Proceedings of Machine Learning Research vol 242:654–666, 2024 6th Annual Conference on Learning for Dynamics and Control

On the Nonsmooth Geometry and Neural Approximation of the Optimal Value Function of Infinite-Horizon Pendulum Swing-up

Haoyu Han HHY20030118@MAIL.USTC.EDU.CN
School of the Gifted Young, University of Science and Technology of China

Heng Yang HANKYANG@SEAS.HARVARD.EDU
School of Engineering and Applied Sciences, Harvard University

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

Abstract

We revisit the inverted pendulum problem with the goal of understanding and computing the true optimal value function. We start with an observation that the true optimal value function must be nonsmooth (*i.e.,* not globally C^1) due to the symmetry of the problem. We then give a result that can certify the optimality of a candidate *piece-wise* C^1 value function. Further, for a candidate value function obtained via numerical approximation, we provide a bound of suboptimality based on its Hamilton-Jacobi-Bellman (HJB) equation residuals. Inspired by Holzhüter (2004), we then design an algorithm that solves backward the Pontryagin's minimum principle (PMP) ODE from terminal conditions provided by the locally optimal LQR value function. This numerical procedure leads to a piece-wise C^1 value function whose nonsmooth region contains periodic *spiral lines* and smooth regions attain HJB residuals about 10^{-4}, hence certified to be the optimal value function up to minor numerical inaccuracies. This optimal value function checks the power of optimality: (i) it sits above a polynomial lower bound; (ii) its induced controller globally swings up and stabilizes the pendulum, and (iii) attains lower trajectory cost than baseline methods such as energy shaping, model predictive control (MPC), and proximal policy optimization (with MPC attaining almost the same cost). We conclude by distilling the optimal value function into a simple neural network. Our code is avilable in https://github.com/ComputationalRobotics/InvertedPendulumOptimalValue.

Keywords: Optimal Control, Inverted Pendulum, Pontryagin's Minimum Principle

1. Introduction

Inverted pendulum is arguably one of the most fundamental problems in nonlinear (optimal) control. It has been frequently used in textbooks (Sontag, 2013; Slotine et al., 1991; Tedrake, 2009; Khalil, 2002) to illustrate foundational concepts such as feedback linearization, Lyapunov stability, proportional-integral-derivative (PID) control, energy shaping, to name a few. More recently, inverted pendulum is also one of the most basic benchmark problems for reinforcement learning, *e.g.,* in the Deepmind control suite (Tassa et al., 2018). Not only is the inverted pendulum a theoretically interesting problem to study, it also relates to practical applications in model-based humanoid control (Feng et al., 2014; Sugihara et al., 2002).

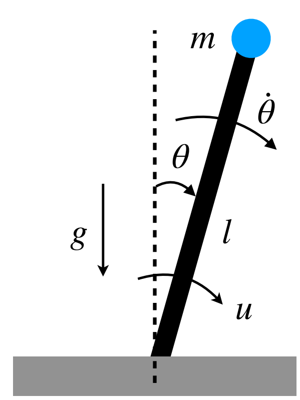


Figure 1: Pendulum.

One can often consider the inverted pendulum as a solved nonlinear control problem because in the model-based paradigm there exists elegant solutions such as energy pumping plus local linear-quadratic-regulator (LQR) stabilization (Åström and Furuta, 2000; Muskinja and Tovornik, 2006); and in the model-free paradigm algorithms such as proximal policy optimization

(PPO) and actor critic work very well (Raffin et al., 2021; Ren et al., 2023). However, from the perspective of *optimal control*, we know very little about the true optimal value function (or cost-to-go) and its associated optimal controller. This leads to the side effect that we cannot evaluate the suboptimality of other (approximately optimal) controllers. Let us state the continuous-time infinite-horizon (undiscounted) pendulum swing-up problem to understand why it is challenging to compute the optimal controller and value function.

Problem Setup. We are given the continuous-time pendulum dynamics as shown in Fig. 1

$$x := \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}, \quad \dot{x}(t) = f(x(t), u(t)) := \begin{bmatrix} \dot{\theta} \\ -\frac{1}{ml^2}\left(b\dot{\theta} - mgl\sin\theta - u\right) \end{bmatrix}, \tag{1}$$

where θ is the angular position, $\dot{\theta}$ is the angular velocity, m is the point mass, l is the length of the pole, b is the damping coefficient, g is the gravity constant, and u is the torque. Our goal is to swing up and stabilize the pendulum from any initial state x_0 to the upright position $x_{\mathrm{U}} = [2k\pi, 0]^{\mathsf{T}}, \forall k \in \mathbb{Z}$, an unstable equilibrium point. We formulate the undiscounted optimal control problem

$$J^*(x_0) = \min_{u(t)} \int_0^{+\infty} c(x(t), u(t)) dt, \quad \text{subject to} \quad x(0) = x_0, \;\; u(t) \in \mathbb{U}, \;\; \text{and (1)} \tag{2}$$

where the cost function $c(x, u)$ is defined as

$$c(x, u) = q_1 \sin^2\theta + q_1(\cos\theta - 1)^2 + q_2\dot{\theta}^2 + ru^2, \tag{3}$$

with $q_1, q_2, r > 0$. We let $\mathbb{U}$ in (2) be either $\mathbb{R}$ (without control saturation) or $[-u_{\max}, u_{\max}]$ with $u_{\max} < mgl$ (with control saturation). Note that we use "$\sin\theta$" and "$\cos\theta$", instead of θ, in the cost function (3) to avoid the modulo 2π issue. It is not difficult to observe that $J^*(x_{\mathrm{U}}) = 0$ because it is already stable. Problem (2) is a nonlinear quadratic regulator problem (Wernli and Cook, 1975). We make an assumption about the set of admissible control trajectories.

Assumption 1 (Admissible Control) *In problem (2), the control sequence $u(t)$ is admissible if (i) $u(t)$ is piece-wise continuous, and (ii) $x(t) \to x_{\mathrm{U}}$ under $u(t)$ when $t \to +\infty$.*

Intuitively, condition (ii) in Assumption 1 allows us to only consider the set of controllers that asymptotically stabilize the pendulum at x_{U}. When b is not too large, energy shaping followed by local LQR is such an admissible controller (hence the admissible control set is nonempty).

1.1. Related Work

Dynamic Programming. A straightforward approach for solving (2) is to discretize the dynamics (1) and perform value iteration with barycentric interpolation (Munos and Moore, 1998). Not only will this approach suffer from the curse of dimensionality, it is also unclear whether it will converge in the undiscounted case, as shown in (Yang, 2023, Example 2.3).

Hamilton-Jacobi-Bellman (HJB) Equation. The HJB theorem (Tedrake, 2009, Theorem 7.1) (Kamalapurkar et al., 2018) states that if one can find a C^1 function $J(x)$ such that $J(x_{\mathrm{U}}) = 0$, $J(x)$ is positive definite and satisfies the HJB equation

$$\min_{u \in \mathbb{U}} c(x, u) + \frac{\partial J}{\partial x}^{\mathsf{T}} f(x, u) = 0, \quad \forall x \tag{4}$$

then $J(x)$ is the optimal value function. Obtaining an analytic solution to (4) is often impossible, hence numerical approximations are needed. The levelset algorithm (Mitchell and Templeton, 2005; Osher and Sethian, 1988; Osher and Fedkiw, 2001) is a popular method to solve Hamilton-Jacobi (HJ)-type equations, in particular those appearing in reachability problems (Bansal et al., 2017). Nevertheless, to the best of our knowledge, it is not yet applicable to the pendulum problem because (4) cannot be transformed into an HJB equation that has a time derivative and terminal condition. A fundamental problem of the HJB equation (4) is that it implicitly assumes the optimal value function is C^1, which is not true for the pendulum problem, as we will show in Theorem 1. One can consider the notion of a *viscosity solution* (Bardi et al., 1997) to avoid this issue, but it does not make the computation any easier. A family of finite-element methods (Jensen and Smears, 2013; Smears and Suli, 2014; Kawecki and Smears, 2022) considers the stochastic optimal control problem where (4) becomes an elliptic PDE. However, they do not consider the infinite-horizon case where a boundary condition is unavailable.

Pontryagin's Minimum Principle (PMP). Another classical result in optimal control is PMP (to be reviewed in Lemma 4) (Bertsekas, 2012), which states the optimal state-control trajectory must satisfy an ODE (but trajectories satisfying the ODE may not be optimal). (Holzhüter, 2004; Hauser and Osinga, 2001) uses the local LQR value function of the pendulum to provide boundary conditions for PMP and computes a value function that swings up the pendulum. However, they only considered the case of no control saturation and did not prove optimality of the value function.

Weak Solution. Due to the difficulty of computing and certifying the optimal value function, Lasserre et al. (2007, 2005) developed a general framework of using convex relaxations to compute smooth *weak solutions* of the HJB (4) (Vinter, 1993). Yang et al. (2023) recently applied this method to compute polynomial lower bounds of the optimal value function. However, because the true optimal value function is nonsmooth, polynomial approximation is not expected to capture the detailed geometry of the optimal value function, as we will show in Fig. 3.

Neural Approximation. In addition to the aforementioned classical methods, using neural networks to approximate the optimal value function becomes increasingly popular (Lutter et al., 2020; Shilova et al., 2023). (Doya, 2000; Munos et al., 1999) first introduced HJB residual, *i.e.,* violation of (4), as a loss to train neural networks (Raissi et al., 2019), followed by Tassa and Erez (2007) showing how to avoid local minima, and Liu et al. (2014) showing how to make it robust to dynamic disturbance. However, the problem remains that only using HJB loss may lead to multiple solutions. Another line of work uses PMP to generate data for training (Nakamura-Zimmerer et al., 2021), but it requires solving a boundary value problem which may also have multiple solutions. In general, neural approximation also faces the same difficulty that the optimal value function may be nonsmooth, and it remains difficult to evaluate its suboptimality.

1.2. Contributions

We start with an observation (Theorem 1) that the optimal value function $J^*(x)$ of (2) must be nonsmooth at the bottomright position due to symmetry of the problem, and hence the HJB equation (4) cannot be satisfied everywhere in the state space. In such cases, little is known about $J^*(x)$ except that it is the so-called *viscosity solution* of the HJB (Bardi et al., 1997), which is difficult to interpret for practitioners. We contribute a result that is easy to interpret (Theorem 2), using elementary proof, that can certify the optimality of a given candidate *piece-wise* C^1 function. For numerically computed approximately optimal value functions, we give a result (Theorem 3) that certifies the *suboptimality* of the numerical solution w.r.t. the true optimal value function.

We then develop a numerical approach that, for the first time, computes the true optimal value function of pendulum swing-up, up to minor numerical inaccuracies. Our algorithm is inspired by the algorithm of Holzhüter (2004) and is based on PMP with boundary conditions provided by local LQR, but it makes several improvements. For example, we handle the case with control saturation, we uncover a nonsmooth curve in the optimal value function, and we can bound the suboptimality of our solution using Theorem 3. We then showcase the power of optimality. (a) The controller induced from the optimal value function swings up and stabilizes the pendulum from any initial state. (b) The induced controller achieves *lower* cost than existing controllers such as energy pumping, reinforcement learning, and model predictive control (MPC), with the MPC controller being the best baseline as it achieves almost the same cost as our controller. (c) The optimal value function indeed sits above the polynomial lower bound obtained from convex relaxations.

Our numerical algorithm is expensive as it requires solving a large amount of PMP trajectories, computing intersections, and storing dense samples of the optimal value function. We therefore ask if we can use a neural network to *distill* and *compress* the optimal value function. In the supervised case, we show that we just need 50 optimal value samples to train a simple neural network whose induced controller can globally swing up the pendulum. In the weakly supervised case, we design a novel loss function to train a neural network directly from *raw PMP trajectories*, and the resulting controller still globally swings up the pendulum. This simple training scheme generalizes to the more challenging cart-pole problem, where we also obtain a global stabilizing controller.

Limitations. Unfortunately, there are still puzzles related to the true optimal value function (in our opinion, due to the limitations of fundamental theoretical tools in optimal control). In the case with control saturation, we observe and conjecture that the optimal value function is *discontinuous*. Although we cannot formally prove our conjecture, we provide numerical evidence based on the limiting discounted viscosity solution idea in Bardi et al. (1997).

Proofs and extra results are available at `https://hyhan0118.github.io/l4dc.pdf`.

2. Certificate of (Sub-)Optimality for the Nonsmooth Value Function

We start with an observation that the optimal value function $J^*(x)$ of (2) must be nonsmooth.

Theorem 1 (Nonsmooth Optimal Value Function) *The optimal value function $J(x)$ to problem (2) is not C^1 at the bottomright position $x_\mathrm{B} := [\pi + 2k\pi, 0]^\mathsf{T}, \forall k \in \mathbb{Z}$.*

Here we provide a brief explanation. If $J(x)$ were smooth at x_B, then it must satisfy the HJB equation (4), implying the optimal controller at x_B must be unique due to strong convexity of the cost (3). However, our physics insight suggests that swinging the pendulum from the left is equivalent to swinging it from the right, resulting in two symmetric optimal controllers and a contradiction.

2.1. Certificate of Optimality

We then state a result that verifies the optimality of a candidate piece-wise C^1 value function for (2).

Theorem 2 (Optimality Certificate of A Piece-wise C^1 Value Function) *Let $\mathbf{O}_{-N}, \dots, \mathbf{O}_N$ be open subsets of $\mathbb{R}^2$ that satisfy*

(i) $\cup_{i=-N}^{N} \mathbf{O}_i = \mathbb{R}^2$,

(ii) $\forall i$, $\mathbf{O}_i \cap \mathbf{O}_{i+j} \neq \emptyset$ *if* $j = \pm 1$, *and* $\mathbf{O}_i \cap \mathbf{O}_{i+j} = \emptyset$ *if* $|j| > 1$,

and $J_{-N}, \dots, J_N(x)$ be C^1 functions defined on them, respectively (N possibly infinite). Define

$$J(x) = \min_i \{J_i(x) | x \in \mathbf{O}_i\}.$$

If $J(x)$, $\mathbf{O}_i$'s, and $J_i(x)$'s are such that

(iii) $J(x)$ *is continuous and piece-wise* C^1 *on* $\mathbb{R}^2$*,*

(iv) $J(x_{\mathrm{U}}) = 0$ *where* $x_{\mathrm{U}} = [2k\pi, 0]^{\mathsf{T}}, \forall k \in \mathbb{Z}$ *is the upright position,*

(v) $\forall i$*,* $J_i(x)$ *satisfies the HJB equation (4) everywhere on* $\mathbf{O}_i$*,*

(vi) the nonsmooth curve $\mathbf{\Gamma} := \{x \in \mathbb{R}^2 | \exists (i,j) \text{ s.t. } J_i(x) = J_j(x)\}$ *can be locally defined by* $\{x | G(x) = 0\}$ *with* G *a* C^1 *function, and every admissible trajectory* $x(t)$ *satisfies* $G(x(t))$ *is monotonic in* t *near an intersection point* $x(t_0)$ *where* $G(x(t_0)) = 0$*,*

(vii) $\forall x_0 \in \mathbb{R}^2$*, there exists a trajectory* $(x(t), u(t))$ *starting from* x_0 *that attains cost* $J(x_0)$*,*

then $J(x)$ *is the optimal value function of (2).*[1]

Theorem 2 provides a list of conditions to certify optimality of a piece-wise C^1 function $J(x)$. The only technical condition that is difficult to verify is (vi), which is necessary to avoid state trajectories that cross the nonsmooth region $\mathbf{\Gamma}$ in a pathological way, *e.g.,* imagine $\sin(\frac{1}{t})$ crossing the x-axis when t tends to 0. In the pendulum problem, each $\mathbf{O}_i$ is an open set containing x_{U} and differs from $\mathbf{O}_{i\pm1}$ by a shift of 2π along the θ-axis, with $J_i(x)$ defined on it ($J_i(x)$ is equal to $J_{i\pm1}(x)$ by shifting 2π). $\mathbf{\Gamma}$ composes of an infinite number of nonsmooth *spiral* lines, again shifted by 2π along the θ-axis, intersected by $J_i(x)$ and $J_{i\pm1}(x)$. The numerical algorithm we develop in Section 3, based on PMP, ensures each $J_i(x)$ satisfies HJB (4) on $\mathbf{O}_i$[2], $J_i(x)$ is C^1, and $J(x)$ is attainable. For more details please refer to Figure 2.

2.2. Certificate of Suboptimality

Finding analytical solutions that exactly satisfy Theorem 2 is intractable. For numerically computed candidate value functions, we wish to compute a suboptimality certificate w.r.t. $J^*(x)$ of (2). Toward this, we need to first review the local LQR controller of the inverted pendulum.

Local LQR. The pendulum dynamics (1) satisfies $f(x_{\mathrm{U}}, 0) = 0$ and we can linearize $f(x, u)$ around $(x_{\mathrm{U}}, 0)$ to obtain a linear system

$$\dot{x} = A(x - x_{\mathrm{U}}) + Bu, \quad A = \frac{\partial f}{\partial x}(x_{\mathrm{U}}, 0), B = \frac{\partial f}{\partial u}(x_{\mathrm{U}}, 0). \tag{5}$$

Similarly, we can perform a quadratic approximation of the cost function $c(x, u)$ around $(x_{\mathrm{U}}, 0)$

$$c(x, u) \approx q_1(\theta - 2k\pi)^2 + q_2\dot{\theta}^2 + ru^2 = (x - x_{\mathrm{U}})^{\mathsf{T}} Q (x - x_{\mathrm{U}}) + ru^2. \tag{6}$$

The optimal value function for minimizing (6) subject to (5) is a quadratic function

$$J_\infty(x) = (x - x_{\mathrm{U}})^{\mathsf{T}} P (x - x_{\mathrm{U}}), \tag{7}$$

where $P \succ 0$ is the unique positive definite solution to the algebraic Riccati equation

$$A^{\mathsf{T}} P + PA - \frac{1}{r} PBB^{\mathsf{T}} P + Q = 0.$$

We now introduce a suboptimality certificate for any candidate C^1 value function.

1. If $J(x)$ is discontinuous, we require admissible trajectories to not cross the discontinuous region to attain lower costs.
2. The satisfaction of HJB and C^1 is not entirely precise as it relies on numerical solutions, hence the development of Theorem 3 for error estimation.

Theorem 3 (Sub-Optimality Certificate of A C^1 Value Function) *Let $\mathcal{L} := \{x \in \mathbb{R}^2 \mid J_\infty(x) \leq \varepsilon\}$ be defined with a sufficiently small $\varepsilon > 0$ such that $\mathcal{L}$ is a region of attraction for x_{U} using the local LQR controller within the control bounds $\mathbb{U}$. Let $T_x > 0$ be the time taken by the optimal controller to enter region $\mathcal{L}$ from initial state $x \in \mathbb{R}^2$. If $J(x)$ is a C^1 function on $\mathbb{R}^2$ that satisfies*

(i) $|J(x) - J_\infty(x)| \leq \delta$ *for any* $x \in \mathcal{L}$*, and*

(ii) there exists a continuous function $l(x)$ *such that*

$$\min_{u\in\mathbb{U}} c(x,u) + \frac{\partial J}{\partial x}^{\mathsf{T}} f(x,u) = l(x), \quad \|l(x)\| < \epsilon, \quad \forall x \in \mathbb{R}^2, \tag{8}$$

(iii) $\forall x_0 \in \mathbb{R}^2$*, there exists a trajectory* $(x(t), u(t))$ *starting from* x_0 *that attains cost* $J(x_0)$*,*

then $J(x)$ *has bounded error from* $J^*(x)$ *as*

$$J^*(x) \leq J(x) \leq J^*(x) + \epsilon T_x + \delta + \varepsilon. \tag{9}$$

Theorem 3 is computationally useful as $J(x)$ is usually a C^1 function interpolated from samples. Condition (i) is easy to realize, in fact, one can choose $J(x) \equiv J_\infty(x)$ for $x \in \mathcal{L}$ so that $\delta = 0$ (as what we will do in Section 3, we will solve backward ODEs from $J_\infty(x)$ to get $J(x)$, so in $\mathcal{L}$ they are the same). Condition (ii) is also checkable as one can compute $l(x)$ from $J(x)$ (the minimization in (8) is closed-form solvable) and evaluate ϵ. T_x needs to be estimated. In practice, we approximate $T_x < 10$ as we can swing up the pendulum to region $\mathcal{L}$ within ten seconds.[3]

3. Numerical Approximation by Pontryagin's Minimum Principle

We design an algorithm based on PMP to compute a value function that verifies Theorem 2-3.

3.1. Numerical Procedure

We begin by recalling Pontryagin's minimum principle, which can be derived using the method of characteristics for the HJB (4).

Lemma 4 (Pontryagin's Minimum Principle) *Let $(u^*(t), x^*(t)), t \in [0, T]$ be a pair of optimal control and state trajectories satisfying dynamics (1) and $x^*(0) = x_0$ as given. Let $p(t)$ be the solution of the adjoint equation* almost everywhere

$$\dot{p}(t) = -\nabla_x H(x^*(t), u^*(t), p(t)), \quad p(T) = \nabla_x J(x^*(T)) \tag{10}$$

where J is the optimal value function and H is the Hamiltonian defined by

$$H(x, u, p) = c(x, u) + p^T f(x, u) \tag{11}$$

Then, for almost every $t \in [0, T]$ we have

$$u^*(t) = \arg\min_{u\in\mathbb{U}} H(x^*(t), u, p(t)) \tag{12}$$

To use Lemma 4, we will (i) solve the problem (12), and (ii) provide a terminal condition $p(T)$. **Solve** u^*. Observe that the pendulum dyamics (1) is control-affine

$$f(x,u) = f_1(x) + f_2(x)u, \quad f_1(x) = \begin{bmatrix} \dot{\theta} \\ -\frac{1}{ml^2}(b\dot{\theta} - mgl\sin\theta) \end{bmatrix}, f_2(x) = \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix},$$

3. Or we can approximate T_x by $J_{\epsilon_1} - J_{\epsilon_2} \approx (\epsilon_1 - \epsilon_2)T_x$, J_ϵ is the calculated value function with error ϵ

and the cost function $c(x, u)$ (3) is quadratic in u. Therefore, the solution to (12) is

$$u^* = \begin{cases} -\frac{1}{2r} p^\mathsf{T} f_2(x) & \text{if } \mathbb{U} = \mathbb{R} \\ \text{clip}\left(-\frac{1}{2r} p^\mathsf{T} f_2(x), -u_{\max}, u_{\max}\right) & \text{if } \mathbb{U} = [-u_{\max}, u_{\max}] \end{cases} \tag{13}$$

where the "clip" function saturates the control between $-u_{\max}$ and $u_{\max}$. Inserting (13) back to the adjoint equation (10) and the original dynamics (1), we obtain an ODE in the optimal state $x^*(t)$ and the co-state $p(t)$, which can be solved when boundary conditions are provided.

Terminal Condition. Inspired by Holzhüter (2004), we provide terminal conditions of the PDE, *i.e.,* a pair of $x^*(T_x)$ and $p(T_x)$ (because the associated $p(0)$ with $x^*(0)$ is unavailable). Because the LQR value function (7) is locally optimal around x_{U}, for any $x^*(T_x)$ that is on the boundary of the small ellipse (such that $\mathcal{L}$ is defined as in Theorem 3)

$$\partial\mathcal{L} = \{x \in \mathbb{R}^2 \mid J_\infty(x) = \varepsilon\}, \tag{14}$$

we can approximate

$$p(T_x) = 2P(x^*(T_x) - x_{\mathrm{U}}).$$

Once $x^*(T_x)$ and $p(T_x)$ are available, we can solve the ODE using *backward integration* to obtain a locally optimal trajectory that satisfies PMP.

Sample $x^*(T_x)$. We then wish to densely sample $x^*(T_x)$ on $\partial\mathcal{L}$ (14) to obtain a large amount of PMP trajectories to densely cover the state space $\mathbb{R}^2$. A naive uniform sampling strategy will lead to trajectories clustered in certain regions and do not fully cover $\mathbb{R}^2$. Inspired by Holzhüter (2004), we sample $x^*(T_x)$ based on a distance metric between two PMP trajectories. Let $x_1^*(t)$ and $x_2^*(t)$ be two PMP trajectories already computed, the distance between these two trajectories is defined as

$$d(x_1^*(t), x_2^*(t)) = \|x_1^c - x_2^c\|, \quad x_i^c = \{x_i(t_c) \mid J(x_i(t_c)) = V_c, t_c \in [0, +\infty]\}, i = 1, 2, \tag{15}$$

with V_c a positive number larger than ε (*e.g.,* $V_c = 10000\varepsilon$). The idea of this metric is to ensure the trajectories stay close after backward integration. The sampling algorithm is designed to make adjacent PMP trajectories have equal distances based on (15). Details are provided in full paper.

Algorithm 1: Compute the Nonsmooth Curve

Input: PMP trajectories $\mathcal{T}$; small value increment $\Delta > 0$; number of values M
Output: Set of intersection points $\mathcal{S}$
for $k \leftarrow 1$ *to* M **do**
  $\mathcal{C} \leftarrow \emptyset$
  for τ *in* $\mathcal{T}$ **do**
    $j_{\max} = \max\{j \mid \tau(j).\text{value} \leq k\Delta\}$
    $\mathcal{C} \leftarrow \mathcal{C} \cup \tau(j_{\max}).\text{state}$
  end
  $S = \text{shift\_intersect}(\mathcal{C})$
  $\mathcal{S} \leftarrow \mathcal{S} \cup S$
end

Intersection of PMP Trajectories & the Nonsmooth Curve. After we obtain a large set of PMP trajectories(in full paper), they will intersect with each other and themselves on a 2D plane.

We calculate the state where a trajectory intersect with others at the first time, in order to stop it there. All these terminal states form a spiral line, which you can find in the middle of Figure 2.

Given two PMP trajectories $x_1^*(t)$ and $x_2^*(t)$, if there exist t_1 and t_2 such that $x_{12}^* = x_1^*(t_1) = x_2^*(t_2)$ and $J_1(x_1^*(t_1)) = J_2(x_2^*(t_2))$ (here J_1 and J_2 are the same as in Theorem 1), then x_{12}^* is an intersection point, from which there exist (at least) two optimal[4] trajectories achieving the same cost. Therefore, by the same reasoning as in Theorem 1, x_{12}^* can be a point at which the optimal value function is nonsmooth. Algorithm 1 presents a method to compute all these intersection points. Given a set of PMP trajectories $\mathcal{T}$ where each trajectory $\tau \in \mathcal{T}$ contains a sequence of states and values (*i.e.,* $x^*(t)$ and $J(x^*(t))$ at discrete timesteps, $\tau(j).\mathrm{value}$ and $\tau(j).\mathrm{state}$ represent the value and state, respectively), line 4-8 computes all the states of the trajectories that have value $k\Delta$. Among these states $\mathcal{C}$, line 9 finds the common states by first forming a polygon using the points in $\mathcal{C}$ and then intersect $\mathcal{C}$ with a copy of $\mathcal{C}$ shifted along θ-axis by 2π[5]. The output $\mathcal{S}$ thus contains all such intersection points forming a spiral line.

Controller Synthesis. After getting the nonsmooth curves, we restrict all raw PMP trajectories to lie inside the nonsmooth curves. Then, we interpolate the value samples to obtain the value function. To synthesize controls, we use the solution in (13) with interpolated co-state p from samples.

3.2. Results

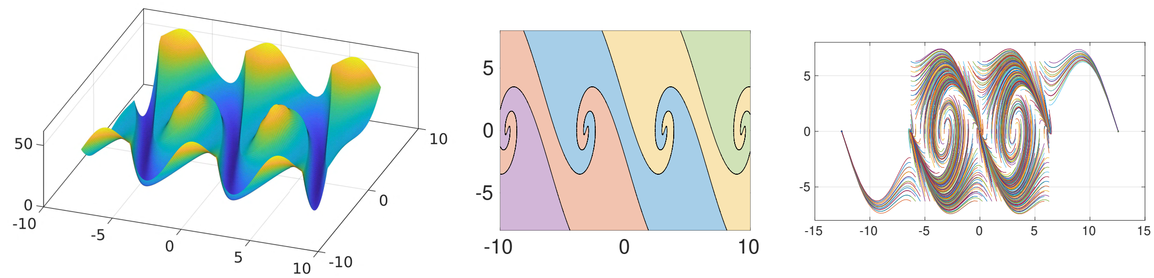
Setup. We use $m = 1$, $b = 0.1$, $l = 1$, $g = 9.8$, $q_1 = 1$, $q_2 = 1$, $r = 1$ in the dynamics (1) and cost function (3). We set $u_{\max} = 2$ in the case of control saturation. We are interested in the optimal value function on the region $x \in [-8, 8] \times [-8, 8]$, as it contains $[0, 0]$, $[2\pi, 0]$ and $[-2\pi, 0]$ (once we obtain J on this region we can shift it by $2k\pi$ to get other regions). We set $\varepsilon = 0.0002$ in (14).

Optimal Value Function. Fig. 2 shows the optimal value functions both (a) without control saturation and (b) with control saturation. The middle column of Fig. 2 draws the nonsmooth curves obtained using Algorithm 1, with the colored regions indicating the regions of attraction to the upright position x_{U} (*e.g.,* for any initial state in the blue region, the optimal trajectory will stay in the blue region and converges to x_{U}). In each of the colored regions, the HJB residuals, *i.e.,* $l(x)$ in Theorem 3, are about 10^{-4}. Therefore, according to Theorems 2 and 3, we can conclude the numerically computed value functions in Fig. 2 are the optimal value functions, up to minor numerical inaccuracies and suboptimality. To further verify the correctness of the optimal value functions, Fig. 3 compares the numerical value function with a smooth degree-7 polynomial lower bound computed using SOS relaxations in the case of control saturation (Yang et al., 2023). As we can see, the optimal value function sits above the polynomial lower bound, and the smooth polynomial hardly captures the nonsmooth geometry, especially around $[\pm 2\pi, 0]$.

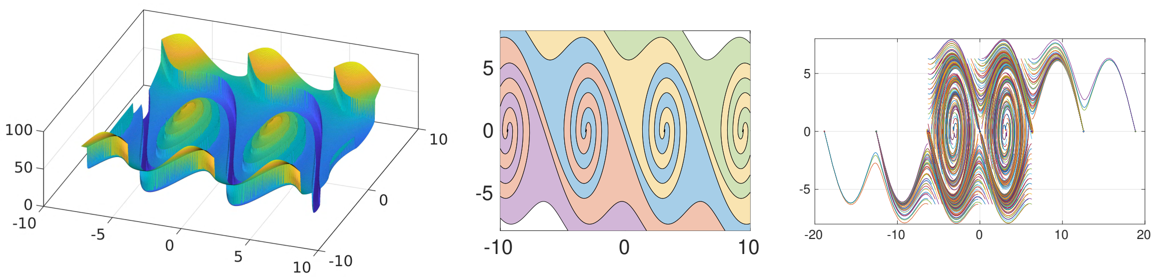
Remark 5 (Discontinuity) *The optimal value function in Fig. 2(b) appears to be* discontinuous. *This is a puzzle that we cannot formally (dis-)prove. Even after adding a discount factor in the cost* (3)*, the discontinuous phenomenon remains, see full paper. As a result, we cannot conclude the (dis-)continuity of the true optimal value function by using (Bardi et al., 1997, Theorem 1.5).*

Optimal Controller. The right column of Fig. 2 plots state trajectories using the optimal controller induced by the optimal value function via (13), starting from a dense grid of 30×30 initial

4. Here "optimal" solely means it satisfied PMP and is a candidate for optimal trajectory
5. One can treat $\mathcal{C}$ as contour line



(a) Without control saturation $\mathbb{U} = \mathbb{R}$



(b) With control saturation $\mathbb{U} = [-u_{\max}, u_{\max}], u_{\max} = 2$

Figure 2: Optimal value function and controller. Left: optimal value function shown in 3D plots. Middle: nonsmooth curves computed from Algorithm 1. Right: global stabilizing trajectories starting from 30×30 initial states. Better viewed when zoomed in.

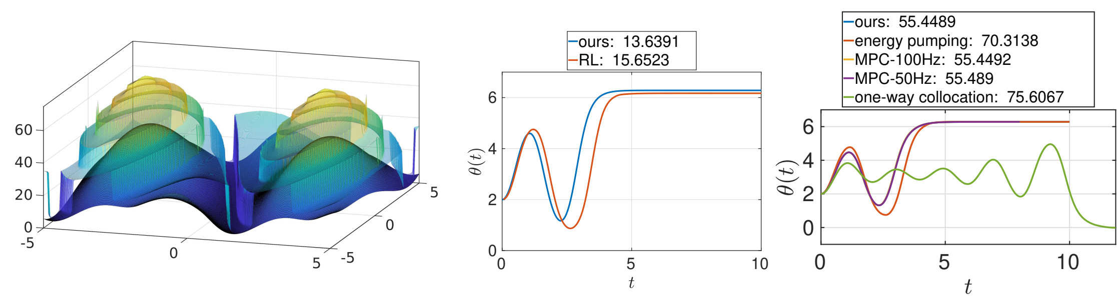


Figure 3: Comparison of the optimal value function and controller with baselines. Left: the optimal value function sits above a polynomial lower bound. Middle: optimal controller achieves lower cost than a controller trained from PPO. Right: optimal controller achieves lower cost than energy pumping, and almost the same cost as MPC.

states. Observe that the optimal controller swings up and stabilizes the pendulum in all cases. We then investigate if the optimal controller outperforms other algorithms. We implement four baselines: (i) energy pumping plus local LQR, (ii) open-loop trajectory optimization using direct collocation with 80 variable timesteps, (iii) model predictive control (MPC) with 5 seconds prediction horizon, at 50 Hz and 100 Hz using (Fiedler et al., 2023), and (iv) proximal policy optimization (PPO) (Schulman et al., 2017; Raffin et al., 2021). Comparison with the first three baselines using the same set of parameters as before are shown in Fig. 3 right column. Observe that the optimal controller achieves lower costs than energy pumping and trajectory optimization, and almost the same cost as MPC. PPO fails in the original set of parameters but succeeds with $q_1 = 1$, $q_2 = 0.1$, $r = 0.01$. We therefore rerun our numerical procedure to compare our controller with PPO, shown in Fig. 3 middle column. Similarly, the optimal controller outperforms PPO in terms of lower cost.

4. Neural Approximation

The power of optimality comes at a price: there are $78,797$ raw PMP trajectories and $371,028,742$ value samples in the optimal value function of Fig. 2(b). We investigate using a neural network $J_{\mathrm{NN}}(x)$ to distill knowledge from the PMP data. We use a neural network with 2 hidden layers each with 200 neurons. The input to J_{NN} is $(\sin\theta, \cos\theta, \dot{\theta})$. We consider the case with control saturation.

Supervised Training. We supervise $J_{\mathrm{NN}}(x)$ using data samples from $J^*(x)$ with the loss

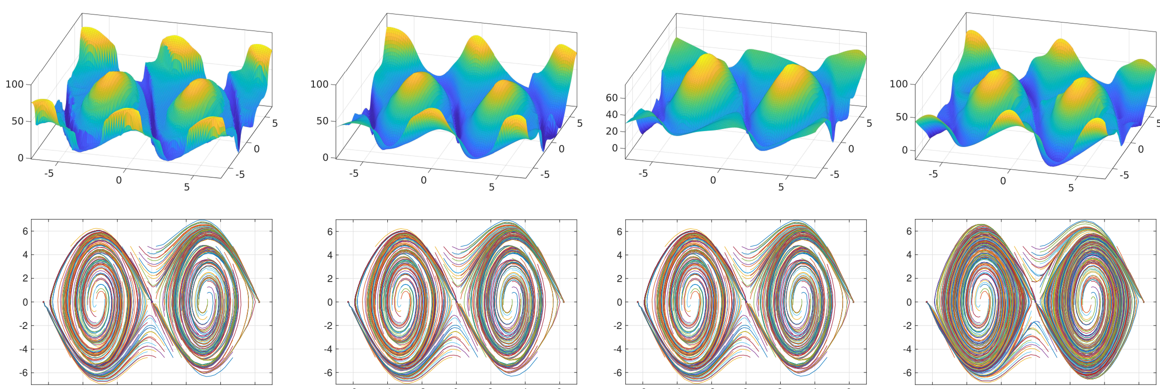
$$\ell_{\mathrm{S}} = \lambda_{\mathrm{LQR}}\ell_{\mathrm{LQR}} + \lambda_{\mathrm{V}}\ell_{\mathrm{V}} + \lambda_{\mathrm{HJB}}\ell_{\mathrm{HJB}} + \lambda_{\mathrm{smooth}}\ell_{\mathrm{smooth}}, \tag{16}$$

where ℓ_{LQR} uses the local LQR value function $J_\infty(x)$ to supervise $J_{\mathrm{NN}}(x)$ around x_{U}; ℓ_{V} uses random samples from $J^*(x)$ in Fig. 2 to supervise $J_{\mathrm{NN}}(x)$; ℓ_{HJB} penalizes violation of the HJB residual (4); and ℓ_{smooth} encourages $J_{\mathrm{NN}}(x)$ to be smooth (more details in full paper). Fig. 4(a) plots trained $J_{\mathrm{NN}}(x)$ and the induced controllers with decreasing samples used in ℓ_{V}. We see even with just 50 value samples, the controller globally swings up and stabilizes the pendulum.

Weakly Supervised Training. The loss ℓ_{V} requires $J^*(x)$ that is expensive to compute due to Algorithm 1. We replace ℓ_{V} with a loss that only requires *raw* PMP trajectories

$$\ell_{\mathrm{PMP}} = \frac{1}{N_{\mathrm{PMP}}} \sum_{i=1}^{N_{\mathrm{PMP}}} \mathrm{LeakyReLU}(J_{\mathrm{NN}}(x_i) - \mathrm{PMP}(x_i)),$$

where $\mathrm{PMP}(x_i)$ indicates the value of x_i along a given PMP trajectory. Choosing $N_{\mathrm{PMP}} = 100000$, we obtain $J_{\mathrm{NN}}(x)$ and its induced controller that globally stabilizes the pendulum in Fig. 4(b). In extra results we show the weakly supervised method generalizes to the 3-dimensional cart-pole.



(a) Supervised training with, left to right, 1000, 100, and 50 samples

(b) Weak supervision

Figure 4: Neural approximations of the optimal value function. (a) Supervised training with decreasing data samples. (b) Weakly supervised training with raw PMP trajectories.

5. Conclusion

We showed the optimal value function of infinite-horizon undiscounted pendulum swing-up is non-smooth. Motivated by this, theoretically, we provide two results that certify the optimality and sub-optimality of candidate value functions; algorithmically, we develop a numerical procedure based on backward solving PMP with local LQR terminal conditions to compute the true optimal value function up to minor numerical inaccuracies. The optimal value function outperforms other baseline algorithms and verified optimality. We demonstrate it is possible to learn simple and effective neural approximations of the optimal value function via either strong or weak supervision.

Acknowledgments

We thank Michael Posa, Jean-Bernard Lasserre, Jiarui Li, Yukai Tang, and Shucheng Kang for discussions about the optimal pendulum swing-up problem; Didier Henrion, Zexiang Liu, and Necmiye Ozay for pointing us to several related works; Lujie Yang and Alexandre Amice for help with computing polynomial lower bounds using SOS programming.

References

Karl Johan Åström and Katsuhisa Furuta. Swinging up a pendulum by energy control. *Automatica*, 36(2):287–295, 2000.

Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J Tomlin. Hamilton-jacobi reachability: A brief overview and recent advances. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2242–2253. IEEE, 2017.

Martino Bardi, Italo Capuzzo Dolcetta, et al. *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*, volume 12. Springer, 1997.

Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*, volume 4. Athena scientific, 2012.

Kenji Doya. Reinforcement learning in continuous time and space. *Neural computation*, 12(1): 219–245, 2000.

Siyuan Feng, Eric Whitman, X Xinjilefu, and Christopher G Atkeson. Optimization based full body control for the atlas robot. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 120–127. IEEE, 2014.

Felix Fiedler, Benjamin Karg, Lukas Lüken, Dean Brandner, Moritz Heinlein, Felix Brabender, and Sergio Lucia. do-mpc: Towards fair nonlinear and robust model predictive control. *Control Engineering Practice*, 140:105676, 2023.

John Hauser and Hinke Osinga. On the geometry of optimal control: the inverted pendulum example. In *Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148)*, volume 2, pages 1721–1726. IEEE, 2001.

Thomas Holzhüter. Optimal regulator for the inverted pendulum via euler–lagrange backward integration. *Automatica*, 40(9):1613–1620, 2004.

Max Jensen and Iain Smears. On the convergence of finite element methods for hamilton–jacobi–bellman equations. *SIAM Journal on Numerical Analysis*, 51(1):137–162, 2013.

Rushikesh Kamalapurkar, Patrick Walters, Joel Rosenfeld, and Warren Dixon. *Reinforcement learning for optimal feedback control*. Springer, 2018.

Ellya L Kawecki and Iain Smears. Convergence of adaptive discontinuous galerkin and c 0-interior penalty finite element methods for hamilton–jacobi–bellman and isaacs equations. *Foundations of Computational Mathematics*, 22(2):315–364, 2022.

Hassan K Khalil. *Nonlinear Systems*. Prentice Hall, 2002.

Jean B Lasserre, Christophe Prieur, and Didier Henrion. Nonlinear optimal control: approximations via moments and lmi-relaxations. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 1648–1653. IEEE, 2005.

Jean Bernard Lasserre, Didier Henrion, Christophe Prieur, and Emmanuel Trélat. Nonlinear optimal control via occupation measures and lmi-relaxations. *SIAM J. Control. Optim.*, 47:1643–1666, 2007.

Derong Liu, Ding Wang, Fei-Yue Wang, Hongliang Li, and Xiong Yang. Neural-network-based online hjb solution for optimal robust guaranteed cost control of continuous-time uncertain nonlinear systems. *IEEE transactions on cybernetics*, 44(12):2834–2847, 2014.

Michael Lutter, Boris Belousov, Kim Listmann, Debora Clever, and Jan Peters. Hjb optimal feedback control with deep differential value functions and action constraints. In *Conference on Robot Learning*, pages 640–650. PMLR, 2020.

Ian M Mitchell and Jeremy A Templeton. A toolbox of hamilton-jacobi solvers for analysis of nondeterministic continuous and hybrid systems. In *International workshop on hybrid systems: computation and control*, pages 480–494. Springer, 2005.

Rémi Munos and Andrew W. Moore. Barycentric interpolators for continuous space and time reinforcement learning. In *Neural Information Processing Systems*, 1998.

Remi Munos, Leemon C Baird, and Andrew W Moore. Gradient descent approaches to neural-net-based solutions of the hamilton-jacobi-bellman equation. In *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339)*, volume 3, pages 2152–2157. IEEE, 1999.

Nenad Muskinja and Boris Tovornik. Swinging up and stabilization of a real inverted pendulum. *IEEE transactions on industrial electronics*, 53(2):631–639, 2006.

Tenavi Nakamura-Zimmerer, Qi Gong, and Wei Kang. Adaptive deep learning for high-dimensional hamilton–jacobi–bellman equations. *SIAM Journal on Scientific Computing*, 43(2):A1221–A1247, 2021.

Stanley Osher and Ronald P Fedkiw. Level set methods: an overview and some recent results. *Journal of Computational physics*, 169(2):463–502, 2001.

Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *The Journal of Machine Learning Research*, 22(1):12348–12355, 2021.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

Tongzheng Ren, Zhaolin Ren, Na Li, and Bo Dai. Stochastic nonlinear control via finite-dimensional spectral dynamic embedding. *arXiv preprint arXiv:2304.03907*, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Alena Shilova, Thomas Delliaux, Philippe Preux, and Bruno Raffin. Revisiting continuous-time reinforcement learning. a study of hjb solvers based on pinns and fems. In *Sixteenth European Workshop on Reinforcement Learning*, 2023.

Jean-Jacques E Slotine, Weiping Li, et al. *Applied nonlinear control*, volume 199. Prentice hall Englewood Cliffs, NJ, 1991.

Iain Smears and Endre Suli. Discontinuous galerkin finite element approximation of hamilton–jacobi–bellman equations with cordes coefficients. *SIAM Journal on Numerical Analysis*, 52(2): 993–1016, 2014.

Eduardo D Sontag. *Mathematical control theory: deterministic finite dimensional systems*, volume 6. Springer Science & Business Media, 2013.

Tomomichi Sugihara, Yoshihiko Nakamura, and Hirochika Inoue. Real-time humanoid motion generation through zmp manipulation based on inverted pendulum control. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 2, pages 1404–1409. IEEE, 2002.

Yuval Tassa and Tom Erez. Least squares solutions of the hjb equation with neural network value-function approximators. *IEEE transactions on neural networks*, 18(4):1031–1041, 2007.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Russ Tedrake. Underactuated robotics: Learning, planning, and control for efficient and agile machines course notes for mit 6.832. *Working draft edition*, `https://underactuated.mit.edu/index.html`, 2009.

Richard Vinter. Convex duality and nonlinear optimal control. *SIAM journal on control and optimization*, 31(2):518–538, 1993.

Andreas Wernli and Gerald Cook. Suboptimal control for the nonlinear quadratic regulator problem. *Automatica*, 11(1):75–84, 1975.

Heng Yang. Optimal control and estimation. *Working draft edition*, `https://hankyang.seas.harvard.edu/OptimalControlEstimation/`, 2023.

Lujie Yang, Hongkai Dai, Alexandre Amice, and Russ Tedrake. Approximate optimal controller synthesis for cart-poles and quadrotors via sums-of-squares. *IEEE Robotics and Automation Letters*, 8(11):7376–7383, 2023.