2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Embedding Learning-based Optimal Controllers with Assured Safety

Filippos Fotiadis[1], George A. Rovithakis[2], Kyriakos G. Vamvoudakis[1]

2024 IEEE 63rd Conference on Decision and Control (CDC) | 979-8-3503-1633-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/CDC56724.2024.10885951

***Abstract*— We consider an off-policy reinforcement learning algorithm that gathers input and state data from a nonlinear system, and uses them to approximate the infinite-horizon optimal control for that system. However, as this algorithm relies on neural networks, its convergence depends on restrictive assumptions regarding the underlying neural network structure. Moreover, the derived approximate optimal controller that it yields is stabilizing only within a compact set Ω of the state space, leading to significant issues of robustness and safety for real-world implementations. Motivated by this, to increase the robustness and safety guarantees of controllers obtained by off-policy reinforcement learning procedures, we combine them with a novel *safety net*. The safety net is minimally interfering, leaving the approximate optimal controller unaltered within the compact set Ω in which it is valid and stabilizing. On the other hand, the safety net interferes with the approximate optimal controller whenever the set Ω is violated, so as to guarantee the boundedness and integrity of the closed loop. Since the proposed net is model-agnostic yet learning-free, it provides, for the first time, hard guarantees of safety, established by rigorous theoretical analysis and subsequently verified in simulations.**

I. Introduction

Designing control laws to regulate nonlinear systems around a nominal point of operation is, without a doubt, one of the most important goals of the field of control theory. In many cases, however, it is equally important to guarantee that these laws are optimal, providing a perfect balance between the time needed to attain system regulation and the overall control effort expended in the closed loop. In the literature, this is often known as the infinite-horizon optimal control problem [1], or as the optimal stabilization problem [2].

The design of optimal controllers, while highly desirable, can become challenging when the dynamics of the underlying system are unknown. In such restrictive settings, reinforcement learning (RL) [3] has emerged as a significant tool to obtain optimal decision-making policies exclusively through data gathered from the system, i.e., through control input and state trajectories. In fact, RL procedures have already been tailored to the optimal stabilization problem of systems with unknown dynamics, typically employing actor-critic neural networks to approximate the optimal control and its associated cost function over a compact set Ω [4]–[9].

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

[2]G. A. Rovithakis is with the Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Thessaloniki, Greece. E-mail: rovithak@ece.auth.gr.

This work was supported in part, by Minerva under grant No. N00014 − 18 − 1 − 2874, by NSF under grant Nos. CAREER CPS-1851588, CPS-2227185, S&AS-1849198, and SATC-2231651, by the Onassis Foundation-Scholarship ID: F ZQ 064 − 1/2020 − 2021, and by the European Union's Horizon Framework Programme for Research and Innovation under grant agreement No. 101120823.

A shortcoming of controllers derived from actor-critic procedures is that they lack robustness guarantees. That is, since these controllers are optimal and stabilizing only in a compact set Ω, any violation of that set during runtime owed to, e.g., a disturbance, is likely to put closed-loop stability and safety in jeopardy. In addition, even if the set Ω is not violated during runtime, the stabilizing properties of actor-critic controllers on Ω rely on restrictive assumptions about the structure of the actor-critic network, such as its size and the selection of its basis functions [4]. These assumptions are typically impossible to verify in advance, leading to significant safety implications for real-world implementations.

It is important to mention that the concept of safety in learning-based optimal stabilization is far from being new. In particular, there is much prior work on augmenting optimal stabilization problems with safety specifications, typically via barrier functions, which allow the incorporation of state constraints within the cost function of the problem [10]–[14]. Nevertheless, since the barrier-augmented optimal stabilization problem is eventually solved using neural networks, all safety assurances still rely on restrictive assumptions regarding the underlying neural network structure and its approximation capabilities. That is, the neural network should be large enough, with cleverly chosen basis functions.

Contributions: Motivated by these limitations, we embed controllers derived from off-policy reinforcement learning with a safety net, to tackle their lack of robustness that stems from the use of neural networks. The safety net we propose is minimally interfering, leaving the learning-based controller unaltered in the compact set Ω where it is valid and stabilizing. On the other hand, the safety net intervenes when the set Ω is violated, to guarantee the safety and integrity of the closed loop. As one of its most important qualities, the safety net is model-agnostic but does not use neural networks. This allows us to establish, for the first time, hard guarantees on its ability to assure safety – even if the underlying learning-based controller performs poorly.

Notation: The symbol ∇ denotes the gradient of a function. For any two matrices A, B, $A \otimes B$ denotes their Kronecker product. For a vector $\psi \in \mathbb{R}^n$, $\psi \otimes_h \psi$ denotes its half-vectorized Kronecker product, i.e., $\psi \otimes_h \psi = [\psi_1^2 \ \psi_1\psi_2 \ \dots \ \psi_1\psi_n \ \psi_2^2 \ \psi_2\psi_3 \ \dots \ \psi_n^2]^{\mathrm{T}}$. For a symmetric matrix A, $\lambda_{\min}(A)$ denotes the minimum eigenvalue of A. The matrix I_n denotes the identity matrix of order n.

II. Preliminaries and Problem Formulation

Consider the continuous-time nonlinear system:

$$\begin{aligned} \dot{x}_j &= f_j(\bar{x}_j) + g_j(\bar{x}_j)x_{j+1}, \ j = 1, \dots, n-1, \\ \dot{x}_n &= f_n(\bar{x}_n) + g_n(\bar{x}_n)u, \end{aligned} \tag{1}$$

where $x_j \in \mathbb{R}$ is the j-th state with initial value $x_j(0) = x_{j,0}$, $\bar{x}_j = [x_1 \ \dots \ x_j]^{\mathrm{T}} \in \mathbb{R}^j$, $u \in \mathbb{R}$ is the control input, and $f_j, g_j : \mathbb{R}^j \to \mathbb{R}$ are locally Lipschitz functions modeling the system's dynamics, with unknown analytical expressions. System (1) can also be written in the compact form

$$\dot{x} = f(x) + g(x)u, \ x(0) = x_0, \tag{2}$$

with $x = [x_1 \ \dots \ x_n]^{\mathrm{T}}$, $x_0 = [x_{1,0} \ \dots \ x_{n,0}]^{\mathrm{T}}$, $f(x) = [f_1(x_1) + g_1(x_1)x_2, \ \dots, \ f_n(\bar{x}_n)]^{\mathrm{T}}$, and $g(x) = [0 \ \dots \ g_n(\bar{x}_n)]^{\mathrm{T}}$.

We assume that $f_j(0) = 0$ for all $j = 1, \dots, n$, so that the origin is an equilibrium point of (1) with $u = 0$. In addition, we require the following standard controllability assumption.

Assumption 1. The functions $g_j(\cdot)$ are either strictly positive or strictly negative. Without loss of generality, we assume $g_j(\cdot) \geqslant \underline{g}_j$ for some unknown $\underline{g}_j > 0$. □

A. Optimal Control and Policy Iteration

For any feedback control policy $\mu : \mathbb{R}^n \to \mathbb{R}$, define its performance cost as

$$J(x_0, \mu) = \int_0^\infty \left(Q(x) + r\mu^2\right) \mathrm{d}\tau, \tag{3}$$

where $Q : \mathbb{R}^n \to \mathbb{R}$ is positive definite, $r > 0$, and the integration is taken over the trajectories of (2) with $u = \mu$. The purpose of the infinite-horizon optimal control problem is to find the policy $\mu^\star : \mathbb{R}^n \to \mathbb{R}$ that minimizes the cost (3) over a set $\Omega \subseteq \mathbb{R}^n$. In other words, we want to find

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

where $\Psi(\Omega)$ denotes the set of admissible control policies, i.e., policies for which the integral (3) is well-defined $\forall x_0 \in \Omega$. The corresponding minimum value of the cost is called the optimal value function, and denoted as $V^\star(x) = J(x, \mu^\star)$.

Towards finding $\mu^\star$ and $V^\star$, note that if the value function $V_\mu(\cdot) := J(\cdot, \mu)$ of an admissible controller μ is continuously differentiable, then it satisfies the following Lyapunov-like partial differential equation (PDE) [1]:

$$\nabla V_\mu^{\mathrm{T}}(x)(f(x)+g(x)\mu(x))+Q(x)+r\mu^2(x) = 0, \ V_\mu(0) = 0. \tag{4}$$

Defining the Hamiltonian

$$H(x, \mu, \nabla V_\mu)=\nabla V_\mu(x)^{\mathrm{T}}(f(x)+g(x)\mu(x))+Q(x)+r\mu^2(x),$$

standard optimal control theory requires that $\mu^\star = \arg\min_\mu H(x, \mu, \nabla V^\star)$ [1]. Employing the stationarity condition to solve this minimization, we obtain

$$\mu^\star(x) = -\frac{1}{2r} g^{\mathrm{T}}(x) \nabla V^\star(x). \tag{5}$$

Combining (5) and (4) under $\mu = \mu^\star$ yields the so-called Hamilton-Jacobi-Bellman PDE:

$$\nabla V^{\star\mathrm{T}}(x) f(x) + Q(x) - \frac{1}{4r} \nabla V^{\star\mathrm{T}} g(x) g^{\mathrm{T}}(x) \nabla V^\star(x) = 0, \quad V^\star(0) = 0. \tag{6}$$

Algorithm 1 Policy Iteration

1: Begin with $\mu_0 \in \Psi(\Omega)$, $\epsilon > 0$, $i = 0$.
2: **repeat**
3: *Policy Evaluation*: Solve for V_i, $\forall x \in \Omega$, from

$$\nabla V_i^{\mathrm{T}}(x)(f(x) + g(x)\mu_i(x)) + Q(x) + r\mu_i^2(x) = 0, \ V_i(0) = 0. \tag{7}$$

4: *Policy improvement*: Let the new policy be given by

$$\mu_{i+1}(x) = -\frac{1}{2r} g^{\mathrm{T}}(x) \nabla V_i(x). \tag{8}$$

5: Set $i = i + 1$.
6: **until** $i \geqslant 2 \ \ \& \ \ \sup_{x \in \Omega} |V_{i-1}(x) - V_{i-2}(x)| < \epsilon$.

Apparently, if one can solve the HJB equation (6) for $V^\star$, then the optimal control policy can be directly computed from (5). Nevertheless, (6) is a nonlinear and complex PDE for which analytical solutions are almost always impossible to obtain. For this reason, *policy iteration* (PI) [15], a procedure that iteratively evaluates a given controller and then improves that controller, is usually employed to at least find $V^\star$ approximately. This procedure is summarized in Algorithm 1, and is known to converge to $V^\star$ [15].

B. Learning-based PI

While Algorithm 1 approximates the optimal value function and control, it is not implementable as it clearly requires knowledge of the system's dynamics functions f and g. Nevertheless, this requirement of system knowledge can be waived by following the steps of [4]. In particular, note that (2) can be written for any $i \in \mathbb{N}$ as

$$\dot{x} = f(x) + g(x)\mu_i(x) + g(x)(u - \mu_i(x)).$$

The derivative of V_i in Algorithm 1 then is

$$\dot{V}_i = \nabla V_i^{\mathrm{T}}(x)(f(x)+g(x)\mu_i(x))+\nabla V_i^{\mathrm{T}}(x)g(x)(u-\mu_i(x)).$$

Using (7)-(8), this equation turns into

$$\dot{V}_i = -Q(x) - r\mu_i^2(x) - 2r\mu_{i+1}(x)(u - \mu_i(x)). \tag{9}$$

Integrating (9) over any interval $[t_k, \ t'_k]$ for some positive instances $t'_k > t_k \geqslant 0$, $k \in \{0, \dots, K\} =: \mathcal{K}$, we obtain the following data-driven equation for expressing V_i and μ_{i+1}:

$$V_i(x(t'_k)) - V_i(x(t_k)) + \int_{t_k}^{t'_k} \Big(2r\mu_{i+1}(x)(u - \mu_i(x)) + Q(x) + r\mu_i^2(x)\Big) \mathrm{d}\tau = 0. \tag{10}$$

Various methods have been proposed in the literature to solve (10) for V_i and μ_{i+1} using neural networks [4], with convergence guarantees. As a result, Algorithm 1 can be implemented in a learning-based manner, where instead of obtaining V_i and μ_{i+1} from the model-based equations (7)-(8), one instead obtains them from the data-driven equation (10) for various values of $k \in \mathbb{N}$.

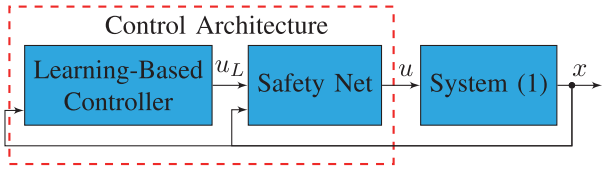

Fig. 1. The proposed control architecture.

C. Problem Formulation

In the core of all implementations of learning-based PI is the need to employ neural networks to solve for V_i and μ_{i+1} in (10) [4]. This requirement creates several issues for the derived approximate optimal controller, such as

1) the derived controller lacking robustness to uncertainties, as those are not considered in the training model;
2) the derived controller being stabilizing only within a compact set Ω of the state space;
3) the derived controller being stabilizing and close to optimality only if the underlying neural network is large enough, and its basis functions properly chosen [4], both of which are requirements that are not verifiable a priori.

The implication of these shortcomings is that when an approximately optimal controller obtained by learning-based PI is applied to (1), in practice there is no way to verify that it will be safe, not leading the system to instability.

Problem Statement: Motivated by the aforementioned, the purpose of this paper is to embed learning-based optimal controllers derived from PI with a safety net. The safety net, as shown in Fig. 1, takes the learning-based optimal control as an input, and outputs a modified control signal for the system, with the following characteristics:

1) it should be continuously differentiable;
2) it should be identically equal to the learning-based optimal control within the compact set Ω;
3) it should guarantee that the system state $x(t)$ remains bounded for all $t \geqslant 0$, irrespectively of the learning-based controller derived from PI; and
4) it should not employ neural networks in addition to those used by the learning-based controller, thus keeping the computational complexity at low levels.

The remainder of this paper is focused on deriving this safety-embedded scheme, depicted in Fig. 1.

III. The Learning-based Controller Block

In this section, we present the design of the learning-based controller block of Fig. 1, which uses input and state data to approximate the optimal control $u^\star$ and value function $V^\star$ in the set Ω. To that end, note that since V_i and μ_{i+1} in (10) are function variables, an approximation has to take place to solve for them. In particular, using an actor-critic network, V_i and μ_{i+1} can be approximated as

$$\begin{aligned} \hat{V}_i(x) &= w_i^{\mathrm{T}}\phi(x), \\ \hat{\mu}_{i+1}(x) &= v_i^{\mathrm{T}}\psi(x), \end{aligned} \tag{11}$$

where $w_i \in \mathbb{R}^{N_c}, v_i \in \mathbb{R}^{N_a}$ are neural network weights, and $\phi(x) \in \mathbb{R}^{N_c}, \psi(x) \in \mathbb{R}^{N_a}$ are basis functions. The weights

Algorithm 2 Learning-based Policy Iteration

1: Begin with $\mu_0 \in \Psi(\Omega)$, $\epsilon > 0$, $i = 0$.
2: **repeat**
3: Compute Θ_i as in (15), and let $i = i + 1$.
4: **until** $i \geqslant 2$ & $\|\Theta_{i-1} - \Theta_{i-2}\| < \epsilon$.
5: Obtain v_i from Θ_i and set

$$u_L = v_i^{\mathrm{T}}\psi(x). \tag{16}$$

w_i and v_i must then be trained to force (10) to hold as closely as possible. To this end, note that the left-most term in (10) can be approximated as

$$\hat{V}_i(x(t'_k)) - \hat{V}_i(x(t_k)) = (\phi(x(t'_k)) - \phi(x(t_k)))^{\mathrm{T}} w_i, \tag{12}$$

and the first term inside the integral in (10) as

$$2r\hat{\mu}_{i+1}(x)(u - \hat{\mu}_i(x)) = 2r(u - \hat{\mu}_i(x))\psi^{\mathrm{T}}(x)v_i. \tag{13}$$

Hence, the overall error by approximating (10) with (11):

$$\begin{aligned} e_{i,k} &= \hat{V}_i(x(t'_k)) - \hat{V}_i(x(t_k)) \\ &\quad + \int_{t_k}^{t'_k} \Big(2r\hat{\mu}_{i+1}(x)(u - \hat{\mu}_i(x)) + Q(x) + r\hat{\mu}_i^2(x)\Big)\mathrm{d}\tau, \end{aligned}$$

may be written in compact form, using (12)-(13), as

$$e_{i,k} = \Psi_{i,k}\Theta_i + \Phi_{i,k} \tag{14}$$

where $\Theta_i = \begin{bmatrix} w_i^{\mathrm{T}} & v_i^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$, $\Phi_{i,k} = \int_{t_k}^{t'_k}(Q(x) + r\hat{\mu}_i^2(x))\mathrm{d}\tau$, and $\Psi_{i,k} = \begin{bmatrix}(\phi(x(t'_k)) - \phi(x(t_k)))^{\mathrm{T}} & \int_{t_k}^{t'_k} 2r(u - \hat{\mu}_i(x))\psi^{\mathrm{T}}(x)\mathrm{d}\tau\end{bmatrix}$. Defining then the matrices

$$\begin{aligned} \Psi_i &= \begin{bmatrix}\Psi_{i,1}^{\mathrm{T}} & \dots & \Psi_{i,K}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}, \\ \Phi_i &= \begin{bmatrix}\Phi_{i,1} & \dots & \Phi_{i,K}\end{bmatrix}^{\mathrm{T}}, \end{aligned}$$

the weights Θ_i, for all $i \in \mathbb{N}$ can be trained using a least-sum-of-squares procedure on the error (14), according to

$$\Theta_i = -(\Psi_i^{\mathrm{T}}\Psi_i)^{-1}\Psi_i^{\mathrm{T}}\Phi_i. \tag{15}$$

Finally, using (15), one obtains Algorithm 2, which is a learning-based reformulation of Algorithm 1.

According to [4], if the number of bases N_c, N_a is large enough, then the control u_L obtained by Algorithm 2 uniformly approximates the optimal control $\mu^\star$ over Ω. However, this is subject to an excitation condition on the measured data, which is needed to guarantee the inversion in (15) is well-defined. This condition is stated as follows.

Assumption 2. There exist constants $\eta > 0$ and $K_0 \in \mathbb{N}$, such that if $K \geqslant K_0$ then $\frac{1}{K}\lambda_{\min}(\Psi_i^{\mathrm{T}}\Psi_i) \geqslant \eta$, $\forall i \in \mathbb{N}$. □

The condition in Assumption 2 is generally difficult to verify a priori over an infinite horizon on $i \in \mathbb{N}$. However, in the following Theorem we provide – for the first time – a way to easily verify that it holds for all $i \in \mathbb{N}$.

Theorem 1. *Let*

$$\bar{\Psi}_k = \begin{bmatrix} \phi(x(t'_k)) - \phi(x(t_k)) \\ \int_{t_k}^{t'_k} 2ru\psi(x(\tau))\mathrm{d}\tau \\ \int_{t_k}^{t'_k} -2r\psi(x(\tau)) \otimes_h \psi(x(\tau))\mathrm{d}\tau \end{bmatrix}$$

and $\bar{\Psi} = [\bar{\Psi}_1 \ \dots \ \bar{\Psi}_K]^{\mathrm{T}}$. *If* $\frac{1}{K}\lambda_{\min}(\bar{\Psi}^{\mathrm{T}}\bar{\Psi}) \geqslant \eta$ *and* $\frac{1}{K}\lambda_{\min}(\Psi_0^{\mathrm{T}}\Psi_0) \geqslant \eta$, *then* $\frac{1}{K}\lambda_{\min}\left(\Psi_i^{\mathrm{T}}\Psi_i\right) \geqslant \eta$ *for all* $i \in \mathbb{N}$.

Proof. Note that for $i \in \mathbb{N}\backslash\{0\}$ one has $\Psi_{i,k} = \tilde{\Psi}_k^{\mathrm{T}} W_i$, where

$$W_i = \begin{bmatrix} I_{N_c} & 0 \\ 0 & I_{N_a} \\ 0 & w_{i-1} \otimes I_{N_a} \end{bmatrix},$$

$$\tilde{\Psi}_k = \begin{bmatrix} \phi(x(t'_k)) - \phi(x(t_k)) \\ \int_{t_k}^{t'_k} 2ru\psi(x(\tau))\mathrm{d}\tau \\ \int_{t_k}^{t'_k} -2r\psi(x(\tau)) \otimes \psi(x(\tau))\mathrm{d}\tau \end{bmatrix}$$

and the zeros indicate null matrices of appropriate dimensions. In addition, there exists a linear transformation $A \in \mathbb{R}^{N_a^2 \times (N_a+1)N_a/2}$ – a duplication matrix – such that $\psi(\cdot) \otimes \psi(\cdot) = A(\psi(\cdot) \otimes_h \psi(\cdot))$. Hence, denoting

$$B = \begin{bmatrix} I_{N_c} & 0 & 0 \\ 0 & I_{N_a} & 0 \\ 0 & 0 & A \end{bmatrix}$$

we obtain $\tilde{\Psi}_k = B\bar{\Psi}_k$, and thus $\Psi_{i,k} = \bar{\Psi}_k^{\mathrm{T}} B^{\mathrm{T}} W_i$. Therefore,

$$\Psi_i^{\mathrm{T}}\Psi_i = \sum_{k=1}^{K} \Psi_{i,k}^{\mathrm{T}}\Psi_{i,k} = \sum_{k=1}^{K} (\bar{\Psi}_k^{\mathrm{T}} B^{\mathrm{T}} W_i)^{\mathrm{T}} \bar{\Psi}_k^{\mathrm{T}} B^{\mathrm{T}} W_i$$
$$= \sum_{k=1}^{K} W_i^{\mathrm{T}} B\bar{\Psi}_k \bar{\Psi}_k^{\mathrm{T}} B^{\mathrm{T}} W_i = W_i^{\mathrm{T}} B\bar{\Psi}^{\mathrm{T}}\bar{\Psi} B^{\mathrm{T}} W_i. \quad (17)$$

If $\frac{1}{K}\lambda_{\min}(\bar{\Psi}^{\mathrm{T}}\bar{\Psi}) \geqslant \eta$, then equation (17) implies:

$$\Psi_i^{\mathrm{T}}\Psi_i \geq K\eta W_i^{\mathrm{T}} B B^{\mathrm{T}} W_i. \quad (18)$$

In addition,

$$W_i^{\mathrm{T}} B B^{\mathrm{T}} W_i =$$
$$\begin{bmatrix} I_{N_c} & 0 & 0 \\ 0 & I_{N_a} & (w_{i-1}\otimes I_{N_a})^{\mathrm{T}} A \end{bmatrix} \begin{bmatrix} I_{N_c} & 0 \\ 0 & I_{N_a} \\ 0 & A^{\mathrm{T}}(w_{i-1}\otimes I_{N_a}) \end{bmatrix} \quad (19)$$
$$= \begin{bmatrix} I_{N_c} & 0 \\ 0 & I_{N_a} + (w_{i-1} \otimes I_{N_a})^{\mathrm{T}} A A^{\mathrm{T}} (w_{i-1} \otimes I_{N_a}) \end{bmatrix} \geq I_{N_c+N_a}.$$

Combining (18) and (19) we obtain $\Psi_i^{\mathrm{T}}\Psi_i \geq K\eta I_{N_c+N_a}$, hence $\lambda_{\min}(\Psi_i^{\mathrm{T}}\Psi_i) \geqslant K\eta$ for all $i \in \mathbb{N}\backslash\{0\}$. Using also the condition $\frac{1}{K}\lambda_{\min}(\Psi_0^{\mathrm{T}}\Psi_0) \geqslant \eta$, the result follows. ■

Remark 1. Based on Theorem 1, to guarantee the inversion in (15) is feasible for all $i \in \mathbb{N}$, one needs to check that it is feasible for $i = 0$, and that $\frac{1}{K}\lambda_{\min}(\bar{\Psi}^{\mathrm{T}}\bar{\Psi}) \geqslant \eta$ holds. Hence, Theorem 1 provides – for the first time – a method to a priori guarantee the execution of learning-based PI (Algorithm 2) will not run into numerical issues as i increases. □

IV. The Safety Net Block

We now proceed to the design of the safety net block of Fig. 1. As explained in the problem formulation, this block must ensure that the final control u entering system (1) is smooth enough, keeps the closed-loop bounded, and is identical to the learning-based control u_L within the set Ω.

Towards designing this block, assume without loss of generality that Ω is a rectangular region, so that

$$\Omega = [-\omega_1, \ \omega_1] \times \ldots \times [-\omega_n, \omega_n] \quad (20)$$

for some strictly positive constants $\omega_1, \ldots, \omega_n$[1]. In addition, for any constants $\bar{\omega} > \omega > 0$, consider the following non-decreasing, continuously differentiable function over $(-1, 1)$, having dead zone in $[-\frac{\omega}{\bar{\omega}}, \frac{\omega}{\bar{\omega}}] \subset (-1, 1)$ and satisfying $\lim_{\xi\to\pm 1} T(\xi;\ \omega, \bar{\omega}) = \pm\infty$:

$$T(\xi;\ \omega, \bar{\omega}) = \begin{cases} (\xi - \frac{\omega}{\bar{\omega}})^2 \cdot \tan(\frac{\pi}{2}\xi), & \xi \in (\frac{\omega}{\bar{\omega}},\ 1), \\ 0, & \xi \in [-\frac{\omega}{\bar{\omega}}, \frac{\omega}{\bar{\omega}}], \\ (\xi + \frac{\omega}{\bar{\omega}})^2 \cdot \tan(\frac{\pi}{2}\xi), & \xi \in (-1, -\frac{\omega}{\bar{\omega}}). \end{cases} \quad (21)$$

The design of the safety net block is as follows.

Step $j = 1, \ldots, n$**:** Select $\bar{\omega}_j > \omega_j$, $k_j > 0$, $a_0 = 0$, and design the virtual control signals:

$$a_j = -k_j T(\xi_j;\ \omega_j,\ \bar{\omega}_j), \quad (22)$$
$$\xi_j = \frac{x_j - a_{j-1}}{\bar{\omega}_j}. \quad (23)$$

Step $n + 1$**:** Select the final control signal as

$$u = u_L + a_n. \quad (24)$$

Remark 2. Core to the safety net is the transformation (21), the design of which can be understood as follows: i) it has a deadzone around the origin, which will allow us to guarantee that the safety net does not interfere with the learning-based control within the set Ω; ii) it increases rapidly once the ξ-variable approaches 1, a quality needed to compensate for the unknown dynamics in (1) should $x(t)$ exit the set Ω; and iii) it is scaled by a quadratically vanishing function, to guarantee that it is continuously differentiable. □

V. Main Result

In this section, we show that the proposed control scheme of Fig. 1, described by (16), (20)-(24), achieves the four qualities required by the Problem Statement. Since items 1) and 4) of this statement can be verified by simple inspection, in what follows we focus on proving items 2) and 3).

Theorem 2. *Consider system* (1) *and Assumption 1. Suppose that* $x(0) \in \Omega$. *Then, the control scheme described by* (16), (20)-(24) *guarantees that*

1) *the closed loop operates approximately optimally within* Ω, *i.e.,* $u(t) = u_L(t)$ *when* $x(t) \in \Omega$.
2) *the closed loop remains bounded when* $x(t) \notin \Omega$.

Proof. To prove item 1, suppose that $x(t) \in \Omega$. Then, $x_1(t) \in [-\omega_1, \omega_1]$, and thus $\xi_1(t) \in [-\frac{\omega_1}{\bar{\omega}_1}, \frac{\omega_1}{\bar{\omega}_1}]$. By the definitions (21), (22), this implies that $a_1(t) = 0$. Next, $x(t) \in \Omega$ also implies that $x_2(t) \in [-\omega_2, \omega_2]$, thus $\xi_2(t) \in [-\frac{\omega_2}{\bar{\omega}_2}, \frac{\omega_2}{\bar{\omega}_2}]$ since $a_1(t) = 0$. By the definitions (21), (22), this implies that $a_2(t) = 0$ and, following this procedure recursively, it follows that $a_n(t) = 0$. Given this, according to (24), the control law applied to the system at time t is $u(t) = u_L(t)$, i.e., the closed-loop operates approximately optimally.

To prove item 2, note that for $j = 1, \ldots, n-1$, (23) yields

$$x_j = \bar{\omega}_j \xi_j + a_{j-1}. \quad (25)$$

[1]If Ω is not rectangular, it can be inner-approximated by a rectangle.

Using (25), the dynamics of ξ_j in (23), $j = 1, \ldots, n-1$, are:

$$\dot{\xi}_j = \frac{1}{\bar{\omega}_j}\left(f_j(\bar{x}_j)+\bar{\omega}_{j+1}g_j(\bar{x}_j)\xi_{j+1}+g_j(\bar{x}_j)a_j-\dot{a}_{j-1}\right), \quad (26)$$

$$\dot{\xi}_n = \frac{1}{\bar{\omega}_n}\left(f_n(\bar{x}_j) + g_n(\bar{x}_n)u_L + g_n(\bar{x}_n)a_n - \dot{a}_{n-1}\right). \quad (27)$$

The right-hand side in (25) depends solely on the variables ξ_j and ξ_{j-1} and, thus, so does x_j. Therefore, denoting $\xi := [\xi_1 \ \ldots \ \xi_n]$, system (26)-(27) can be written compactly as

$$\dot{\xi} = F(\xi) \quad (28)$$

where $F : (-1, 1) \to \mathbb{R}^n$ models the right-hand sides of (27). Since T is continuously differentiable on $(-1, 1)$, and since u_L as well as f_j, g_j, $j = 1, \ldots, n$, are locally Lipschitz, it follows that $F(\cdot)$ is also locally Lipschitz on $(-1, 1)^n$. Moreover, based on the reasoning of the previous paragraph, $x(0) \in \Omega$ implies $\xi(0) \in [-\frac{\omega_1}{\bar{\omega}_1}, \frac{\omega_1}{\bar{\omega}_1}] \times \ldots \times [-\frac{\omega_n}{\bar{\omega}_n}, \frac{\omega_n}{\bar{\omega}_n}] \subset (-1, 1)^n$. Hence, following standard arguments, system (28) yields a unique maximal solution $\xi(t) \in (-1, 1)^n$ for $t \in [0, \ \tau_{\max})$, where $\tau_{\max} \in (0, +\infty]$. Next, denote $\epsilon_j = T(\xi_j; \ \omega_j, \ \bar{\omega}_j)$ for all $j = 1, \ldots, n$. The rest of the proof follows a recursive procedure for all $t \in [0, \tau_{\max})$.

Step 1: Consider the energy function $V_1 = \frac{1}{2}\epsilon_1^2$. Using (26), (22) and the chain rule, its derivative is:

$$\begin{aligned}\dot{V}_1 &= \frac{\epsilon_1}{\bar{\omega}_1}\frac{\partial T(\xi_1)}{\partial \xi_1}\left(f_1(\bar{x}_1) + \bar{\omega}_2 g_1(\bar{x}_1)\xi_2 - k_1 g_1(\bar{x}_1)\epsilon_1 - \dot{a}_0\right) \\ &\leqslant \left|\frac{\epsilon_1}{\bar{\omega}_1}\frac{\partial T(\xi_1)}{\partial \xi_1}\right|\left(M_1(t) - k_1 \underline{g}_1|\epsilon_1|\right), \quad (29)\end{aligned}$$

where $M_1(t) = |f_1(\bar{x}_1) + \bar{\omega}_2 g_1(\bar{x}_1)\xi_2 - \dot{a}_0|$, and where the non-decreasing property of T was used. Note that by (25), $|x_1| \leqslant \bar{\omega}_1|\xi_j| + |a_0| \leqslant \bar{\omega}_1$ while $|\xi_2| \leqslant 1$ and $\dot{a}_0 = 0$. Therefore, by the continuity of f_1 and g_1, the extreme value theorem implies the existence of a positive constant $\bar{M}_1 < \infty$ such that $|M_1(t)| \leqslant \bar{M}_1$ for all $t \in [0, \tau_{\max})$. Hence, (29) yields $|\epsilon_1(t)| = |T(\xi_1; \ \omega_1, \ \bar{\omega}_1)| \leqslant \max\{|\epsilon_1(0)|, \frac{\bar{M}_1}{k_1 \underline{g}_1}\} =: \bar{\epsilon}_1$ for all $t \in [0, \tau_{\max})$. Using this, if we assume that $\limsup_{t\to\tau_{\max}} |\xi_1(t)| = 1$, by the continuity and monotonicity of T this would imply that $\limsup_{t\to\tau_{\max}} |\epsilon_1(t)| = \infty$, which is contradicting. Hence $\limsup_{t\to\tau_{\max}} |\xi_1(t)| < 1$, hence there exists $\bar{\xi}_1 \in (-1, 1)$ such that $\xi_1(t) \in [-\bar{\xi}_1, \bar{\xi}_1]$ for all $t \in [0, \tau_{\max})$. Finally, since $\epsilon_1(t)$ remains bounded, the existence of finite positive constants $\bar{a}_1, \bar{\dot{a}}_1$ such that $|a_1| \leqslant \bar{a}_1$ and $|\dot{a}_1| \leqslant \bar{\dot{a}}_1$ also follows from (22) and (26).

Step $j = 2, \ldots, n-1$**:** Defining the energy function $V_j = \frac{1}{2}\epsilon_j^2$ and using the boundedness of $\bar{a}_1$ and $\bar{\dot{a}}_1$ established from Step 1, the existence of finite positive constants $\bar{\epsilon}_j, \bar{a}_j, \bar{\dot{a}}_j$ such that $|\epsilon_j| \leqslant \bar{\epsilon}_j$, $|a_j| \leqslant \bar{a}_j$ and $|\dot{a}_j| \leqslant \bar{\dot{a}}_j$ also follows recursively for all $t \in [0, \tau_{\max})$. Similarly, the existence of a constant $\bar{\xi}_j \in (-1, 1)$ such that $\xi_j \in [-\bar{\xi}_j, \bar{\xi}_j]$ also follows.

Step n**:** Consider the energy function $V_n = \frac{1}{2}\epsilon_n^2$. Using (27), (24), (22) and the chain rule, its derivative is:

$$\begin{aligned}\dot{V}_n &= \frac{\epsilon_n}{\bar{\omega}_n}\frac{\partial T(\xi_n)}{\partial \xi_n}\left(f_n(\bar{x}_n)+g_n(\bar{x}_n)u_L - k_n g_n(\bar{x}_n)\epsilon_n - \dot{a}_{n-1}\right) \\ &\leqslant \left|\frac{\epsilon_n}{\bar{\omega}_n}\frac{\partial T(\xi_n)}{\partial \xi_n}\right|\left(M_n(t) - k_n \underline{g}_n|\epsilon_n|\right), \quad (30)\end{aligned}$$

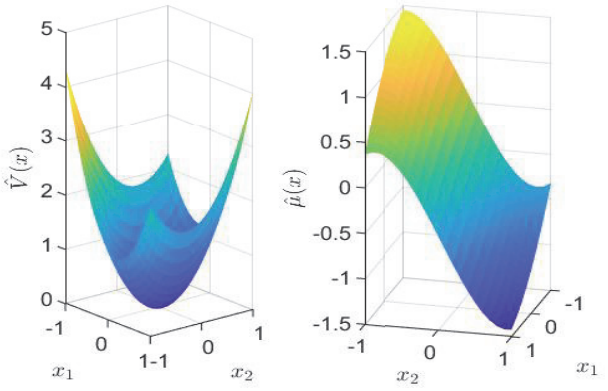

Fig. 2. The approximate value function $\hat{V}$ and optimal control $\hat{\mu}$ provided by the actor-critic network over Ω.

where $M_n(t) = |f_n(\bar{x}_n)+g_n(\bar{x}_n)u_L - \dot{a}_{n-1}|$, and where the non-decreasing property of T was used. Note that by (25), $|x_n| \leqslant \bar{\omega}_n|\xi_n| + |a_{n-1}| \leqslant \bar{\omega}_1 + \bar{a}_{n-1}$ while $|\dot{a}_{n-1}| \leqslant \bar{\dot{a}}_{n-1}$ from the previous step. Therefore, by the continuity of f_n, g_n and u_L, application of the extreme value theorem implies the existence of a positive constant $\bar{M}_n < \infty$ such that $|M_n(t)| \leqslant \bar{M}_n$ for all $t \in [0, \tau_{\max})$. Hence, (30) yields $|\epsilon_n(t)| = |T(\xi_n; \ \omega_n, \ \bar{\omega}_n)| \leqslant \max\{|\epsilon_n(0)|, \frac{\bar{M}_n}{k_n \underline{g}_n}\} =: \bar{\epsilon}_n$ for all $t \in [0, \tau_{\max})$. Using this, if we assume that $\limsup_{t\to\tau_{\max}} |\xi_n(t)| = 1$, by the continuity and monotonicity of T this would imply that $\limsup_{t\to\tau_{\max}} |\epsilon_n(t)| = \infty$, which is contradicting. Hence $\limsup_{t\to\tau_{\max}} |\xi_n(t)| < 1$, hence there exists $\bar{\xi}_n \in (-1, 1)$ such that $\xi_n(t) \in [-\bar{\xi}_n, \bar{\xi}_n]$ for all $t \in [0, \tau_{\max})$. Finally, since $\epsilon_n(t)$ remains bounded, the existence of finite positive constants $\bar{a}_n, \bar{\dot{a}}_n$ such that $|a_n| \leqslant \bar{a}_n$ and $|\dot{a}_n| \leqslant \bar{\dot{a}}_1$ also follows from (22) and (27). Accordingly, $u(t)$ remains bounded by the boundedness of a_n, x_n, and the continuity of u_L.

To conclude, note that by the previous steps, $\xi(t)$ remains in the compact set $[-\bar{\xi}_1, \bar{\xi}_1] \times \ldots \times [-\bar{\xi}_n, \bar{\xi}_n] \subset (-1, 1)^n$ for all $t \in [0, \ \tau_{max})$. Since $F(\cdot)$ is locally Lipschitz on $(-1, 1)^n$, it follows that $\tau_{max} = \infty$ [[16], Theorem 3.3]. ■

VI. Simulations

We perform simulations on the Van der Pol oscillator [17]:

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = -x_1 + 0.5x_2 - x_2^3 + u. \quad (31)$$

The objective is to approximate the controller that minimizes the cost (3) with $Q(x) = \|x\|^2$, $r = 1$, while having guarantees of closed-loop boundedness and robustness. To this end, an actor-critic network is employed with basis functions of monomials up to the third order, and whose weights are trained by gathering input and state data over $\Omega = [-1, 1] \times [-1, 1]$ and performing Algorithm 2. The safety net of Section IV is also concurrently employed, with parameters $\omega_1 = \omega_2 = 1$, $\bar{\omega}_1 = \bar{\omega}_2 = 1.25$ and $k_1 = k_2 = 1$.

The approximated optimal value function and optimal control provided by the actor-critic network over Ω are depicted in Fig. 2. One may notice that their values are reasonable, since the former is positive definite while the latter has a

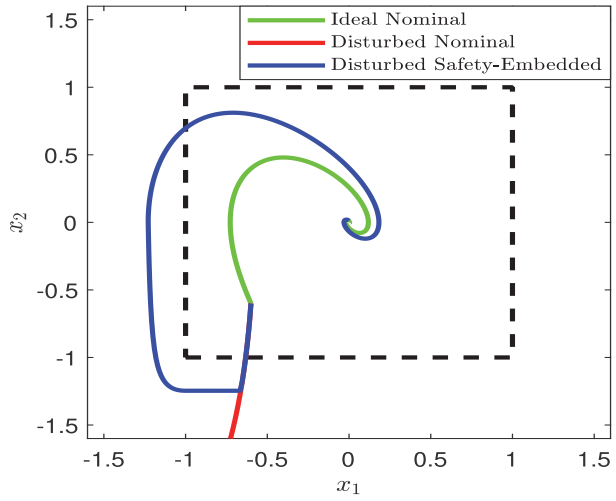

Fig. 3. The phase portrait of the closed loop in the three tested scenarios. The dashed lines indicate the boundary of Ω.

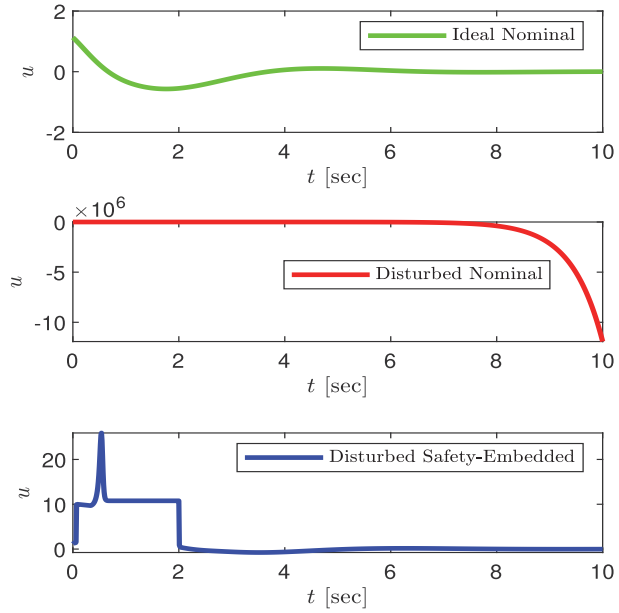

Fig. 4. The control input trajectories in the three tested scenarios.

stabilizing control direction throughout the state space. To showcase the efficiency of the proposed combined control scheme, the approximated optimal control is subsequently tested with $x_0 = [-0.6 \ \ -0.6]^{\mathrm{T}}$ in three cases: first, in the case where $u = u_L$ in (31) and where no uncertainty enters the system; second, in the case where $u = u_L$, but where an additive disturbance $d(t) = 12$ is temporarily present in the control input over $t \in [0, 2]$, and vanishes thereafter; and finally, in the case where u is given by the combined scheme (24), and where the same additive disturbance is present.

The phase portraits and control inputs for all three cases are depicted in Fig. 3-4. It can be noticed that while the standalone learning-based control works quite well in the first, ideal scenario, it fails to keep the closed-loop bounded in the disturbed case. This is because the temporary additive disturbance manages to drive the state trajectory outside the set Ω over which the approximate optimal control is valid, and this invalid controller ends up driving the closed loop to instability. On the other hand, in the case where the approximate optimal control is applied in tandem with the safety net, the state trajectories are unable to escape to infinity, and are in fact constrained to remain close to Ω when the disturbance is active. In addition, once the disturbance vanishes, the state trajectory quickly returns into Ω, where it is driven to the origin by the approximate optimal controller.

VII. Conclusion

We proposed a safety net, which enhances and ensures the robustness of approximate optimal controllers derived by off-policy reinforcement learning, with hard guarantees. Future work includes an extension to multi-agent systems.

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] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[4] Y. Jiang and Z.-P. Jiang, "Robust adaptive dynamic programming and feedback stabilization of nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 882–893, 2014.

[5] H. Modares, F. L. Lewis, and M.-B. Naghibi-Sistani, "Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems," *Automatica*, vol. 50, no. 1, pp. 193–202, 2014.

[6] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, 2009.

[7] B. Kiumarsi, F. L. Lewis, and D. S. Levine, "Optimal control of nonlinear discrete time-varying systems using a new neural network approximation structure," *Neurocomputing*, vol. 156, pp. 157–165, 2015.

[8] 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.

[9] W. Gao and Z.-P. Jiang, "Learning-based adaptive optimal tracking control of strict-feedback nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2614–2624, 2017.

[10] A. Kanellopoulos, F. Fotiadis, C. Sun, Z. Xu, K. G. Vamvoudakis, U. Topcu, and W. E. Dixon, "Temporal-logic-based intermittent, optimal, and safe continuous-time learning for trajectory tracking," in *Proc. IEEE 60th Conf. Decis. Control*, pp. 1263–1268, 2021.

[11] M. H. Cohen and C. Belta, "Approximate optimal control for safety-critical systems with control barrier functions," in *Proc. IEEE 59th Conf. Decis. Control*, pp. 2062–2067, 2020.

[12] B. Fan, Q. Yang, X. Tang, and Y. Sun, "Robust adp design for continuous-time nonlinear systems with output constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2127–2138, 2018.

[13] Y. Yang, Y. Yin, W. He, K. G. Vamvoudakis, H. Modares, and D. C. Wunsch, "Safety-aware reinforcement learning framework with an actor-critic-barrier structure," in *Proc. IEEE Amer. Control Conf.*, pp. 2352–2358, 2019.

[14] P. Rousseas, C. P. Bechlioulis, and K. J. Kyriakopoulos, "A continuous off-policy reinforcement learning scheme for optimal motion planning in simply-connected workspaces," in *Proc. IEEE Int. Conf. Robot. Automat.*, pp. 10247–10253, 2023.

[15] 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.

[16] H. K. Khalil, *Nonlinear systems*. Upper Saddle River, N.J.: Prentice Hall, 2002.

[17] C. I. Byrnes, F. D. Priscoli, A. Isidori, and W. Kang, "Structurally stable output regulation of nonlinear systems," *Automatica*, vol. 33, no. 3, pp. 369–385, 1997.