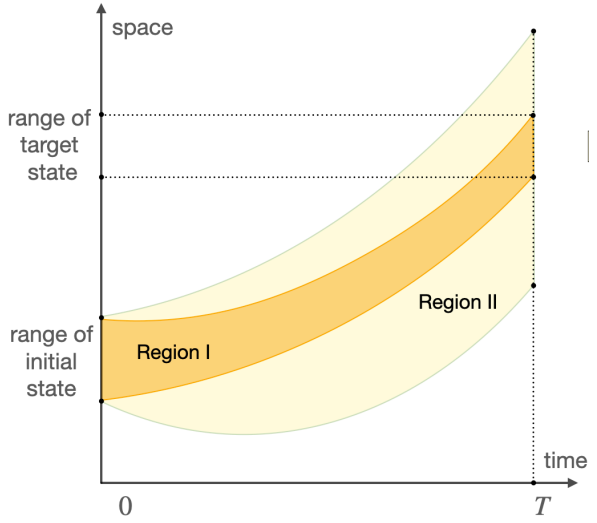
Define $\mu_{\boldsymbol{u}}(t)$ the distribution of $\boldsymbol{x}(t)$:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t, \boldsymbol{x}(t))), \quad \boldsymbol{x}(0) \sim \mu_0,$$

- $\mu_{\boldsymbol{u}^*}(t)$: the distribution of the state at time t in the training data.
- $\mu_{\boldsymbol{u}^{\mathbf{NN}}}(t)$: the distribution of the input state of $\boldsymbol{u}^{\mathbf{NN}}$ at time t when applying the learned NN controller in the dynamics.

Due to the learning error, $\mu_{\boldsymbol{u}^*}(t) \neq \mu_{\boldsymbol{u}^{\mathbf{NN}}}(t)$, and its discrepancy increases over time due to compounding error.

When t is large, the training data fails to represent the states encountered when keeping applying $\boldsymbol{u}^{\mathbf{NN}}$, and the error between $\boldsymbol{u}^*$ and $\boldsymbol{u}^{\mathbf{NN}}$ dramatically increases.



Region I: optimal paths

Region II: Paths controlled by the NN controller trained on data in Region I

distribution mismatch phenomenon is common when involving machine learning and dynamical systems, such as reinforcement learning and imitation learning.

IVP Enhanced Sampling Method

We propose initial value problem (IVP) enhanced sampling method to mitigate the distribution mismatch phenomenon.

Key idea: improve the quality of the NN controller iteratively by enlarging the training dataset with the states seen by the NN controller at previous times.

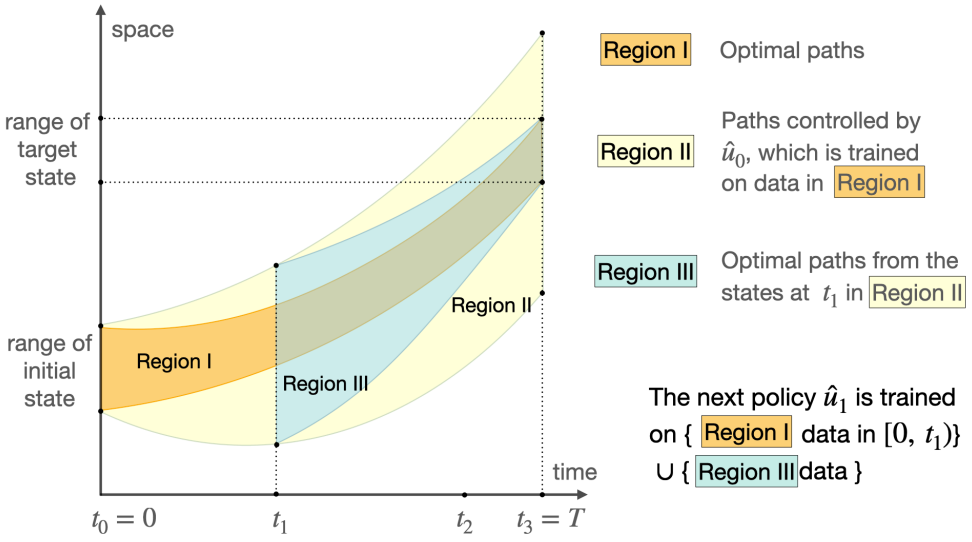


Figure: after first training

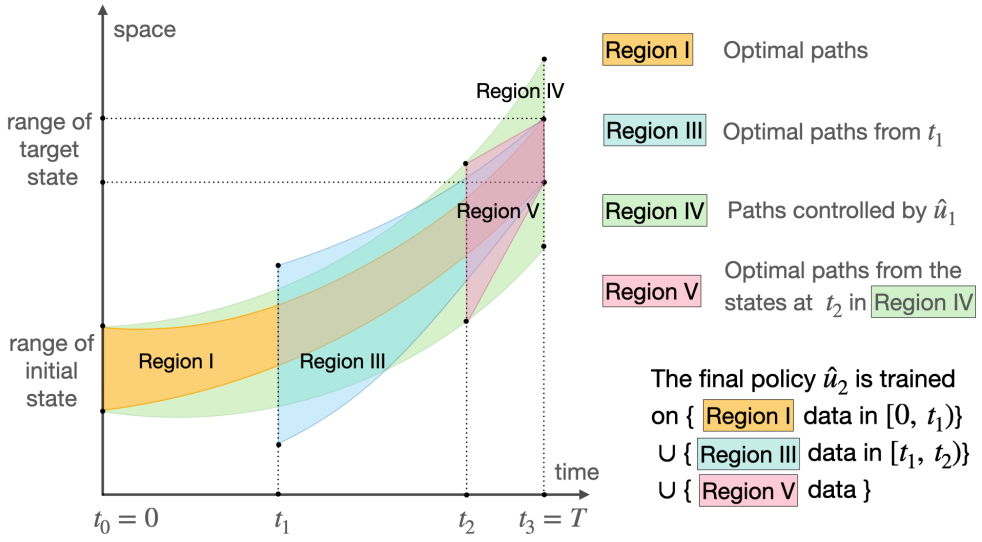
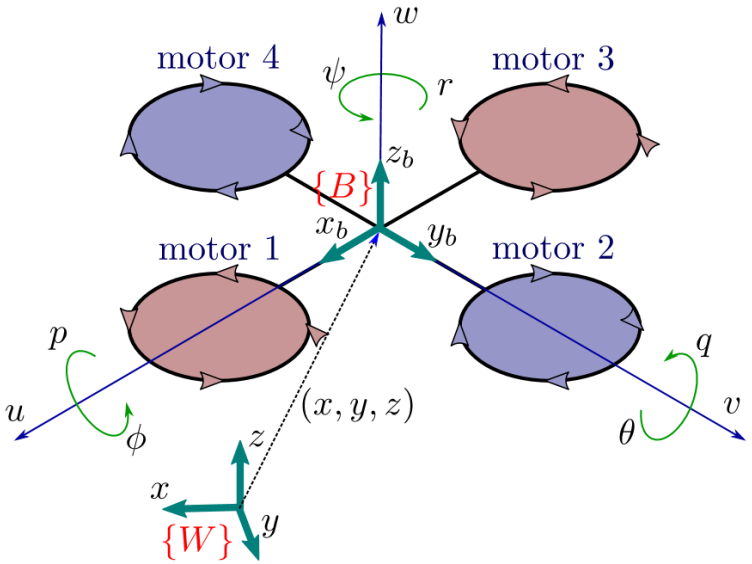


Figure: after second training

The Optimal Landing Problem of Quadrotor



Full dynamic model of a quadrotor with 12-dimensional state and 4-dimensional control.

The goal is to find the optimal landing paths with the minimum control effort from 0 to $T = 16$.

The IVP enhanced sampling method is implemented on the time grid $[0, 10, 14, 16]$.

Results on the Optimal Landing Problem

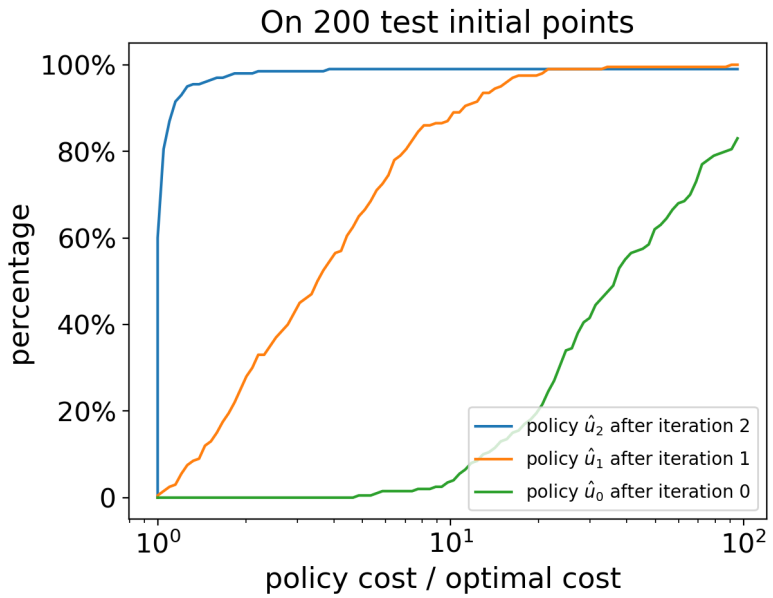


Figure: Cumulative distribution on 200 test initial points

Results on the Optimal Landing Problem

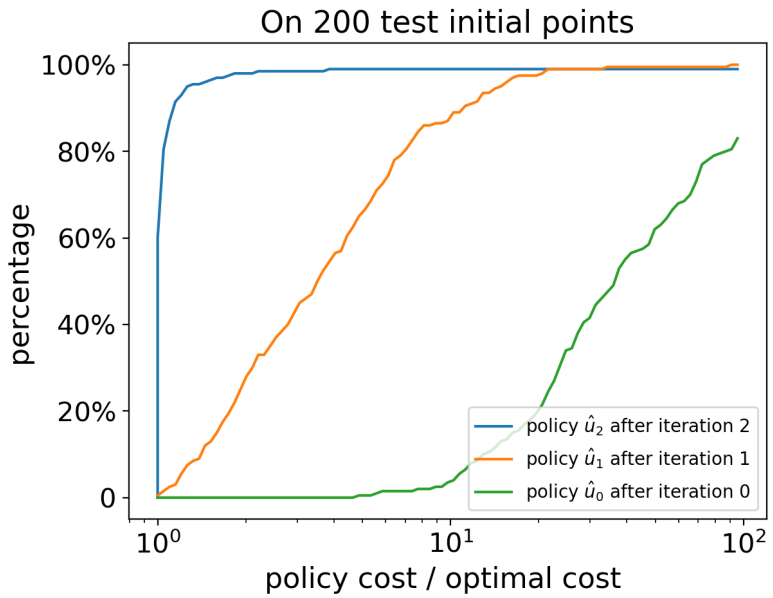
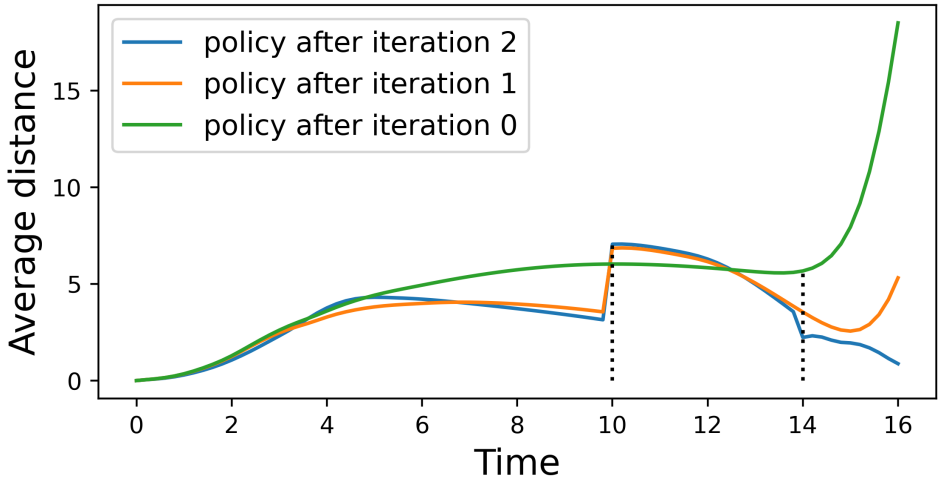


Figure: Cumulative distribution on 200 test initial points



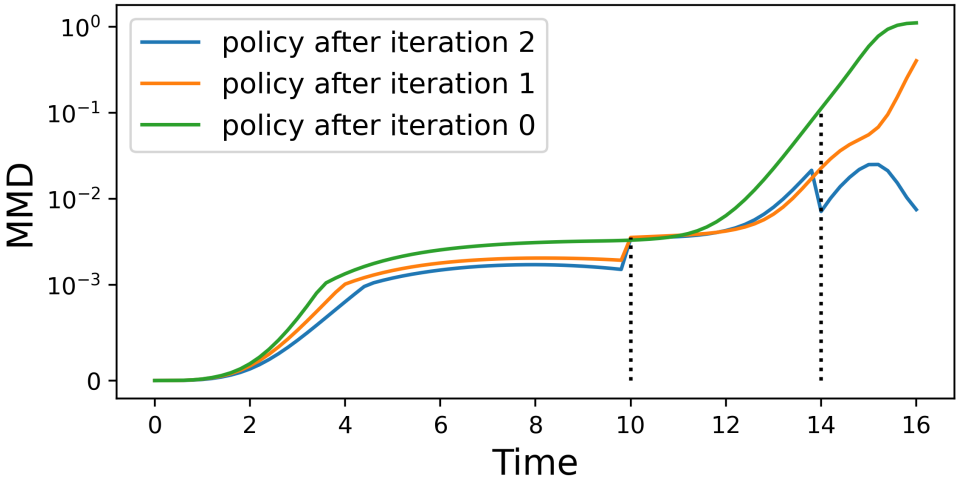
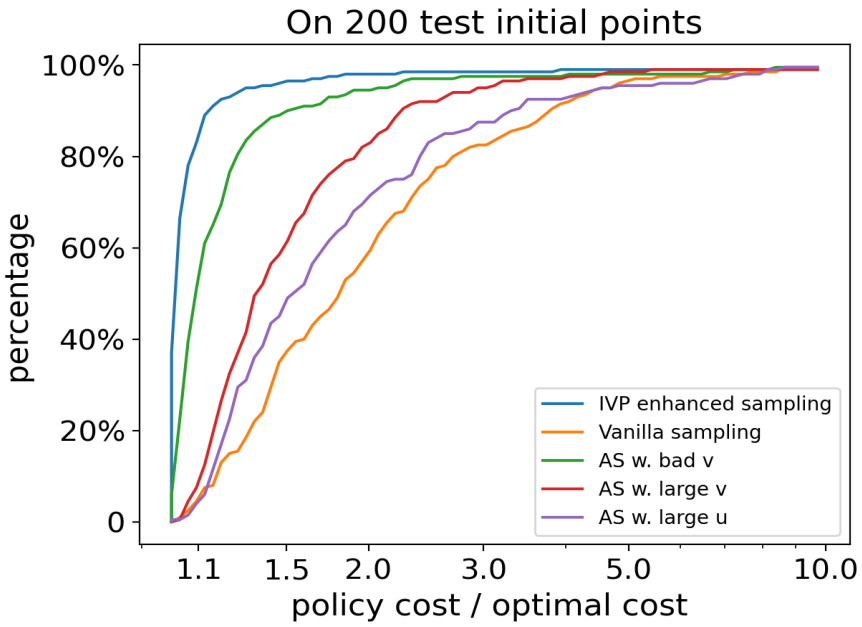


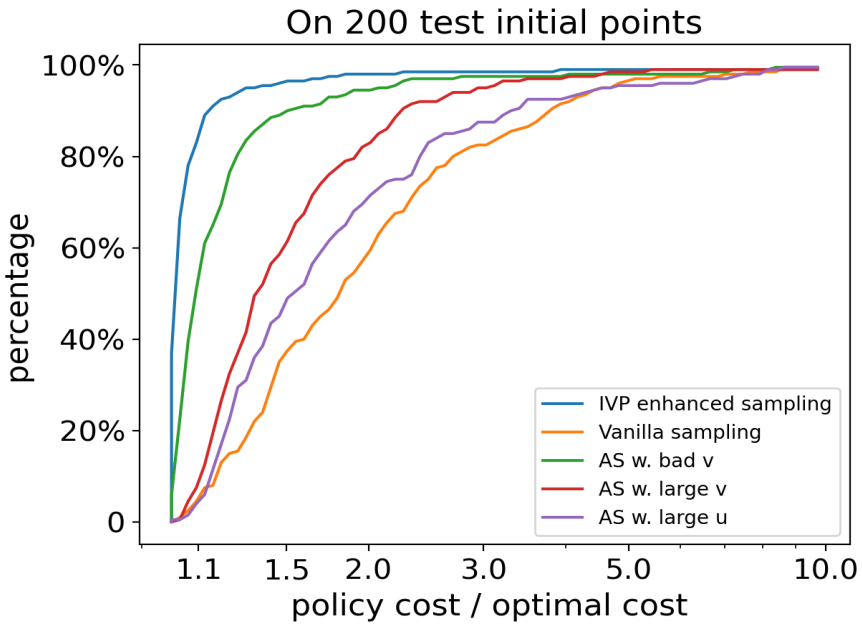
Figure: Discrepancy between the training data and the data reached by controllers at every time

Comparison with Other Methods



- Vanilla sampling: no adaptive
- AS w. bad v: choose initial points on which the NN controller performs bad
- AS w. large v: choose initial points on which the IVPs induced by the NN controller have large cost
- AS w. large u: choose initial points whose NN-predicted controls are large

Comparison with Other Methods



- Vanilla sampling: no adaptive
- AS w. bad v: choose initial points on which the NN controller performs bad
- AS w. large v: choose initial points on which the IVPs induced by the NN controller have large cost
- AS w. large u: choose initial points whose NN-predicted controls are large

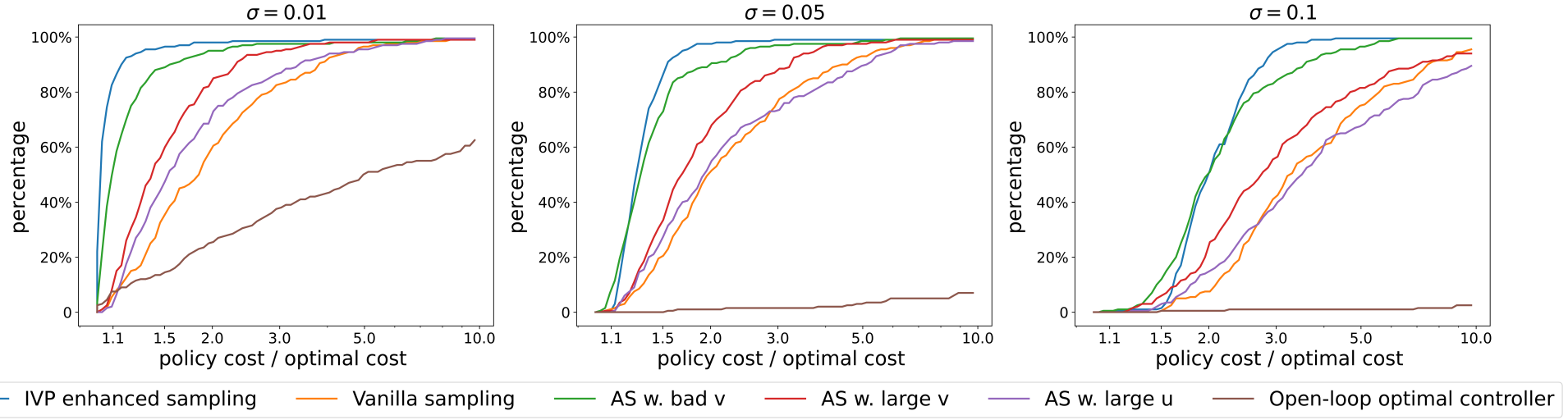
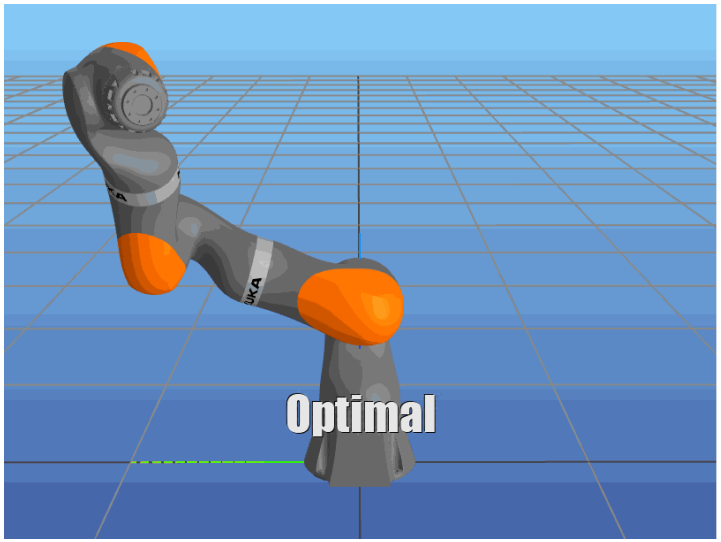
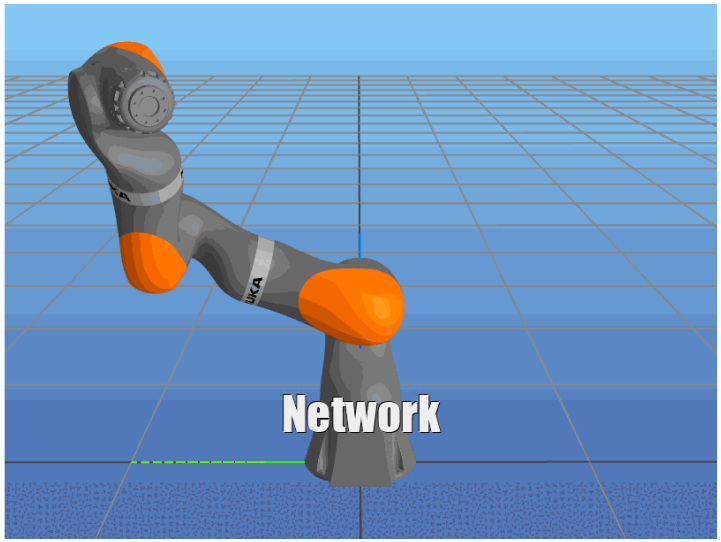


Figure: Cumulative distribution function of cost ratio between NN controlled value and optimal value under disturbance.

Optimal Reaching Problem of a 7-DoF Manipulator

The reaching problem on a 7-DoF torque-controlled manipulator, the KUKA LWR iiwa R820 14 with 14-dimensional state and 7-dimensional control.





Theoretical Analysis on an LQR Example

Consider the 1-d linear quadratic regulator (LQR) problem:

$$\min_{x(t),u(t)} \frac{1}{T}\int_{t_0}^{T} |u(t)|^2 \,\mathrm{d}t + |x(T)|^2$$

$$\text{s.t.} \quad \begin{cases} \dot{x}(t) = u(t),\ t \in [t_0, T] \\ x(t_0) = x_0 \end{cases}$$

The optimal controls are

$$\begin{cases} u^*(t; t_0, x_0) = -\frac{T}{T(T-t_0)+1} x_0, & \text{(open-loop optimal control)} \\ u^*(t, x) = -\frac{T}{T(T-t)+1} x. & \text{(closed-loop optimal control)} \end{cases}$$

Theoretical Analysis on an LQR Example

Consider the 1-d linear quadratic regulator (LQR) problem:

$$\min_{x(t),u(t)} \frac{1}{T}\int_{t_0}^{T} |u(t)|^2 \,\mathrm{d}t + |x(T)|^2$$

$$\text{s.t.} \quad \begin{cases} \dot{x}(t) = u(t),\ t \in [t_0, T] \\ x(t_0) = x_0 \end{cases}$$

The optimal controls are

$$\begin{cases} u^*(t; t_0, x_0) = -\frac{T}{T(T-t_0)+1} x_0, & \text{(open-loop optimal control)} \\ u^*(t, x) = -\frac{T}{T(T-t)+1} x. & \text{(closed-loop optimal control)} \end{cases}$$

Model 1: $u_\theta(t, x) = -\frac{T}{T(T-t)+1} x + b(t)$, where $\theta = \{\theta_t\}_{0\le t\le T} = \{b(t)\}_{0\le t\le T}$.

Theoretical Analysis on an LQR Example

Consider the 1-d linear quadratic regulator (LQR) problem:

$$\min_{x(t),u(t)} \frac{1}{T}\int_{t_0}^{T} |u(t)|^2\,\mathrm{d}t + |x(T)|^2$$
$$\text{s.t.} \begin{cases} \dot{x}(t) = u(t),\ t \in [t_0, T] \\ x(t_0) = x_0 \end{cases}$$

The optimal controls are

$$\begin{cases} u^*(t; t_0, x_0) = -\frac{T}{T(T-t_0)+1}x_0, & \text{(open-loop optimal control)} \\ u^*(t, x) = -\frac{T}{T(T-t)+1}x. & \text{(closed-loop optimal control)} \end{cases}$$

Model 1: $u_\theta(t, x) = -\frac{T}{T(T-t)+1}x + b(t)$, where $\theta = \{\theta_t\}_{0\le t\le T} = \{b(t)\}_{0\le t\le T}$.

Assume the open-loop optimal control solver gives the data with noise Z

$$\begin{cases} \hat{u}(t; t_0, x_0) = -\frac{T}{T(T-t_0)+1}x_0 + \epsilon Z, \\ \hat{x}(t; t_0, x_0) = x_0 + \int_{t_0}^{t} \hat{u}(t; t_0, x_0)\,\mathrm{d}t = \frac{T(T-t)+1}{T(T-t_0)+1}x_0 + (t - t_0)\epsilon Z. \end{cases}$$

Theoretical Superiority of the IVP Enhanced Sampling

Theorem

With Model 1, we do vanilla sampling with NT samples and IVP enhanced sampling with N initial points on temporal grids $0 < 1 < 2 < \cdots < T$. Let u_o, u_v and u_a be the optimal controller, the controller learned by the vanilla method, and the controller learned by the IVP enhanced sampling method, respectively. Define

$$\dot{x}_s(t) = u_s(t) = u_s(t, x_s(t)), \quad x_s(0) = x_{init},\ 0 \le t \le T, \quad s \in \{o, v, a\}.$$

1. *If x_{init} is a standard normal random variable. Let $\{\hat{x}_v^j(t)\}_{j=1}^{NT}$ and $\{\hat{x}_a^j(t)\}_{j=1}^{N}$ be the state variables in the training data of the vanilla method and the last iteration of the IVP enhanced sampling method. Then, $\hat{x}_v^j(t)$, $\hat{x}_a^j(t)$, $x_v(t)$ and $x_a(t)$ are mean-zero normal random variables and*

$$|\mathbb{E}|\hat{x}_v^j(t)|^2 - \mathbb{E}|x_v(t)|^2| = (1 - \frac{1}{NT})\epsilon^2 t^2, \quad |\mathbb{E}|\hat{x}_a^j(t)|^2 - \mathbb{E}|x_a(t)|^2| \le \epsilon^2.$$

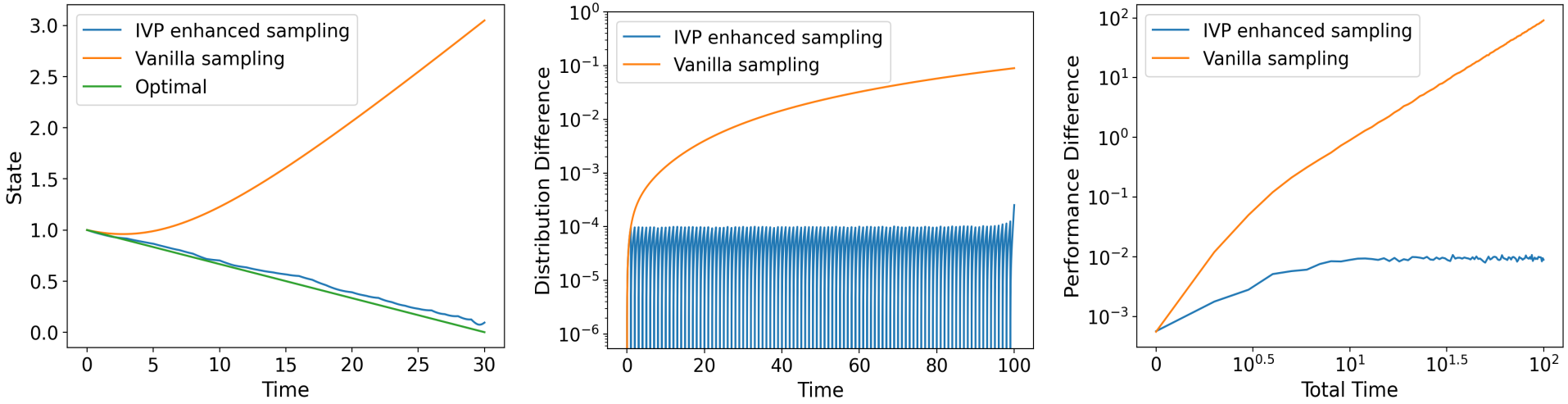
2. *If x_{init} is a fixed initial point, define the total cost*

$$J_s = \frac{1}{T}\int_0^T |u_s(t)|^2 \,\mathrm{d}t + |x_s(T)|^2, \quad s \in \{o, v, a\}.$$

Then, $\mathbb{E}J_v - J_o = \dfrac{T^2+1}{NT}\epsilon^2$, $\quad \mathbb{E}J_a - J_o \le \dfrac{3\epsilon^2}{N}$.

Superiority of the IVP Enhanced Sampling

Model 2: $u_\theta(t,x) = a(t)x + b(t)$, where $\theta = \{\theta_t\}_{0\le t\le T} = \{(a(t), b(t))\}_{0\le t\le T}$,



(a) **(b)** **(c)**

Figure: Numerical results on learning Model 2. (a) The optimal path and the paths generated by the vanilla sampling method and the IVP enhanced sampling method. (b) Differences of the second order moments (in the logarithm scale) between the distributions of the training data and the data reached by the controllers at different times. (c) Performance differences (in the logarithm scale) of the vanilla sampling method and the IVP enhanced sampling method for different total times (in the logarithm scale).

Conclusions

- Traditional methods for the closed-loop optimal control suffer from the curse of dimensionality while deep learning is promising in high dimensional closed-loop optimal control problems.

- Distribution mismatch phenomenon is an essential challenge in the supervised-learning-based approach for optimal control

- IVP enhanced sampling method can mitigate the distribution mismatch phenomenon and significantly improve the performance of the NN controller.

Thank you for your attention!