2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

A Physics-Informed Neural Networks Framework to Solve the Infinite-Horizon Optimal Control Problem

Filippos Fotiadis, Kyriakos G. Vamvoudakis

Abstract— **In this work, we leverage physics-informed neural networks (PINNs) to approximately solve the infinite-horizon optimal control problem for nonlinear systems. Specifically, since PINNs are generally able to solve a class of partial differential equations, they can be employed to approximate the value function in the infinite-horizon optimal control problem, via solving the associated steady-state Hamilton-Jacobi-Bellman (HJB) equation. However, the issue with such a direct approach is that the steady HJB equation generally yields more than one solution, hence directly employing PINNs to solve it can lead to divergence of the method. To tackle this problem, we instead apply PINNs to a finite-horizon variant of the steady-state HJB equation which has a unique solution, and which uniformly approximates the infinite-horizon optimal value function as the horizon increases. A method to verify whether the selected horizon is large enough is also provided, as well as an algorithm to increase it with reduced computations if it is not. Unlike conventional methods, the proposed approach does not require knowledge of a stabilizing controller, the execution of computationally expensive iterations, or polynomial basis functions for approximation.**

I. Introduction

A common objective in control theory is the regulation of a dynamical system's state around some nominal point of operation. This objective is usually solvable through various and possibly infinitely many control designs, but in practice only a few of those are able to offer overall good performance, in terms of the control effort expended over an infinite horizon as well as the time taken to achieve regulation. The problem of finding the best-performing such control design is known as the infinite-horizon optimal control problem [1], or alternatively as the optimal stabilization problem [2].

From a mathematical point of view, solving the continuous-time infinite-horizon optimal control problem is equivalent to finding the so-called optimal value function of the problem, which is a solution of the steady-state HJB partial differential equation (PDE) [1]. However, solving the steady-state HJB is not straightforward; it is a nonlinear equation that admits many solutions [3], and hence is difficult to solve analytically. For this reason, considerable effort has been made to at least find its solutions approximately [4]–[9].

In the core of many methods that approximately solve the HJB equation is a procedure known as successive approximations, or policy iteration (PI) [4], [10]. This is an iterative algorithm that converges to the positive-definite solution of the steady-state HJB, and can be implemented using approximation structures [5], [10]. However, the iterative nature of PI can lead to increased computational complexity, and requires knowledge of an "initial admissible policy" that can stabilize the system. While the latter is a requirement that could possibly be relaxed, another drawback of PI and its variations is that they are almost always inapplicable unless the basis functions of the underlying approximator structure are polynomials [11]–[13]. This can be a problematic issue in practice, owing to the inherent global nature of polynomial basis functions.

A vastly different and emerging approach that could be used to solve the infinite-horizon optimal control problem, is that of physics-informed neural networks (PINNs). In particular, PINNs have been shown to be efficient in solving a certain class of PDEs [14], and could thus be employed to solve the underlying HJB equation of the optimal stabilization problem, while avoiding all of the aforementioned technicalities of PI. Indeed, this is an approach followed, for example, in [15], where PINNs were used to solve the steady-state HJB and compute the infinite-horizon optimal control law. However, since this HJB has many solutions, and as observed in [15], convergence can only be attained locally, i.e., the initial weights of the PINN need to be properly pre-trained. Otherwise, one may end up approximating a solution to the steady-state HJB that is totally unrelated to the optimal value function. Notably, [14] also points out that PINNs may not work well when applied to PDEs with multiple solutions.

Contributions: Motivated by these limitations, in this paper we develop a PINN-based procedure to approximate the optimal value function of the infinite-horizon optimal control problem, while avoiding convergence to other, unrelated solutions of the underlying HJB equation. In particular, since the steady-state HJB connected to the optimal control problem has many solutions, we instead employ PINNs to solve its finite-horizon variant that admits a unique solution. We justify this choice by proving that the unique solution of the finite-horizon HJB uniformly approximates the optimal value function if the length of the horizon is sufficiently large. A method to verify whether this horizon length is indeed large enough is also provided, as well as a procedure to extend it (in case it is not) with reduced computations. The proposed method does not suffer from the drawbacks of PI, which rarely works without polynomial basis functions, requires knowledge of an initially admissible control policy, and has to execute computationally expensive iterations.

Notation: The set $\mathbb{R}$ denotes the set of real numbers. The operator ∇_z is used to denote the gradient of a function, with respect to the argument implied by z. For a Lebesgue measurable set S, $|S|$ will denote this set's Lebesgue measure.

F. Fotiadis and K. G. Vamvoudakis are with The Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA. Email: {ffotiadis, kyriakos}@gatech.edu.

This work was supported in part, by ARO under grant No. W911NF-19 − 1 − 0270, by NSF under grant Nos. CAREER CPS-1851588, S&AS-1849198, and SATC-1801611, and by the Onassis Foundation-Scholarship ID: F ZQ 064 − 1/2020 − 2021.

II. Problem Formulation

Consider the continuous-time system:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t),\ x(0) = x_0,\ t \geqslant 0, \quad (1)$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^m$ is the control input, and $f(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$, $g(\cdot) : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are the system's dynamics functions. To guarantee the existence and uniqueness of solutions to (1), we assume that $f(\cdot),\ g(\cdot)$ are locally Lipschitz on $\mathbb{R}^n$. In addition, we assume $f(0) = 0$, so that the origin is an equilibrium point of (1) with $u = 0$.

Given a feedback policy $\mu : \mathbb{R}^n \to \mathbb{R}^m$, define its infinite horizon performance cost as:

$$J(x_0, \mu) = \int_0^\infty \Big(Q(x(\tau)) + r\left(\mu(x(\tau))\right)\Big)\mathrm{d}\tau, \quad (2)$$

where $Q : \mathbb{R}^n \to \mathbb{R}$ is a positive definite function, $r(\star) = \star^{\mathrm{T}} R \star$, $R \in \mathbb{R}^{m\times m}$ is a positive definite matrix, and the integration in (2) is over the trajectories of (1) under $u(t) = \mu(x(t))$. The integral (2) is well-defined for any $x_0 \in \Omega$, where $\Omega \subseteq \mathbb{R}^n$, if the policy μ is *admissible* on Ω. [5]

Definition 1. A control policy $\mu : \mathbb{R}^n \to \mathbb{R}^m$ is defined as *admissible* on a set $\Omega \subseteq \mathbb{R}^n$, and denoted as $\mu \in \Psi(\Omega)$, if it is continuous on Ω with $\mu(0) = 0$ and, given $x_0 \in \Omega$, $u = \mu$ asymptotically stabilizes (1) to the origin and the cost $J(x_0, \mu)$ is finite. □

Given the cost (2), the infinite-horizon optimal control problem is concerned with finding the admissible control policy $\mu^\star$ that minimizes it, satisfying

$$\mu^\star(x) := \arg\min_{\mu\in\Psi(\Omega)} J(x, \mu),\ \forall x \in \Omega.$$

The corresponding minimum cost value is denoted as $V^\star(x) = \min_{\mu\in\Psi(\Omega)} J(x, \mu)$, and is known as the optimal value function. In the case that this function is continuously differentiable, an explicit expression can be derived for $\mu^\star$. Specifically, by defining the Hamiltonian:

$$H(x, \nabla_x V(x), \mu(x)) = \nabla_x V^{\mathrm{T}}(x)(f(x) + g(x)\mu(x)) + Q(x) + \mu^{\mathrm{T}}(x)R\mu(x),$$

where $V = J(\cdot, \mu)$, and using the stationarity condition $\frac{\partial H}{\partial \mu} = 0$, one can derive

$$\mu^\star(x) = -\frac{1}{2}R^{-1}g^{\mathrm{T}}(x)\nabla_x V^\star(x). \quad (3)$$

An analogous expression for $V^\star$, though in a less explicit form, can also be obtained. To this end, note that if a value function V of a policy μ is continuously differentiable, then it satisfies a Lyapunov-like equation [1] of the form

$$\nabla_x V^{\mathrm{T}}(x)(f(x) + g(x)\mu(x)) + Q(x) + \mu^{\mathrm{T}}(x)R\mu(x) = 0,\ V(0) = 0.$$

Substituting $\mu = \mu^\star$ in this equation leads to the so-called Hamilton-Jacobi-Bellman (HJB) equation:

$$0 = \nabla_x V^{\star\mathrm{T}}(x)f(x) + Q(x) - \frac{1}{4}\nabla_x V^{\star\mathrm{T}}(x)g(x)R^{-1}g^{\mathrm{T}}(x)\nabla_x V^\star(x),\ V^\star(0) = 0, \quad (4)$$

a solution of which corresponds to the optimal value function $V^\star$. It is evident that if (4) is solved with respect to $V^\star$, then the optimal control $\mu^\star$ can be directly computed by simply evaluating (3). Nevertheless, the task of obtaining $V^\star$ from (4) is not a straightforward one. One of the main difficulties in this direction is the fact that (4) is a nonlinear partial differential equation that only implicitly describes $V^\star$. Therefore, it is practically impossible to derive an analytical expression of $V^\star$ from it, and most works of the literature focus on solving it approximately [5], [6], [16]. However, these works usually require either knowledge of an admissible control policy for (1), the execution of computationally expensive iterations, or the use of polynomial basis functions for the underlying approximation structure, all of which can be restrictive requirements in practice.

Motivated by the aforementioned, the purpose of this work is to propose an alternative method to approximate the optimal value function $V^\star$, which imposes neither of the aforementioned three requirements. The proposed method is directly influenced by the method of physics-informed neural networks (PINNs), which can approximate solutions to a certain class of nonlinear PDEs.

III. A PINNs-Based Solution to the Infinite-Horizon Optimal Control Problem

In this section, we propose a scheme to approximate the infinite-horizon optimal control $\mu^\star$ via the use of PINNs. In this direction, one of the main difficulties is the fact that the HJB equation (4) has multiple solutions, and only one of them corresponds to the optimal value function $V^\star$. Therefore, directly employing the method of PINNs to solve this equation can be problematic [14], owing to the possibility of the method converging to an unwanted solution.

A remedy to the above issue is possible if we take into account that (4) is the steady state version of another HJB equation, which admits only one solution. Specifically, consider the following, time-dependent HJB equation:

$$0 = \nabla_t V_T(x, t) + \nabla_x V_T^{\mathrm{T}}(x, t)f(x) + Q(x) - \frac{1}{4}\nabla_x V_T^{\mathrm{T}}(x, t)g(x)R^{-1}g^{\mathrm{T}}(x)\nabla_x V_T(x, t),\ V_T(x, T) = 0, \quad (5)$$

where $T > 0$. The unique solution V_T to this PDE corresponds to the optimal value function of the finite-horizon version (2), namely of:

$$J_T(x_0, \mu) = \int_0^T \Big(Q(x(\tau)) + r\left(\mu(x(\tau), \tau)\right)\Big)\mathrm{d}\tau, \quad (6)$$

where we note that the policy μ here is allowed to be time-varying. Therefore, as T increases, it is expected (see next section) that $V_T(x, 0)$ will converge pointwise to $V^\star(x)$, for all $x \in \mathbb{R}^n$. Accordingly, as T increases, it is expected that the control policy $\mu_T : \mathbb{R}^n \times [0, T] \to \mathbb{R}^m$ that minimizes (6), and is given by:

$$\mu_T(x, t) = -\frac{1}{2}R^{-1}g^{\mathrm{T}}(x)\nabla_x V_T(x, t) \quad (7)$$

will approach $\mu^\star$ as T increases, i.e. that $\mu_T(x, 0)$ will converge pointwise to $\mu^\star(x)$ for all $x \in \mathbb{R}^n$. One can thus proceed to approximate $\mu^\star$ and $V^\star$, by solving (5) via the

method of PINNs for a sufficiently large horizon $T > 0$. Of course, certain important questions arise when adopting this approach; see the end of this section for more details.

In the direction of approximating the solution V_T of (5) with the method of PINNs, and following [14], define the residual function for any $v_T : \Omega \times [0, T] \to \mathbb{R}$:

$$F_e(x, t; v_T) = \nabla_t v_T(x, t) + \nabla_x v_T^{\mathrm{T}}(x, t) f(x) + Q(x)$$
$$-\frac{1}{4} \nabla_x v_T^{\mathrm{T}}(x, t) g(x) R^{-1} g^{\mathrm{T}}(x) \nabla_x v_T(x, t),\ x \in \Omega, t \in [0, T],$$

where Ω is a compact set. This function is essentially a residual of the PDE (5), and is equal to zero if and only if v_T is a solution to (5) for any boundary condition. In a similar manner, another residual function is defined as

$$F_b(x; v_T) = v_T(x, T),\ x \in \Omega,$$

which is equal to zero if and only if v_T satisfies the boundary condition of (5). Hence, it follows that v_T is a solution to (5) if and only if $F_e(\cdot, \cdot; v_T)$ and $F_b(\cdot; v_T)$ are identically zero.

To find a solution to (5), a neural network $\hat{v}_T(\cdot, \cdot; w) : \Omega \times [0, T] \to \mathbb{R}$ is constructed, where $w \in \mathbb{R}^N$ denote the network's parameters, and $\Omega \subseteq \mathbb{R}^n$ is a compact set where approximation will take place. The parameters w are trained by attempting to force the residuals $F_e(x, t; \hat{v}_T)$, $F_b(x; \hat{v}_T)$ to be zero across a grid $(x_e^i, t_e^i) \in \Omega \times [0, T]$, $i = 1, \ldots, N_e$, and $x_b^i \in \Omega$, $i = 1, \ldots, N_b$. This is done by defining the following mean square error (MSE)

$$\mathrm{MSE} = \mathrm{MSE}_e + \mathrm{MSE}_b, \tag{8}$$

where

$$\mathrm{MSE}_e = \frac{1}{N_e} \sum_{i=1}^{N_e} F_e(x_e^i, t_e^i; \hat{v}_T)^2,$$

$$\mathrm{MSE}_b = \frac{1}{N_b} \sum_{i=1}^{N_b} F_b(x_b^i; \hat{v}_T)^2,$$

and training w so that the MSE given by (8) is minimized. The name *physics informed neural network* is usually used for the function $F_e(\cdot, \cdot, \hat{v}_T)$, because it is essentially a neural network with the same parameters w as $\hat{v}_T$, but whose activation functions have been "informed" about the physics of the system through the underlying HJB equation. Although strict theoretical guarantees of convergence do not exist for the method of PINNs, they have been empirically shown to perform well when the solution to the underlying PDE is unique and the neural network architecture is expressive enough [14].

Apparently, in the context of PINNs, dealing with the HJB equation (5) is much more convenient than handling the HJB (4). Unlike (4), the HJB equation (5) is exactly of the form for which PINNs have been developed [14]. In addition, given that V_T is continuously differentiable, then it is certain that (5) admits only one solution, equal to V_T; hence, the possibility of the PINN approximating a function entirely different than V_T is excluded. Nevertheless, simply substituting the infinite-horizon HJB with a finite-horizon one with a large horizon, and anticipating that the solution $V_T(\cdot, 0)$ of the latter will be close to the solution $V^\star$ of the former, could possibly prove to be a careless action. In that respect, a couple of important questions must be answered to validate such an approach:

1) As the horizon T increases, do the functions $V_T(\cdot, 0)$ and $\mu_T(\cdot, 0)$ provide *uniform* approximations of the optimal value function $V^\star(\cdot)$ and control $\mu^\star(\cdot)$?
2) How can one evaluate, for a fixed T, whether the derived control policy $\mu_T(\cdot, 0)$ is close enough to the infinite-horizon optimal control policy $\mu^\star(\cdot)$? In addition, if it is concluded that $\mu_T(\cdot, 0)$ and $V_T(\cdot, 0)$ are not close enough to $\mu^\star(\cdot)$ and $V^\star(\cdot)$, how can these functions be used to obtain better ones without completely restarting the training process of the PINN?

IV. Uniform Approximation of $V^\star$ and $\mu^\star$ using the Finite-Horizon HJB

This section shows that as T increases, $V_T(\cdot, 0)$ indeed provides a uniform approximation of $V^\star(\cdot)$. The following theorem is the most important step towards this direction, showing that $V_T(\cdot, 0)$ and $\mu_T(\cdot, 0)$ *pointwise* approximate $V^\star(\cdot)$ and $\mu^\star(\cdot)$.

Theorem 1. *For all $x \in \mathbb{R}^n$, the sequence $V_T(x, 0)$ is increasing with respect to T and upper bounded by $V^\star(x)$, i.e., for every real $T_2 \geqslant T_1 > 0$, it holds that:*

$$V_{T_1}(x, 0) \leqslant V_{T_2}(x, 0) \leqslant V^\star(x),\ \forall x \in \mathbb{R}^n.$$

In addition,

$$\lim_{T \to \infty} V_T(x, 0) = V^\star(x),$$
$$\lim_{T \to \infty} \mu_T(x, 0) = \mu^\star(x).$$

Proof. The proof is omitted due to space limitations. ∎

Given the monotonicity and pointwise convergence properties stated in Theorem 1, it then follows using analysis results that, in fact, $V_T(\cdot, 0)$ and $\mu_T(\cdot, 0)$ provide *uniform* and *almost uniform* approximations of the optimal value function $V^\star(\cdot)$ and control $\mu^\star(\cdot)$, over any compact subset $\Omega \subset \mathbb{R}^n$.

Corollary 1. *Let $\Omega \subset \mathbb{R}^n$ be compact. Then, $V_T(\cdot, 0) \to V^\star$ uniformly and $\mu_T(\cdot, 0) \to \mu^\star$ almost uniformly on Ω as $T \to \infty$, i.e., for every $\epsilon > 0$ there exists $T^\star > 0$ and a measurable set $\Omega_\epsilon \subset \Omega$ with Lebesgue measure $|\Omega_\epsilon| < \epsilon$, such that if $T \geqslant T^\star$ then:*

$$\sup_{x \in \Omega} |V_T(x, 0) - V^\star(x)| < \epsilon,$$
$$\sup_{x \in \Omega \setminus \Omega_\epsilon} |\mu_T(x, 0) - \mu^\star(x)| < \epsilon.$$

Proof. From Theorem 1, the sequence of continuous functions $V_T(\cdot, 0)$ is increasing and converges pointwise everywhere on Ω to the continuous function $V^\star$. Since Ω is compact, it follows from Dini's theorem that the mode of convergence is uniform [17]. Next, since $\mu_T(\cdot, 0)$ converges pointwise everywhere to $\mu^\star$ and the set Ω is Lebesgue measurable with finite measure, it follows from Egorov's theorem [18] that for every $\epsilon > 0$ there exists a measurable set $\Omega_\epsilon \subset \Omega$ with Lebesgue measure $|\Omega_\epsilon| < \epsilon$, such that $\mu_T(\cdot, 0)$ converges uniformly to $\mu^\star$ on $\Omega \setminus \Omega_\epsilon$. ∎

Combining Corollary 1 with the universal approximation property of neural networks, we conclude that approximating the finite-horizon value function V_T with PINNs for a large horizon T is a valid procedure towards estimating the infinite horizon value function $V^\star$ over a compact set.

V. Evaluating a Good Horizon Length T

In the previous section, the finite horizon value function $V_T(\cdot,0)$ was shown to uniformly approximate the infinite-horizon one, namely $V^\star$. Therefore, employing the method of PINNs to obtain an estimate of $V_T(\cdot,0)$ is a valid procedure towards obtaining an estimate of $V^\star$, as long as the horizon length T is large. This brings forward another question: how can we evaluate whether the horizon T is large enough, so that $\mu_T(\cdot,0)$, $V_T(\cdot,0)$ are close enough to $\mu^\star$, $V^\star$?

A natural way to check whether $V_T(\cdot,0)$ is close enough to $V^\star$ is by verifying whether it satisfies the infinite-horizon HJB (4), since $V^\star$ is in fact a solution to that equation. Towards this end, one can define the following flow residual:

$$E_e(x;V_T(\cdot,0)) = \nabla_x V_T^{\mathrm{T}}(x,0)f(x) + Q(x) - \frac{1}{4}\nabla_x V_T^{\mathrm{T}}(x,0)g(x)R^{-1}g^{\mathrm{T}}(x)\nabla_x V_T(x,0). \quad (9)$$

This is essentially an error indicating how far $V_T(\cdot,0)$ is from satisfying the infinite-horizon HJB (4) at the point x; it is zero if and only if $V_T(\cdot,0)$ solves (4) at this specific point. Accordingly, one can define the boundary residual:

$$E_b(V_T(\cdot,0)) = V_T(0,0),$$

which is zero if and only if $V_T(0,0)=0$, i.e. if and only if $V_T(\cdot,0)$ satisfies the boundary condition of the infinite-horizon HJB (4). Therefore, by aggregating the residuals into a single error term:

$$E = \frac{1}{N_c}\sum_{i=1}^{N_c} E_e(x_c^i;V_T(\cdot,0))^2 + E_b(V_T(\cdot,0))^2, \quad (10)$$

where $x_c^i \in \Omega$, $i=1,\dots,N_c$, a procedure to check whether $V_T(\cdot,0)$ provides a good approximation to $V^\star$ would be to check how close E is to zero.

Remark 1. Since PINNs are used to approximate V_T, only an approximation $\hat{V}_T$ of V_T is available. Therefore, in practice, one would only be able to compute the residuals $\hat{E} = \frac{1}{N_c}\sum_{i=1}^{N_c} E_e(x_e^i;\hat{V}_T(\cdot,0))^2 + E_b(\hat{V}_T(\cdot,0))^2$, and check whether those are close to zero or not. Since $\hat{V}_T$ would only approximate V_T, a nonzero residual may not necessarily imply that T is not large enough, but it could mean that the underlying neural network architecture is not expressive enough to sufficiently approximate V_T. □

In case that the horizon T is deemed to be small, by means of the residual error E being large, then one can increase this horizon to get a better approximation of $V^\star$. In that respect, one could recompute $V_{T'}$ from scratch for some larger horizon $T'>T$ by reemploying the PINNs method of Section III over the larger domain $\Omega\times[0,T'] \supset \Omega\times[0,T]$. However, this can be a tenuous procedure, and in fact unnecessary; instead, the value function V_T that has already been computed for a smaller horizon can be used as an aid to find $V_{T'}$, $T'>T$, without resolving the whole problem from scratch over $[0,T']$, but rather by solving another smaller problem only over $[0,T'-T]$. To see this, consider the finite-horizon cost functional, which is similar to (6) but augmented with a terminal cost dependent on V_T:

$$J'(x_0,\mu) = \int_0^{T'-T} \Big(Q(x(\tau)) + r\left(\mu(x(\tau),\tau)\right)\Big)\mathrm{d}\tau + V_T(x(T'-T),0).$$

The optimal value function $V' : \mathbb{R}^n \times [0,T'-T] \to \mathbb{R}$ and control $\mu' : \mathbb{R}^n \times [0,T'-T] \to \mathbb{R}^m$ of this problem satisfy:

$$0 = \nabla_t V'(x,t) + \nabla_x V'^{\mathrm{T}}(x,t)f(x) + Q(x) - \frac{1}{4}\nabla_x V'^{\mathrm{T}}(x,t)g(x)R^{-1}g^{\mathrm{T}}(x)\nabla_x V'(x,t), \quad (11)$$
$$V'(x,T'-T) = V_T(x,0),$$

and

$$\mu'(x,t) = -\frac{1}{2}R^{-1}g^{\mathrm{T}}(x)\nabla_x V'(x,t). \quad (12)$$

Notice that these two functions are defined only over $t\in[0,T'-T]$ and not over $t\in[0,T']$, and the same holds for the corresponding PDE (11). In what follows, we show that, in fact, $V'(\cdot,0)=V_{T'}(\cdot,0)$ and $\mu'(\cdot,0)=\mu_{T'}(\cdot,0)$.

Theorem 2. *It holds that* $V'(\cdot,0)=V_{T'}(\cdot,0)$, *and* $\mu'(\cdot,0)=\mu_{T'}(\cdot,0)$.

Proof. The proof is omitted due to space limitations. ■

Based on Theorem 2, if one knows the value function $V_T(\cdot,0)$ for some horizon length $T>0$, then they can compute the value function $V_{T'}(\cdot,0)$ over a larger horizon $T'>T$ by solving the PDE (11). This PDE is defined only over a time length of $T'-T$, so it is significantly less tenuous to deal with than resolving the PDE (5) from scratch. In that respect, if one needs to increase the horizon length T so that $V_T(\cdot,0)$ better approximates $V^\star$, it would be less tenuous to apply the PINN method to (11), instead of applying it from scratch to (5) for a larger horizon.

Remark 2. The PDE (11) can be approximately solved similarly to (5), by sampling training points on the flow and the boundary condition and defining a loss function of the form (8). □

VI. Simulations

Consider a Van der Pol oscillator [19], with $f(x) = \begin{bmatrix} x_1 & -x_1 - \frac{1}{2}x_2(1-x_1^2)\end{bmatrix}^{\mathrm{T}}$ and $g(x) = \begin{bmatrix}0 & x_1\end{bmatrix}^{\mathrm{T}}$, where $x=[x_1\ x_2]^{\mathrm{T}}$ is the state. If the infinite horizon performance cost is chosen so that $Q(x)=x_2^2$ and $R=1$, then it is known by the converse optimal control problem that the optimal value function is given by $V^\star(x)=x_1^2+x_2^2$, and the optimal control by $\mu^\star(x)=-x_1x_2$ [19]. Note that while Q is only a positive semi-definite function, this is no issue as it is zero-state observable.

A. Scheme Validation

To approximate $V^\star$ over the compact set $\Omega=[-1,1]\times[-1,1]$, a 3-layer neural network is employed, where each

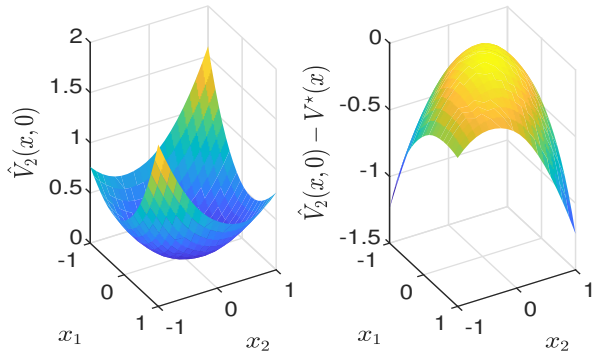

Fig. 1. The learnt value function $\hat{V}_2(\cdot,0)$ (left) and the error from optimality $\hat{V}_2(\cdot,0) - V^\star(\cdot)$ (right), for $T = 2$.

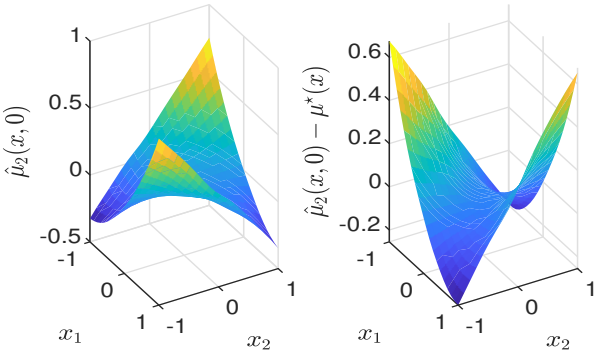

Fig. 2. The learnt control policy $\hat{\mu}_2(\cdot,0)$ (left) and the error from optimality $\hat{\mu}_2(\cdot,0) - \mu^\star(\cdot)$ (right), for $T = 2$.

layer consists of 150 neurons, with the corresponding activation function chosen to be the hyperbolic tangent. This network is trained by defining the loss function to be the residual (8) of the finite-horizon HJB with horizon length $T = 2$, with $N_e = 10000$ flow training points and $N_b = 1600$ boundary training points. This loss function is then minimized using Adam's algorithm [20], with an initial learning rate equal to 0.01, decaying at a rate of 0.002 each time a mini-batch of 100 flow training points is used. The procedure terminates once the loss function becomes less than 10^{-3} for each mini-batch.

The results can be seen in Figures 1-2. Figure 1 shows the approximation of the finite-horizon value function $V_2(\cdot,0)$ as well as the distance of this approximation from the optimal value function $V^\star(\cdot)$, while Figure 2 depicts the same information regarding the control policy. In general, it can be seen that while a modest approximation of $V^\star(\cdot)$ and $\mu^\star(\cdot)$ is attained, the optimality gap is still large, because the value of the horizon T was not chosen to be large enough. This can also be verified by calculating the residual (10) over $N_c = 400$ testing points; its value is 0.0352, which is 35 times larger than the desirable threshold.

To derive a better approximation of $V^\star(\cdot)$ and $\mu^\star(\cdot)$, the estimates of $V_2(\cdot,0)$ and $\mu_2(\cdot,0)$ are used to obtain an estimate of $V_4(\cdot,0)$ and $\mu_4(\cdot,0)$ using the method of Section V, and this process is repeated 3 additional times in order to derive an estimate of $V_{10}(\cdot,0)$ and $\mu_{10}(\cdot,0)$.

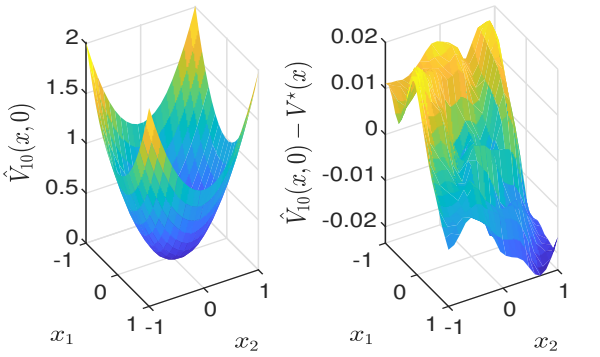

Fig. 3. The learnt value function $\hat{V}_{10}(\cdot,0)$ (left) and the error from optimality $\hat{V}_{10}(\cdot,0) - V^\star(\cdot)$ (right).

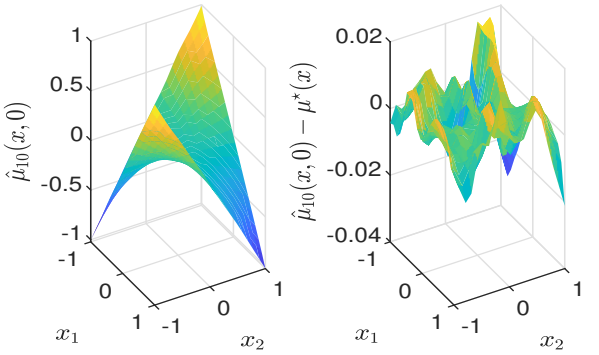

Fig. 4. The learnt control policy $\hat{\mu}_{10}(\cdot,0)$ (left) and the error from optimality $\hat{\mu}_{10}(\cdot,0) - \mu^\star(\cdot)$ (right).

The results can be seen in Figures 3-4. Figure 3 shows the approximation of the finite horizon value function $V_{10}(\cdot,0)$ as well as the distance of this approximation from the optimal value function $V^\star(\cdot)$, while Figure 4 depicts the same information regarding the control policy. Evidently, the computed functions approximate $V^\star(\cdot)$ and $\mu^\star(\cdot)$ much more closely than the estimates of $V_2(\cdot,0)$ and $\mu_2(\cdot,0)$. This also becomes clear after re-calculating the residual (10), the value of which now is 0.00026 – more than 100 times smaller than the value 0.0352 previously calculated.

B. Comparison with the Direct Approach

Next, we show that if we do not use the proposed approach to approximate the optimal value function and control, but instead employ the method of PINNs directly on the steady-state HJB, then issues of convergence can arise. To this end, we consider again the Van der Pol system of the previous subsection, with the difference that the PINN is now employed to directly approximate a solution of (4). All parameters are chosen as in the preceding simulation, and the approximated optimal value function $\hat{V}$ and optimal control $\hat{\mu}$ are shown in Figures 5-6.

Clearly, while convergence has been attained – at least for the value function – in some areas of the state space, in other areas the estimation is far from being accurate. Intuition regarding this result can be obtained from Figure 7, which shows the squared form $E_e(x;\hat{V})^2$ of the flow residual (9). We particularly observe that the PINN essentially approxi-

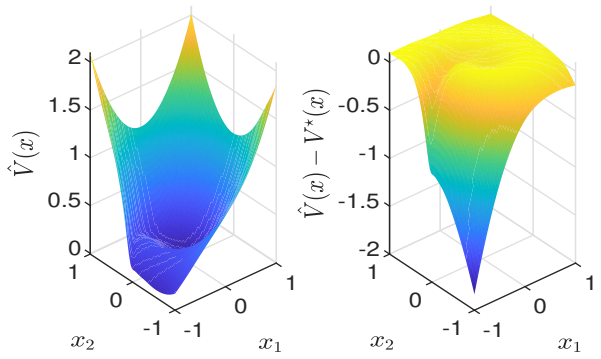

Fig. 5. The learnt value function $\hat{V}(\cdot)$ (left) and the error from optimality $\hat{V}(\cdot) - V^\star(\cdot)$ (right), with the direct approach.

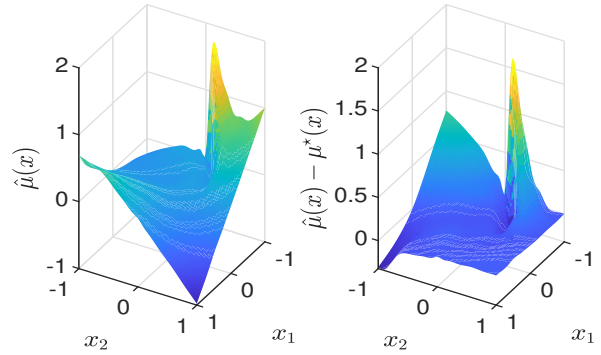

Fig. 6. The learnt control policy $\hat{\mu}(\cdot)$ (left) and the error from optimality $\hat{\mu}(\cdot) - \mu^\star(\cdot)$ (right), with the direct approach.

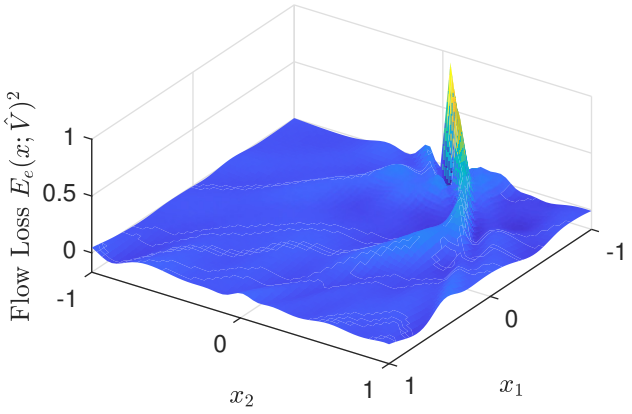

Fig. 7. The squared form $E_e(x;\hat{V})^2$ of the flow residual (9), with the direct approach.

mated a combination of solutions of the steady-state HJB: the loss function is close to zero almost everywhere, apart from what looks like an "1-dimensional" manifold where the two solutions intersect. Because the area of this manifold is quite small relative to the area of $[-1,1]\times[-1,1]$, the mean squared error still managed to become less than 10^{-3}, leading to the termination of the learning procedure.

VII. Conclusion

This work utilized PINNs to solve the infinite-horizon optimal control problem. Because the underlying HJB equation associated with this problem has many solutions, the method of PINNs was employed on its finite-horizon variant instead, which has a unique solution that uniformly approximates the infinite-horizon value function. A method to verify whether the length of the horizon is large enough was also provided, as well as an algorithm to extend this horizon with reduced computations. Future work includes extending the proposed approach to solve differential games.

References

[1] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal control*. John Wiley & Sons, 2012.

[2] W. M. Haddad and V. Chellaboina, "Nonlinear dynamical systems and control," in *Nonlinear Dynamical Systems and Control*, Princeton university press, 2011.

[3] P. Deptula, Z. I. Bell, E. A. Doucette, J. W. Curtis, and W. E. Dixon, "Data-based reinforcement learning approximate optimal control for an uncertain nonlinear system with control effectiveness faults," *Automatica*, vol. 116, p. 108922, 2020.

[4] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2042–2062, 2017.

[5] M. Abu-Khalaf and F. L. Lewis, "Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network hjb approach," *Automatica*, vol. 41, no. 5, pp. 779–791, 2005.

[6] K. G. Vamvoudakis and F. L. Lewis, "Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.

[7] H. Modares, F. L. Lewis, and M.-B. Naghibi-Sistani, "Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks," *IEEE Transactions on neural networks and learning systems*, vol. 24, no. 10, pp. 1513–1525, 2013.

[8] R. Kamalapurkar, J. A. Rosenfeld, and W. E. Dixon, "Efficient model-based reinforcement learning for approximate online optimal control," *Automatica*, vol. 74, pp. 247–258, 2016.

[9] M. Mazouchi, Y. Yang, and H. Modares, "Data-driven dynamic multiobjective optimal control: An aspiration-satisfying reinforcement learning approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 11, pp. 6183–6193, 2021.

[10] R. W. Beard, G. N. Saridis, and J. T. Wen, "Approximate solutions to the time-invariant hamilton–jacobi–bellman equation," *Journal of Optimization theory and Applications*, vol. 96, pp. 589–626, 1998.

[11] M. Sassano, T. Mylvaganam, and A. Astolfi, "Model-based policy iterations for nonlinear systems via controlled hamiltonian dynamics," *IEEE Transactions on Automatic Control*, 2022.

[12] M. L. Greene, Z. I. Bell, S. Nivison, and W. E. Dixon, "Deep neural network-based approximate optimal tracking for unknown nonlinear systems," *IEEE Transactions on Automatic Control*, 2023.

[13] F. Fotiadis, A. Kanellopoulos, K. G. Vamvoudakis, and J. Hugues, "Impact of sensor and actuator clock offsets on reinforcement learning," in *2022 American Control Conference (ACC)*, pp. 2669–2674, IEEE, 2022.

[14] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[15] R. Furfaro, A. D'Ambrosio, E. Schiassi, and A. Scorsoglio, "Physics-informed neural networks for closed-loop guidance and control in aerospace systems," in *AIAA SCITECH 2022 Forum*, p. 0361, 2022.

[16] Y. Jiang and Z.-P. Jiang, "Robust adaptive dynamic programming and feedback stabilization of nonlinear systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 882–893, 2014.

[17] W. Rudin *et al.*, *Principles of mathematical analysis*, vol. 3. McGraw-hill New York, 1976.

[18] G. B. Folland, *Real analysis: modern techniques and their applications*, vol. 40. John Wiley & Sons, 1999.

[19] V. Nevistić and J. A. Primbs, "Constrained nonlinear optimal control: a converse hjb approach," 1996.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.