Hamiltonian-Driven Adaptive Dynamic Programming With Approximation Errors

Yongliang Yang, *Member, IEEE*, Hamidreza Modares, *Senior Member, IEEE*, Kyriakos G. Vamvoudakis, *Senior Member, IEEE*, Wei He, *Senior Member, IEEE*, Cheng-Zhong Xu, *Fellow, IEEE*, and Donald C. Wunsch, *Fellow, IEEE*

***Abstract*—In this article, we consider an iterative adaptive dynamic programming (ADP) algorithm within the Hamiltonian-driven framework to solve the Hamilton–Jacobi–Bellman (HJB) equation for the infinite-horizon optimal control problem in continuous time for nonlinear systems. First, a novel function, "min-Hamiltonian," is defined to capture the fundamental properties of the classical Hamiltonian. It is shown that both the HJB equation and the policy iteration (PI) algorithm can be formulated in terms of the min-Hamiltonian within the Hamiltonian-driven framework. Moreover, we develop an iterative ADP algorithm that takes into consideration the approximation errors during the policy evaluation step. We then derive a sufficient condition on the iterative value gradient to guarantee closed-loop stability of the equilibrium point as well as convergence to the optimal value. A model-free extension based on an off-policy reinforcement learning (RL) technique is also provided. Finally, numerical results illustrate the efficacy of the proposed framework.**

***Index Terms*—Hamilton-Jacobi-Bellman (HJB) equation, Hamiltonian-driven framework, inexact adaptive dynamic programming (ADP), optimal control.**

Manuscript received March 31, 2021; accepted August 20, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB2102100; in part by the National Natural Science Foundation of China under Grant 61903028; in part by the Science and Technology Development Fund, Macao, under Grant 0015/2019/AKP; in part by the UM Macao Talent Programme under Grant UMMTP-2019-02; in part by the Guangdong–Hong Kong–Macao Joint Laboratory of Human-Machine Intelligence-Synergy Systems under Grant 2019B121205007; in part by the National Science Foundation under Grant CPS1851588 and Grant S&AS1849198; in part by the Mary K. Finley Missouri Endowment; in part by the Missouri S&T Intelligent Systems Center; in part by the Lifelong Learning Machines Program from DARPA/Microsystems Technology Office; and in part by the Army Research Laboratory under Agreement W911NF-18-2-0260. This article was recommended by Associate Editor A. Katriniok. *(Corresponding author: Cheng-Zhong Xu.)*

Yongliang Yang and Wei He are with the School of Automation and Electrical Engineering and the Institute of Artificial Intelligence, University of Science and Technology Beijing, Beijing 100083, China (e-mail: yangyongliang@ieee.org; weihe@ieee.org).

Hamidreza Modares is with the Mechanical Engineering Department, Michigan State University, East Lansing, MI 48824 USA (e-mail: modaresh@msu.edu).

Kyriakos G. Vamvoudakis is with the Daniel Guggenheim School of Aerospace Engineering, Georgia Tech, Atlanta, GA 30332 USA (e-mail: kyriakos@gatech.edu).

Cheng-Zhong Xu is with the State Key Laboratory of Internet of Things for Smart City, Faculty of Science and Technology, University of Macau, Macau, China (e-mail: czxu@um.edu.mo).

Donald C. Wunsch is with the Department of Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, MO 65401 USA (e-mail: dwunsch@mst.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2021.3108034.

Digital Object Identifier 10.1109/TCYB.2021.3108034

I. Introduction

OPTIMIZATION-BASED control focuses on finding optimal policies for dynamical systems by optimizing user-defined performances, which capture desired objectives [1]–[5]. Pontryagin's minimum principle, which provides a necessary condition for optimality [6], requires the solution of a two-point boundary value problem in an open-loop sense. On the other hand, Bellman's principle of optimality [7] provides a necessary and sufficient condition in a closed-loop sense that is found by solving an algebraic Riccati equation (ARE) for linear systems and a Hamilton–Jacobi–Bellman (HJB) equation for nonlinear systems. But there is generally no analytical solution for nonlinear dynamical systems and HJB. Toward that, efficient approximation algorithms for solving the HJB equations have been widely presented in the literature. Specifically, approximate/adaptive dynamic programming (ADP) [8] and neuroadaptive dynamic programming (NDP) [9] are developed to solve the HJB equation forward-in-time by using function approximators [10]. Since then, different structures for ADP have been proposed, including action-dependent globalized dual heuristic programming [11], single critic learning [12], goal representation learning [13], model-free heuristic dynamic programming [14], model-free dual heuristic dynamic programming [15], and intermittent ADP with periodic and aperiodic event generators [16]–[18].

Related Works: Integral reinforcement learning (RL) has been developed to solve the optimal regulation problem [19], the optimal tracking problem [20], and differential games [21] without requiring the drift dynamics of the system. The actor–critic-based ADP method, where both the policy evaluation and policy improvement are performed in an online adaptive fashion, has been further developed [22]–[25]. In order to obviate the requirement of the exact knowledge of the system dynamics, identifiers have been incorporated into the actor–critic structure in [26]. However, the control performance is affected by the identification accuracy. In order to avoid the identification step, off-policy RL algorithms directly learn the value function and have been successfully applied to optimal regulation [27]–[30]; optimal tracking [31]; differential games [32]–[36]; stochastic systems [37], [38]; and robust stabilization [39]–[41]. In the aforementioned approaches, one requires the control policy to be evaluated (in the policy evaluation step) and updated (in the policy improvement step)

precisely at each iteration assuming that there is no approximation error, which is not always feasible. It is thus necessary to investigate the effect of the approximation errors on the convergence ADP algorithms, such as the policy iteration (PI) [42], value iteration [43]–[47], and optimistic PI algorithms [48]. Motivated by this shortcoming, in this work, we refer to the class of iterative ADP algorithms with approximation errors in the policy evaluation step as inexact ADP. We first develop a novel inexact ADP method and investigate the condition that guarantees convergence and closed-loop stability. Then, a model-free extension of the inexact ADP method is further designed to obviate the requirement of complete system dynamics. Therefore, the adverse effects of system identification errors on the system stability and control performance can be mitigated.

Contributions: To guarantee safety and performance even in the presence of approximation errors, it is of vital importance to analyze the adverse effects that they can have on the closed-loop stability of the equilibrium point and to the convergence to the optimal solution. The contribution of the present article is four-fold. First, a novel function, referred to as "min-Hamiltonian," is defined to unify the HJB equation and the PI algorithm. In this way, a quasi-Newton method can be applied to iteratively solve the HJB equation by using the min-Hamiltonian. Second, we investigate the dependency of the closed-loop stability and performance guarantee on the approximation residual resulting from inexact policy evaluation. On this basis, we further derive a sufficient condition that guarantees both the convergence of the iterative learning algorithm and the closed-loop stability with the iterative learning policy. Therefore, the inexact method developed in this article is robust against the bounded approximation error in terms of both stability guarantee and performance improvement. In addition, it is shown that for linear systems, the iterative ADP algorithm reduces to the Newton–Kleinman iteration [49].

Structure: The remainder of this article is structured as follows. Section II provides the problem formulation. The Hamiltonian-driven framework of the exact ADP is briefly reviewed in Section III. The extension of exact Hamiltonian-driven ADP to the inexact case is shown in Section IV. Case studies for both linear and nonlinear dynamical systems are presented in Section VI. Section VII provides the conclusion and suggests future research directions.

Notations: In this article, we denote $\mathbb{R}$ and $\mathbb{R}^+$ as the set of reals and non-negative reals, respectively. $\mathbb{Z}^+$ denotes the set of non-negative integers. $\|\cdot\|$ denotes the Euclidean norm for vectors or the induced matrix norm for matrices. A real-valued function f defined on the compact set Ω containing the origin is said to be of class $C^k(\Omega)$ with $k \in \mathbb{Z}^+$ if f is continuous when $k = 0$, or f is k-times continuously differentiable on Ω when $k \geq 1$. A function $f : \Omega \to \mathbb{R}+$ defined on compact set Ω containing the origin is said to be positive definite (semidefinite) if $f(x) > 0 (\geq 0)$ for all $x \in \Omega \backslash \{0\}$, and $f(0) = 0$. In addition, the function f is called proper if $\Omega = \mathbb{R}^n$ and $\lim_{\|x\| \to \infty} f(x) = \infty$.

II. Problem Formulation

Consider the following continuous-time nonlinear system:

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

with states $x \in \mathbb{R}^n$, $f(x) \in \mathbb{R}^n$, and $g(x) \in \mathbb{R}^{n \times m}$ and control input $u \in \mathbb{R}^m$. We assume that $f(0) = 0$, $f(x) + g(x)u$ is Lipschitz continuous on a compact set $\Omega_x \subset \mathbb{R}^n$ that contains the origin, $g(x)$ is bounded on Ω_x, and system (1) is stabilizable on Ω_x.

We consider the cost functional as

$$J(u; x_0) = \int_{t_0}^{\infty} L(x(t), u(t)) \mathrm{d}t \tag{2}$$

with

$$L(x, u) = Q(x) + \|u\|_R^2 \tag{3}$$

for all $(x, u) \in \mathbb{R}^n \times \mathbb{R}^m$, where $Q(x)$ is a positive-definite function, $\|u\|_R^2 = u^T R u$, and R is a positive-definite symmetric matrix. In addition, the cost (2) is assumed to be zero-state observable [50].

Problem 1: Find a control policy, if possible, $u^\star(x)$ and its associated value function $V^\star(x)$ such that

$$V^\star(x_0) = \inf_u J(u; x_0), \quad u^\star = \arg\inf_u J(u; x_0). \tag{4}$$

Next, for system (1), the Hamiltonian is defined as

$$H(x, p, u) = p^T \big[f(x) + g(x)u(x) \big] + L(x, u) \tag{5}$$

for all $(x, p, u) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m$. By following [29], the following assumption is made.

Assumption 1: Problem 1 is solvable in the sense that there exist a positive-definite, continuously differentiable, and proper function $V^\star(x)$ and a unique continuous function $\alpha : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^m$ such that:

1) $\alpha(x, p) = \arg\inf_{u \in \mathbb{R}^m} H(x, p, u)$;
2) the infinite-horizon HJB equation

$$\inf_{u \in \mathbb{R}^m} H\left(x, \frac{\partial V^\star(x)}{\partial x}, u\right) = 0,\ V^\star(0) = 0 \tag{6}$$

is satisfied.

As investigated in [5], Assumption 1 implies that $V^\star(x)$ is the unique solution to (6) provided that $u^\star(x) = \alpha(x, [(\partial V^\star(x))/(\partial x)])$ exists.

The cost (2) is a functional of the policy $u(\cdot)$ evaluated along the state trajectory $x(t)$, which also depends on the initial state x_0. However, there is no analytical solution to Problem 1 for general nonlinear systems. Therefore, an alternative to the cost functional independent of the solution is needed. In the following, an alternative evaluation of an arbitrary state-space trajectory, given a policy with the prescribed property, is introduced.

Definition 1 (Admissible Control [51]): A mapping $u : \Omega_x \to \mathbb{R}^m$ is said to be an admissible control policy, denoted as $u(x) \in \Psi(\Omega_x)$, with respect to the cost functional (2) on $\mathbb{R}^n$, if the following conditions are satisfied:

1) $u(x)$ is continuous on $\mathbb{R}^n$;
2) $u(0) = 0$;
3) $u(x)$ stabilizes (1) on $\mathbb{R}^n$;
4) the cost functional is finite $\forall x_0 \in \mathbb{R}^n$.

For a given admissible policy $u(x) \in \Psi(\Omega_x)$, the function $V(x) : \Omega_x \to \mathbb{R}$ is called a value function if

$$\begin{cases} H\left(x, \frac{\partial V(x)}{\partial x}, u\right) = 0 \\ V(0) = 0 \end{cases} \tag{7}$$

holds for all $x \in \Omega_x$ [29], [52]. The positive definiteness of the value function $V(x)$ that satisfies the generalized HJB (GHJB) equation (7) can be guaranteed by the zero-state observability of the cost $J(u; x_0)$ (2) [53]. In keeping with other works in [22] and [29], it is assumed that the value function $V(x)$ that satisfies the GHJB equation (7) exists in the space $C^1(\Omega_x)$.

In contrast to the cost functional (2), the value function $V(x)$ is an evaluation of an arbitrary state x in the state space. The value function of a given state $V(x)$ is equivalent to the cost functional $J(x; u)$ starting from a predetermined state x with a given admissible policy u, as discussed in [4] and [5]. The value function $V(x)$ that satisfies (7) is also referred to as the GHJB equation [54].

From the discussion above, it is shown that the Hamiltonian plays an important role in evaluating an arbitrary admissible policy, which is not necessarily optimal.

Similar to the Hamiltonian $H(x, p, u)$ defined in (5), we define h as

$$h(x, p) = -\frac{1}{4}p^T g(x)R^{-1}g^T(x)p + p^T f(x) + Q(x) \tag{8}$$

for all $(x, p) \in \mathbb{R}^n \times \mathbb{R}^n$, where the functions $\{f(x), g(x)\}$ and $\{Q(x), R\}$ are defined as in (1) and (3), respectively.

Remark 1: From (8), for $V : \Omega_x \subset \mathbb{R}^n \to \mathbb{R}$, one has

$$\begin{aligned}\left\|h\left(x, \frac{\partial V(x)}{\partial x}\right)\right\| &= \left\|L(x, u) + \left[\frac{\partial V(x)}{\partial x}\right]^T \left[f(x) + g(x)u\right]\right\| \\ &\le \frac{1}{4}\left\|\frac{\partial V(x)}{\partial x}\right\| \cdot \|g(x)\| \cdot \left\|R^{-1}\right\| \cdot \left\|g^T(x)\right\| \\ &\quad \cdot \left\|\frac{\partial V(x)}{\partial x}\right\| \\ &\quad + \|Q(x)\| + \left\|\frac{\partial V(x)}{\partial x}\right\| \cdot \|f(x) + g(x)u\|\end{aligned}$$

with $u = -(1/2)R^{-1}g^T(x)[(\partial V(x))/(\partial x)]$. In addition, consider the fact that $g(x)$ is bounded on Ω_x and $f(\cdot) + g(\cdot)u$ is Lipschitz continuous on Ω_x. Then, $\|f(x) + g(x)u\|$ is also bounded on Ω_x because the Lipschitz continuous functions are guaranteed to be bounded on compact sets [55], provided that $[(\partial V(x))/(\partial x)]$ is bounded on Ω_x. Therefore, given admissible policy $u \in \Psi(\Omega_x)$, it is assumed that the solution $V(x)$ to the GHJB equation (7) is in the space $C^1(\Omega_x)$ [22], [29]. Finally, the boundedness of $h(x, [(\partial V(x))/(\partial x)])$ can be guaranteed for all $x \in \Omega_x$.

Considering system (1) with a utility function (3) and a cost functional (2), via completion of squares, the Hamiltonian can be equivalently expressed $\forall x$ as

$$\begin{aligned}H\left(x, \frac{\partial V(x)}{\partial x}, u\right) &= Q(x) + \left[\frac{\partial V(x)}{\partial x}\right]^T f(x) \\ &\quad + \left[u + \frac{1}{2}R^{-1}g^T\frac{\partial V(x)}{\partial x}\right]^T \\ &\quad \times R\left[u + \frac{1}{2}R^{-1}g^T\frac{\partial V(x)}{\partial x}\right] \\ &\quad - \frac{1}{4}\left[\frac{\partial V(x)}{\partial x}\right]^T g(x)R^{-1}g^T(x)\frac{\partial V(x)}{\partial x}.\end{aligned}$$

Therefore, the minimum of the Hamiltonian w.r.t. u can be equivalently formulated by $h(\cdot, \cdot)$ as

$$\begin{aligned}\inf_u H\left(x, \frac{\partial V(x)}{\partial x}, u\right) &= h\left(x, \frac{\partial V(x)}{\partial x}\right) \\ &= -\frac{1}{4}\left[\frac{\partial V}{\partial x}\right]^T g(x)R^{-1}g^T(x)\frac{\partial V}{\partial x} \\ &\quad + \left[\frac{\partial V}{\partial x}\right]^T f(x) + Q(x).\end{aligned}$$

In addition, we define

$$\bar{u}(x, p) = -\frac{1}{2}R^{-1}g^T(x)p \tag{9}$$

for all $(x, p) \in \mathbb{R}^n \times \mathbb{R}^n$. Then

$$\bar{u}\left(x, \frac{\partial V(x)}{\partial x}\right) = \arg\inf_u H\left(x, \frac{\partial V(x)}{\partial x}, u\right). \tag{10}$$

Therefore, $h(x, [(\partial V(x))/(\partial x)])$ can be viewed as a minimization of the given Hamiltonian $H(x, [(\partial V(x))/(\partial x)], u)$. In the following, we refer $h(\cdot, \cdot)$ as the min-Hamiltonian.

Based on [1]–[5], a necessary and sufficient condition for optimality is

$$0 = \inf_u H\left(x, \frac{\partial V^\star(x)}{\partial x}, u\right) \tag{11}$$

with a boundary condition $V^\star(x(\infty)) = 0$. Based on Assumption 1 and (10), the optimal control can be determined $\forall x$ as

$$u^\star(x) = \bar{u}\left(x, \frac{\partial V^\star(x)}{\partial x}\right) = -\frac{1}{2}R^{-1}g^T(x)\frac{\partial V^\star(x)}{\partial x}. \tag{12}$$

Inserting (12) into (11), the HJB equation can be equivalently formulated in terms of the min-Hamiltonian as

$$0 = \inf_u H\left(x, \frac{\partial V^\star(x)}{\partial x}, u\right) = h\left(x, \frac{\partial V^\star(x)}{\partial x}\right). \tag{13}$$

III. Hamiltonian-Driven Framework for Exact Adaptive Dynamic Programming

Here, we will develop a Hamiltonian-driven framework for iterative ADP with convergence proofs to iteratively approximate the solution of HJB equation (11).

A. Background

Three fundamental steps are required to solve the performance optimization problem (2), given the dynamical system (1). These steps are as follows [56].

1) *Policy Evaluation:* To build a criterion that evaluates an arbitrary admissible control $u(\cdot)$, that is, calculate the corresponding cost $J(u(\cdot))$.
2) *Policy Comparison:* To establish a rule that compares two different admissible policies $u(\cdot)$ and $v(\cdot)$.
3) *Policy Improvement:* Based on the current admissible control $u_k(\cdot)$, $k \in \mathbb{Z}$, design a successive control $u_{k+1}(\cdot)$ with an improved cost $J(u_{k+1}(\cdot))$.[1]

[1]Since we are considering a minimization problem, "improved" is achieved given that the cost is monotonically decreasing, that is, $J(u_{k+1}(\cdot)) < J(u_k(\cdot))$, $k \in \{0, 1, 2, \ldots\}$.

Algorithm 1 Model-Based Hamiltonian-Driven Exact ADP Algorithm

1. Initialization: Start with an admissible control policy $u_0(x)$ and set the iteration index as $k = 0$;
2. *Determine iterative value function:* Solve GHJB equation (7) for value function $V_k(x)$ corresponding to $u_k(x)$;
3. *Determine iterative Hamiltonian:* According to (5), calculate the iterative Hamiltonian as $H\left(x, \frac{\partial V_k}{\partial x}, u\right)$;
4. *Update policy:* Update u_{k+1} to minimize $H\left(x, \frac{\partial V_k}{\partial x}, u\right)$, i.e., $u_{k+1} = \arg\inf_u H\left(x, \frac{\partial V_k}{\partial x}, u\right)$;
5. Stop if convergence is achieved. Otherwise, set $k = k + 1$ and go to Step 2.

Lemma 1 (Policy Evaluation [56]): Assume that the system trajectory x is generated by applying an admissible policy $u(\cdot)$ to system (1). Moreover, assume that there exists a unique continuously differentiable and positive-definite solution $V(x)$ to the GHJB equation (7). Then, $V(x)$ and $J(u)$ are equivalent, that is

$$V(x) = J(u; x) \quad \forall x \in \Omega_x.$$

According to Lemma 1, the GHJB equation (7) provides the relationship between the cost (2) of an arbitrary admissible control $u(x)$ and the corresponding value function $V(x)$. Note that by evaluating a given admissible policy, according to Lemma 1, one can avoid solving (1) by only solving the GHJB equation (7) for $V(x)$.

Lemma 2 (Policy Comparison [56]): For $x \in \Omega_x$, let $u_i(x)$ for $i = 1, 2$ be two different admissible policies with the corresponding value functions given by $V_i(x)$ obtained by solving the GHJB equation (7). Denote $h(x, [(\partial V_i(x))/(\partial x)]) = \inf_u H(x, [(\partial V_i(x))/(\partial x)], u)$ as h_i for $i = 1, 2$. Denote $d(u_i, \bar{u}_i) = \|u_i - \bar{u}_i\|_R$ as d_i for $i = 1, 2$, where $\bar{u}_i$ represents $\bar{u}(x, [(\partial V_i(x))/(\partial x)]) = \arg\inf_u H(x, [(\partial V_i(x))/(\partial x)], u)$. Then, the following results hold:

1) $h_i \leq 0$, $i = 1, 2$;
2) $h_1 \leq h_2 \Rightarrow V_1(x) \geq V_2(x) \quad \forall x$;
3) $d_1 \geq d_2 \Rightarrow V_1(x) \geq V_2(x) \quad \forall x$.

Note that the invariant feature of the Hamiltonian is $H(x, [(\partial V_i(x))/(\partial x)], u_i) = 0$ given that $V_i(x)$ is the corresponding value function with respect to the admissible policy $u_i(\cdot)$ that satisfies GHJB equation (7). As proved in Lemma 2, more details about the Hamiltonian $H(x, [(\partial V_i(x))/(\partial x)], u_i)$, that is, d_i and h_i, provide a solution that compares $J(u_1; x_0)$ and $J(u_2; x_0)$, that is, the cost of two different admissible policies u_1 and u_2. Therefore, instead of solving the GHJB equation (7) for $V_i(x)$ on Ω_x, one can compare the performance of two different admissible policies using d_i and h_i.

The PI algorithm can be formulated in terms of the Hamiltonian as in Algorithm 1.

The convergence of Algorithm 1 is investigated as follows.

Lemma 3 (Policy Improvement [56]): Assume that the policy sequence $\{u_k(x)\}_{k=1}^{\infty}$ starts from an admissible policy $u_1(x)$ and that $V_k(x)$ is obtained by solving (7). The policy sequence is generated by

$$u_{k+1}(x) = -\frac{1}{2}R^{-1}g^T(x)\frac{\partial V_k(x)}{\partial x} \quad \forall x \in \Omega_x. \tag{14}$$

Then, the following statements hold.

1) The value function sequence $\{V_k(x)\}$ is a nonincreasing sequence $\forall x$, that is
$$V_k(x) \geq V_{k+1}(x(\cdot)), \ k = 1, 2, \ldots$$
2) Both $\{V_k(x)\}$ and $\{u_k(\cdot)\}$ converge to $V^\star(x)$ and $u^\star(\cdot)$ $\forall x$, respectively, that is
$$\begin{cases} \lim_{k\to\infty} V_k(x) = V^\star(x) \\ \lim_{k\to\infty} u_k(x) = u^\star(x). \end{cases}$$

B. Model-Based Hamiltonian-Driven Exact ADP

The PI algorithm can be interpreted as a successive minimization of the iterative Hamiltonian $H(x, [(\partial V_k)/(\partial x)], u)$. This can be viewed as a special case of the policy comparison step. Compared now to the more general case of Lemmas 2 and 3, we provide an explicit method to obtain the policy $u_{k+1}(\cdot)$ with an improved performance. Note that in the step of the value function determination, the GHJB equation is solved precisely. Therefore, Algorithm 1 is referred to as the exact ADP algorithm in this article.

For subsequent discussions, we define $\mathcal{F}_1$ as

$$\mathcal{F}_1(x, p, q) = q^T f(x) - \frac{1}{2}q^T g(x)R^{-1}g^T(x)p \tag{15}$$

for all $(x, p, q) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n$ and define $\mathcal{F}_2$ as

$$\mathcal{F}_2(x, p) = -\frac{1}{2}p^T g(x)R^{-1}g^T(x)p \tag{16}$$

for all $(x, p) \in \mathbb{R}^n \times \mathbb{R}^n$.

The exact ADP Algorithm 1 can be represented in terms of the min-Hamiltonian $h(\cdot, \cdot)$ as follows.

Corollary 1: Given $V_k(x)$, the model-based Hamiltonian-driven exact ADP algorithm is equivalent to the following iteration:

$$\mathcal{F}_1\left(x, \frac{\partial V_k(x)}{\partial x}, \frac{\partial V_{k+1}(x)}{\partial x} - \frac{\partial V_k(x)}{\partial x}\right) + h\left(x, \frac{\partial V_k(x)}{\partial x}\right) = 0$$

with $V_{k+1}(0) = 0$.

The iteration in Corollary 1 is essentially a nonlinear the Lyapunov equation of $V_{k+1}(x)$ $\forall x \in \Omega_x$, that is

$$0 = \left[\frac{\partial V_{k+1}(x)}{\partial x}\right]^T \left[f(x) - \frac{1}{2}g(x)R^{-1}g^T(x)\frac{\partial V_k(x)}{\partial x}\right] + Q(x) + \frac{1}{4}\left(\frac{\partial V_k(x)}{\partial x}\right)^T g(x)R^{-1}g(x)\frac{\partial V_k(x)}{\partial x}. \tag{17}$$

Equivalently, the exact ADP method can be divided as the policy evaluation and the policy improvement steps, that is, $H(x, [(\partial V_k(x))/(\partial x)], u_k(x)) = 0$ $\forall x \in \Omega_x$ for the policy evaluation step and $u_{k+1}(x) = -(1/2)R^{-1}g^T(x)[(\partial V_k(x))/(\partial x)]$ $\forall x \in \Omega_x$ for the policy improvement step.

Remark 2: The exact iterative ADP algorithm, as shown in Algorithm 2, requires solving exactly a nonlinear Lyapunov equation in each iteration as shown in (17). Due to the

inherent nonlinearity, it is impossible to obtain $V_{k+1}(x)$ satisfying (17), given $V_k(x)$ and there might exist a residual for the nonlinear Lyapunov equation. This residual would propagate through iterations and further affect the convergence of the iterative learning algorithm and the closed-loop stability, which might lead to unreliable results [47]. In this article, we present an inexact iterative ADP algorithm with the existence of the approximation residual in each iteration and convergence guarantee.

IV. Inexact Adaptive Dynamic Programming

A. Model-Based Hamiltonian-Driven Inexact ADP

In the following, we investigate the case when there exists residual error in the PI algorithm and present a sufficient condition to guarantee the monotonic convergence.

In the exact ADP algorithm, given $V_k(x)$ on Ω_x, it is difficult to find $V_{k+1}(x)$ satisfying (17). Therefore, the inexact ADP method is formulated as a value function recursion satisfying

$$\mathcal{F}_1\left(x, \frac{\partial \hat{V}_k(x)}{\partial x}, \frac{\partial \hat{V}_{k+1}(x)}{\partial x} - \frac{\partial \hat{V}_k(x)}{\partial x}\right) + h\left(x, \frac{\partial \hat{V}_k(x)}{\partial x}\right) = \varepsilon_{k+1}(x) \tag{18}$$

for $x \in \Omega_x$ with $V_{k+1}(0) = 0$, which is equivalent to

$$\varepsilon_{k+1}(x) = \left[\frac{\partial \hat{V}_{k+1}}{\partial x}\right]^T \left[f(x) - \frac{1}{2}g(x)R^{-1}g(x)^T\frac{\partial \hat{V}_k}{\partial x}\right] + Q(x) + \frac{1}{4}\left(\frac{\partial \hat{V}_k}{\partial x}\right)^T g(x)R^{-1}g(x)\frac{\partial \hat{V}_k}{\partial x} \tag{19}$$

where $\hat{V}_k(x)$ denotes the iterative value function generated by the inexact ADP algorithm and $\varepsilon_{k+1}(x)$ is the value function recursive approximation residual.

Corollary 2: The iteration in (19) can be viewed as a merge of the inexact policy evaluation step with residual as

$$\varepsilon_k(x) = \left[\frac{\partial \hat{V}_k(x)}{\partial x}\right]^T \left[f(x) + g(x)u_k(x)\right] + Q(x) + \|u_k(x)\|_R^2 \tag{20}$$

for $x \in \Omega_x$ with $V_k(0) = 0$ and the policy update rule

$$u_{k+1}(x) = -\frac{1}{2}R^{-1}g^T(x)\frac{\partial \hat{V}_k(x)}{\partial x} \quad \forall x \in \Omega_x. \tag{21}$$

To this end, the model-based inexact ADP method can be summarized in Algorithm 2. In the next, we investigate the sufficient condition to guarantee the monotonic convergence and the stability guarantee for Algorithm 2.

Lemma 4: For $x \in \Omega_x$, let $u_i(x)$, $i = 1, 2$, be different admissible policies and their corresponding positive definite and continuously differentiable functions value functions be given by $V_i(x)$, $i = 1, 2$, obtained by solving the GHJB equation (7). Assume that for $\forall t \geq 0$

$$\mathcal{F}_1\left(x(t), \frac{\partial V_1(x(t))}{\partial x(t)}, \frac{\partial V_2(x(t))}{\partial x(t)} - \frac{\partial V_1(x(t))}{\partial x(t)}\right) \geq 0. \tag{22}$$

Then, $V_1(x) \geq V_2(x)$ $\forall x \in \Omega_x$.

Algorithm 2 Model-Based Hamiltonian-Driven Inexact ADP

1. Initialization: Start with an initial positive definite value function $\hat{V}_0(\cdot)$ satisfying $h\left(x, \frac{\partial \hat{V}_0(x)}{\partial x}\right) \leq 0$ and set the iteration index as $k = 0$;
2. *Inexact Policy Evaluation:* Find the positive definite value function $\hat{V}_{k+1}(x)$ with $\hat{V}_{k+1}(0) = 0$ satsifying condition (25);
3. *Policy Improvement:* Update $u_{k+1}(\cdot)$ according to (14);
4. Stop if convergence is achieved. Otherwise, set $k = k + 1$ and go to Step 2.

Proof: Denote $\Delta(x) = V_2(x) - V_1(x)$. Then, from the definition of $\mathcal{F}_2$ in (16), condition (22) is equivalent to

$$\frac{d}{dt}\Delta(x(t)) = \left[\frac{\partial \Delta(x)}{\partial x}\right]^T \left[f(x) - \frac{1}{2}g(x)R^{-1}g^T(x)\frac{\partial V_1(x)}{\partial x}\right] \geq 0. \tag{23}$$

Note that $V_i(x)$, $i = 1, 2$, are obtained by solving GHJB equation (7) with admissible policies $u_i(x)$, $i = 1, 2$. Then, $V_i(x)$ can be also viewed as the Lyapunov function satisfying $\lim_{t\to\infty} V_2(x(t)) = \lim_{t\to\infty} V_1(x(t)) = 0$, which implies that

$$\lim_{t\to\infty} \Delta(x(t)) = \lim_{t\to\infty} [V_2(x(t)) - V_1(x(t))] = 0. \tag{24}$$

Combining (23) and (24), one can conclude that $\Delta(x) \leq 0$, that is, $V_1(x) \geq V_2(x)$ $\forall x \in \Omega_x$. This completes the proof. ■

Theorem 1: Let the initial control policy $u_0(\cdot)$ be admissible. Assume that the sequences $\{\hat{V}_k(\cdot)\}_{k=0}^{\infty}$ and $\{u_k(\cdot)\}_{k=0}^{\infty}$ are determined by Algorithm 2. Given that

$$0 \leq \mathcal{F}_1\left(x, \frac{\partial \hat{V}_k(x)}{\partial x}, \frac{\partial \hat{V}_{k+1}(x)}{\partial x} - \frac{\partial \hat{V}_k(x)}{\partial x}\right) + h\left(x, \frac{\partial \hat{V}_k(x)}{\partial x}\right) \leq -\frac{1}{2}\mathcal{F}_2\left(x, \frac{\partial \hat{V}_{k+1}(x)}{\partial x} - \frac{\partial \hat{V}_k(x)}{\partial x}\right) \tag{25}$$

then, the followings hold.

1) $\lim_{k\to\infty} \hat{V}_k(x) = \hat{V}_\infty(x)$ and $\hat{V}_{k+1}(x) \leq \hat{V}_k(x)$ $\forall x \in \Omega_x$, $k = 0, 1, 2, \ldots$
2) The iterative control policy $u_k(\cdot)$ stabilizes the system (1), for $k = 1, 2, \ldots$
3) The value function sequence $\{\hat{V}_k(\cdot)\}_{k=1}^{\infty}$ converges to the solution of the HJB equation.

Proof: From (18), condition (25) implies

$$\varepsilon_{k+1}(x) + \frac{1}{2}\mathcal{F}_2\left(x, \frac{\partial \hat{V}_{k+1}(x)}{\partial x} - \frac{\partial \hat{V}_k(x)}{\partial x}\right) \leq 0. \tag{26}$$

Given $\hat{V}_k(x)$, the iterative value function $\hat{V}_{k+1}(x)$ satisfies, $\forall x \in \Omega_x$

$$h\left(x, \frac{\partial \hat{V}_{k+1}(x)}{\partial x}\right) = \varepsilon_{k+1}(x) + \frac{1}{2}\mathcal{F}_2\left(x, \frac{\partial \hat{V}_{k+1}(x)}{\partial x} - \frac{\partial \hat{V}_k(x)}{\partial x}\right). \tag{27}$$

From (18) and (27), one has

$$\varepsilon_{k+1}(x) = \mathcal{F}_1\left(x, \frac{\partial \hat{V}_k(x)}{\partial x}, \frac{\partial \hat{V}_{k+1}(x)}{\partial x} - \frac{\partial \hat{V}_k(x)}{\partial x}\right)$$

$$+ h\left(x, \frac{\partial \hat{V}_k(x)}{\partial x}\right)$$
$$= -\frac{1}{2}\mathcal{F}_2\left(x, \frac{\partial \hat{V}_{k+1}(x)}{\partial x} - \frac{\partial \hat{V}_k(x)}{\partial x}\right) + h\left(x, \frac{\partial \hat{V}_{k+1}(x)}{\partial x}\right). \quad (28)$$

Based on (27) and (28), the iterative value functions $\hat{V}_{k+1}(x)$ and $\hat{V}_{k+2}(x)$ and approximation residual $\varepsilon_{k+2}(x)$ satisfy

$$\begin{aligned}&\mathcal{F}_1\left(x, \frac{\partial \hat{V}_{k+1}(x)}{\partial x}, \frac{\partial \hat{V}_{k+2}(x)}{\partial x} - \frac{\partial \hat{V}_{k+1}(x)}{\partial x}\right)\\ &= \varepsilon_{k+2}(x) - h\left(x, \frac{\partial \hat{V}_{k+1}(x)}{\partial x}\right)\\ &= \varepsilon_{k+2}(x) - \varepsilon_{k+1}(x) - \frac{1}{2}\mathcal{F}_2\left(x, \frac{\partial \hat{V}_{k+1}(x)}{\partial x} - \frac{\partial \hat{V}_k(x)}{\partial x}\right)\\ &\geq \varepsilon_{k+2}(x) \geq 0 \end{aligned} \quad (29)$$

where the inequality results from (25).

Finally, from Lemma 4, (29) further implies $\hat{V}_{k+2}(x) \leq \hat{V}_{k+1}(x)$. In addition, note that the iterative value function $\hat{V}_k(x)$ generated by Algorithm 2 is positive definite. Then, the value function sequence $\{\hat{V}_k(x)\}_{k=0}^{\infty}$ is nonincreasing and lower bounded. Therefore, $\lim_{k\to\infty} \hat{V}_k(x) = \hat{V}_\infty(x)$ exists.

2) First, when $k = 0$, the initial value function $\hat{V}_0(\cdot)$ satisfying $h(x, [(\partial \hat{V}_0(x))/(\partial x)]) \leq 0$ implies that

$$h\left(x, \frac{\partial \hat{V}_0(x)}{\partial x}\right) = H\left(x, \frac{\partial \hat{V}_0}{\partial x}, -\frac{1}{2}R^{-1}g^T(x)\frac{\partial \hat{V}_0}{\partial x}\right) = H\left(x, \frac{\partial \hat{V}_0}{\partial x}, u_1(x)\right) \leq 0$$

which is equivalent to

$$\dot{\hat{V}}_0(x) = \frac{\partial \hat{V}_0(x)}{\partial x}\left[f(x) + g(x)u_1(x)\right] \leq -Q(x) - u_1^T(x)Ru_1(x).$$

Therefore, $u_1(x)$ can stabilize the closed-loop system.

Second, for the successive value function update, from (19), (25), and (27), one has

$$h\left(x, \frac{\partial \hat{V}_{k+1}(x)}{\partial x}\right) = \varepsilon_{k+1}(x) + \frac{1}{2}\mathcal{F}_2\left(x, \frac{\partial \hat{V}_{k+1}(x)}{\partial x} - \frac{\partial \hat{V}_k(x)}{\partial x}\right) \leq 0.$$

Then, replacing $p \in \mathbb{R}^n$ in (8) with $[(\partial \hat{V}_{k+1}(x))/(\partial x)]$ yields

$$\begin{aligned}h\left(x, \frac{\partial \hat{V}_{k+1}(x)}{\partial x}\right) &= \left[\frac{\partial \hat{V}_{k+1}}{\partial x}\right]^T \times \left[f(x) - \frac{1}{2}g(x)R^{-1}g^T(x)\frac{\partial \hat{V}_{k+1}}{\partial x}\right]\\ &+ \frac{1}{4}\left[\frac{\partial \hat{V}_{k+1}}{\partial x}\right]^T g(x)R^{-1}g^T(x)\frac{\partial \hat{V}_{k+1}}{\partial x}\\ &+ Q(x) \leq 0\end{aligned}$$

for $x \in \Omega_x$, which implies that

$$\dot{\hat{V}}_{k+1}(x) \leq -\frac{1}{4}\left[\frac{\partial \hat{V}_{k+1}}{\partial x}\right]^T g(x)R^{-1}g^T(x)\frac{\partial \hat{V}_{k+1}}{\partial x} - Q(x).$$

Note that $Q(x)$ is a positive-definite function and R is a positive-definite matrix. Then, with the iterative value function $\hat{V}_{k+1}(x)$ being the Lyapunov function, the closed-loop system

$$\dot{x} = f(x) - \frac{1}{2}g(x)R^{-1}g^T(x)\frac{\partial \hat{V}_{k+1}}{\partial x}, \quad k \geq 0,\ t \geq 0$$

has a stable equilibrium given the control policy

$$u(x) = -\frac{1}{2}R^{-1}g^T(x)\frac{\partial \hat{V}_{k+1}(x)}{\partial x}, \quad k \geq 0.$$

Therefore, the iterative policy $u_k(x)$ stabilizes system (1).

3) Since the value function sequence $\{\hat{V}_k(\cdot)\}_{k=0}^{\infty}$ converges, then from (15) and (16), one has

$$\begin{aligned}\lim_{k\to\infty} \hat{V}_{k+1}(x) - \hat{V}_k(x) &= 0\\ \lim_{k\to\infty} \frac{\partial \hat{V}_{k+1}(x)}{\partial x} - \frac{\partial \hat{V}_k(x)}{\partial x} &= 0\\ \lim_{k\to\infty} \mathcal{F}_1\left(x, \frac{\partial \hat{V}_k(x)}{\partial x}, \frac{\partial \hat{V}_{k+1}(x)}{\partial x} - \frac{\partial \hat{V}_k(x)}{\partial x}\right) &= 0\\ \lim_{k\to\infty} \mathcal{F}_2\left(x, \frac{\partial \hat{V}_{k+1}(x)}{\partial x} - \frac{\partial \hat{V}_k(x)}{\partial x}\right) &= 0\end{aligned}$$

for $x \in \Omega_x$. From proposition 1), both $\hat{V}_\infty(x) = \lim_{i\to\infty} \hat{V}_i(x)$ and $u_\infty(x) = -(1/2)R^{-1}g^T(x)[(\partial \hat{V}_\infty(x))/(\partial x)]$ exist. Then, as $k \to \infty$, according to (18) and (25), we have $\lim_{k\to\infty} \varepsilon_k(x) = 0$ and $h(x, [(\partial \hat{V}_\infty(x))/(\partial x)]) = 0$, for $x \in \Omega_x$. Therefore, the iterative value function sequence $\{\hat{V}_k\}_{k=0}^{\infty}$ converges to the solution of the HJB equation (13). This completes the proof. ■

Remark 3: In the kth iteration, given the iterative value function $\hat{V}_k(x)$, the iterative ADP algorithm aims to find the updated iterative value function $\hat{V}_{k+1}(x)$ with convergence to the optimal value function $V^\star(x)$. In Theorem 1, we presented a sufficient condition (25) on the iterative value function gradient $[(\partial \hat{V}_{k+1}(x))/(\partial x)]$ to guarantee the monotonicity of the inexact iterative ADP algorithm. Based on Theorem 1, the iterative value function sequence $\{\hat{V}_k(x)\}_{k=0}^{\infty}$ generated by Algorithm 2 with the recursive condition (25) guarantees both the monotonicity and the closed-loop stability in each iteration. In addition, based on the definition of $\mathcal{F}_2$ in (16), one has $-(1/2)\mathcal{F}_2(x, [(\partial \hat{V}_{k+1})/(\partial x)] - [(\partial \hat{V}_k)/(\partial x)]) \geq 0$ $\forall x \in \Omega_x$. Therefore, condition (25) on $[(\partial \hat{V}_{k+1}(x))/(\partial x)]$ is always feasible.

B. Special Case: Inexact Kleinman-Newton Iteration

The inexact ADP can be applied to linear systems, which is a special case of the developed results, as shown in [57]. Consider the following linear–quadratic regulator (LQR) problem:

$$J(u; x_0) = \int_0^\infty \left[x^T(t)Qx(t) + u^T(t)Ru(t)\right]\mathrm{d}t$$

$$\text{such that} \quad \dot{x} = Ax + Bu, x(0) = x_0, \ t \geq 0.$$

The necessary and sufficient condition for the optimal control can be presented by solving the algebra Riccati equation

$$A^T P + PA - PBR^{-1}B^T P + Q = 0.$$

The ARE is represented as the following matrix equation:

$$\mathcal{R}(X) = A^T X + XA - XBR^{-1}B^T X + Q$$

with

$$\mathcal{F}'_{\mathcal{R}}(X, Y) = A^T Y + YA - YBR^{-1}B^T X - XBR^{-1}B^T Y.$$

The exact PI algorithm and the inexact PI algorithm for solving ARE can be represented as follows:

$$\begin{aligned} &\mathcal{F}'_{\mathcal{R}}(X_k, X_{k+1} - X_k) + \mathcal{R}(X_k) = 0 \\ &\mathcal{F}'_{\mathcal{R}}(X_k, X_{k+1} - X_k) + \mathcal{R}(X_k) = \varepsilon_{k+1}. \end{aligned} \tag{30}$$

Interested readers are referred to [57] for more details.

V. Model-Free Hamiltonian-Driven Inexact ADP

In this section, we present a model-free extension of the inexact iterative ADP algorithm using the off-policy integral RL (IRL) algorithm.

Assumption 2: There exists a value function $\hat{V}_0(x) \in C^1(\Omega_x)$ such that $h(x, [(\partial \hat{V}_0(x))/(\partial x)]) \leq 0$.

In [58], the initial condition is assumed to satisfy

$$H\left(x, \frac{\partial \hat{V}_0(x)}{\partial x}, u_1\right) \leq 0. \tag{31}$$

Note that $h(x, [(\partial \hat{V}_0(x))/(\partial x)]) = \inf_u H(x, [(\partial \hat{V}_0(x))/(\partial x)], u) \leq H(x, [(\partial \hat{V}_0(x))/(\partial x)], u_1)$. Therefore, condition (31) implies that Assumption 2 holds.

A. Off-Policy Integral Reinforcement Learning

Consider the system with the behavior policy $u_b \in \mathbb{R}^m$ as

$$\begin{aligned} \dot{x} &= f(x) + g(x)u_b \\ &= f(x) + g(x)u_k(x) + g(x)[u_b - u_k(x)], \ t \geq 0 \end{aligned}$$

where $u_k(x)$ is the iterative learning policy to approximate the optimal policy $u^\star(x)$.

The time derivative of $\hat{V}_k(x)$ along the state trajectory $\forall t \geq 0$ is

$$\begin{aligned} \frac{\mathrm{d}\hat{V}_k(x)}{\mathrm{d}t} &= \left[\frac{\partial \hat{V}_k(x)}{\partial x}\right]^T \cdot \frac{\mathrm{d}x}{\mathrm{d}t} \\ &= \left[\frac{\partial \hat{V}_k(x)}{\partial x}\right]^T \cdot \{f(x) + g(x)u_k(x) \\ &\qquad + g(x)[u_b - u_k(x)]\}. \end{aligned} \tag{32}$$

From Corollary 2, one has

$$\begin{aligned} &\left[\frac{\partial \hat{V}_k(x)}{\partial x}\right]^T \big[f(x) + g(x)u_k(x)\big] \\ &\quad = \varepsilon_k(x) - Q(x) - u_k^T R u_k. \end{aligned} \tag{33}$$

Consider the data collection phase with instant sequence $\{t_j\}_{j=0}^M$ with $t_{j+1} = t_j + \Delta$, $t_0 \geq 0$ and $t_M \leq T$ on the interval $[0, T]$. Then, in each iteration, integrating both sides of (33) over the interval $[0, T]$ yields the off-policy IRL Bellman equation

$$\begin{aligned} &\int_{t_j}^{t_{j+1}} \left[\frac{\partial \hat{V}_k(x)}{\partial x}\right]^T \big[f(x) + g(x)u_k(x)\big]\mathrm{d}\tau \\ &\quad = \int_{t_j}^{t_{j+1}} \varepsilon_k(x)\mathrm{d}\tau - \int_{t_j}^{t_{j+1}} r(x, u_k(x))\mathrm{d}\tau. \end{aligned} \tag{34}$$

In addition, with the policy improvement update rule (14), one has

$$\left[\frac{\partial \hat{V}_k(x)}{\partial x}\right]^T g(x)[u_b - u_k(x)] = -2u_{k+1}^T(x)R[u_b - u_k(x)]$$

which after integrating yields

$$\begin{aligned} &\int_{t_j}^{t_{j+1}} \left[\frac{\partial \hat{V}_k(x)}{\partial x}\right]^T g(x)[u_b - u_k(x)]\mathrm{d}\tau \\ &\quad = -2\int_{t_j}^{t_{j+1}} u_{k+1}^T(x)R[u_b - u_k(x)]\mathrm{d}\tau. \end{aligned} \tag{35}$$

Combining (34) and (35), by integrating (32), one can obtain

$$\begin{aligned} &\hat{V}_k(x(t+\Delta)) - \hat{V}_k(x(t)) \\ &\quad = \int_{t_j}^{t_{j+1}} \varepsilon_k(x)\mathrm{d}\tau - \int_{t_j}^{t_{j+1}} L(x, u_k(x))\mathrm{d}\tau \\ &\qquad - 2\int_{t_j}^{t_{j+1}} u_{k+1}^T(x)R[u_b(x) - u_k(x)]\mathrm{d}\tau. \end{aligned} \tag{36}$$

Therefore, the term $\int_{t_j}^{t_{j+1}} \varepsilon_k(x)\mathrm{d}\tau$ can be calculated as

$$\begin{aligned} \int_{t_j}^{t_{j+1}} \varepsilon_k \mathrm{d}\tau &= \hat{V}_k(x(t_{k+1})) - \hat{V}_k(x(t_k)) \\ &\quad - 2\int_{t_j}^{t_{j+1}} u_{k+1}^T R[u_k - u_b]\mathrm{d}\tau \\ &\quad + \int_{t_j}^{t_{j+1}} L(x, u_k)\mathrm{d}\tau. \end{aligned} \tag{37}$$

From the definition of $\mathcal{F}_2$ in (16), one has

$$\begin{aligned} \frac{1}{2}\mathcal{F}_2\left(x, \frac{\partial \hat{V}_k}{\partial x} - \frac{\partial \hat{V}_{k-1}}{\partial x}\right) &= -\frac{1}{4}\left[\frac{\partial\left(\hat{V}_k - \hat{V}_{k-1}\right)}{\partial x}\right]^T \\ &\quad \times gR^{-1}g^T \frac{\partial\left(\hat{V}_k - \hat{V}_{k-1}\right)}{\partial x} \\ &= -(u_k - u_{k+1})^T R(u_k - u_{k+1}). \end{aligned} \tag{38}$$

Therefore, the condition $\varepsilon_k \leq -(1/2)\mathcal{F}_2(x, [(\partial \hat{V}_k)/(\partial x)] - [(\partial \hat{V}_{k-1})/(\partial x)])$ in (25) can be rewritten as in the integral form as

$$\int_{t_j}^{t_{j+1}} \varepsilon_k(x)\mathrm{d}\tau \leq \int_{t_j}^{t_{j+1}} -\frac{1}{2}\mathcal{F}_2\left(x, \frac{\partial \hat{V}_k}{\partial x} - \frac{\partial \hat{V}_{k-1}}{\partial x}\right)\mathrm{d}\tau. \tag{39}$$

From (37) and (38), (39) can be further rewritten as

$$\hat{V}_k(x(t_{k+1})) - \hat{V}_k(x(t))$$

$$-2\int_{t_j}^{t_{j+1}} u_{k+1}^T R(u_k-u_b)\mathrm{d}\tau+\int_{t_j}^{t_{j+1}} L(x,u_k)\mathrm{d}\tau \le -\int_{t_j}^{t_{j+1}} (u_k-u_{k+1})^T R(u_k-u_{k+1})\mathrm{d}\tau. \quad (40)$$

Similarly, from (37), the condition $\varepsilon_k \ge 0$ in (25) can be rewritten in an integral form as

$$0 \le \hat{V}_k(x(t_{k+1}))-\hat{V}_k(x(t_k))+\int_{t_j}^{t_{j+1}} L(x,u_k)\mathrm{d}\tau -2\int_{t_j}^{t_{j+1}} u_{k+1}^T R[u_k-u_b]\mathrm{d}\tau. \quad (41)$$

Therefore, the inexact ADP algorithm aims to find the iterative value function $\hat{V}_{k+1}(x)$ that satisfies (40) and (41).

B. Off-Policy Integral Reinforcement Learning With Neural Network

Given an actor–critic representation, namely, an actor to approximate the optimal policy and a critic to approximate the optimal cost, one has

$$\hat{V}_k(x)=\hat{W}_{c,k}^T\phi_c(x),\ u_k(x)=\phi_a^T(x)\hat{W}_{a,k} \quad (42)$$

where $\hat{W}_{c,k}\in\mathbb{R}^{N_1}$ is the critic weight, $\phi_c:\mathbb{R}^n\to\mathbb{R}^{N_1}$ is the critic basis, $\hat{W}_{a,k}\in\mathbb{R}^{N_2}$ is the actor weight, and $\phi_a:\mathbb{R}^n\to\mathbb{R}^{N_1\times m}$ is the actor basis. It is desired to determine a rigorously justifiable form for the actor–critic structure. We assume that both the actor and critic basis are dense in the Sobolev norm $W^{1,\infty}$, since the Sobolev norm is desired for value function approximation as well as the value gradient approximation [22]. Conventional utilization of the Weierstrass high-order approximation theorem for the basis function design is polynomial functions [53].

Equation (40) can be further parameterized as

$$\begin{aligned}&\hat{W}_{c,k}^T[\phi_c(x(t+\Delta))-\phi_c(x(t))]+\int_{t_j}^{t_{j+1}} L(x,u_k)\mathrm{d}\tau\\&\quad-2\int_{t_j}^{t_{j+1}} \hat{W}_{a,k+1}^T\phi_a(x)R(u_k-u_b)\mathrm{d}\tau\\&\le-\int_{t_j}^{t_{j+1}} \hat{W}_{a,k+1}^T\phi_a(x)R\phi_a^T(x)\hat{W}_{a,k+1}\mathrm{d}\tau\\&\quad-\int_{t_j}^{t_{j+1}} u_k^T R u_k\mathrm{d}\tau+2\int_{t_j}^{t_{j+1}} \hat{W}_{a,k+1}^T\phi_a(x)Ru_k\mathrm{d}\tau \quad (43)\end{aligned}$$

with a compact form given by

$$\hat{W}_k^T A_j\hat{W}_k+\hat{W}_k^T b_{j,k}+c_{j,k}\le 0 \quad (44)$$

where $\hat{W}_k=\begin{bmatrix}\hat{W}_{c,k}\\ \hat{W}_{a,k+1}\end{bmatrix}$ with

$$A_j=\begin{bmatrix}0 & 0\\ 0 & \int_{t_j}^{t_{j+1}}\phi_a(x)R\phi_a^T(x)\mathrm{d}\tau\end{bmatrix}$$
$$b_{j,k}=\begin{bmatrix}\phi_c\big(x(t_{j+1})\big)-\phi_c\big(x(t_j)\big)\\ \varpi_{j,k}\end{bmatrix}$$
$$c_{j,k}=\int_{t_j}^{t_{j+1}} u_k^T R u_k\mathrm{d}\tau+\int_{t_j}^{t_{j+1}} L(x,u_k)\mathrm{d}\tau$$

Algorithm 3 Model-Free Hamiltonian-Driven Inexact ADP

1. Initialization: Start with policy $u_1(\cdot)$ and value function $\hat{V}_0(x)$, set $k=0$;
1. *Data Collection Phase:* Collect the online data to assemble data-based matrix A_j, $b_{j,k}$, $c_{j,k}$, $d_{j,k}$ and $\gamma_{j,k}$ with instant sequence $\{t_j\}_{j=0}^M$;
2. *Learning Phase:* Solve QCQP (49) to obtain $\{W_{c,k}, W_{a,k+1}\}$ for $\{\hat{V}_k(x), u_{k+1}(x)\}$, respectively;
3. Stop if convergence is achieved. Otherwise, set $k=k+1$ and go to Step 2.

$$\varpi_{j,k}=-2\int_{t_j}^{t_{j+1}}\phi_a(x)R(u_k-u_b)\mathrm{d}\tau-2\int_{t_j}^{t_{j+1}}\phi_a(x)Ru_k\mathrm{d}\tau. \quad (45)$$

Similarly, (41) can be further parameterized as

$$\begin{aligned}&\hat{W}_{c,k}^T\big[\phi_c(x(t_{k+1}))-\phi_c(x(t_k))\big]\\&\quad-2\int_{t_k}^{t_{k+1}}\hat{W}_{a,k+1}^T\phi_a(x)R(u_k-u_b)\mathrm{d}\tau\\&\ge-\int_{t_k}^{t_{k+1}} L(x,u_k)\mathrm{d}\tau \quad (46)\end{aligned}$$

with a compact form given by

$$\hat{W}_k^T d_{j,k}+\gamma_{j,k}\le 0 \quad (47)$$

where

$$d_{j,k}=\begin{bmatrix}\phi_c\big(x(t_j)\big)-\phi_c\big(x(t_{j+1})\big)\\ 2\int_{t_j}^{t_{j+1}}\phi_a(x)R(u_k-u_b)\mathrm{d}\tau\end{bmatrix}$$
$$\gamma_{j,k}=-\int_{t_j}^{t_{j+1}} L(x,u_k)\mathrm{d}\tau. \quad (48)$$

Then, finding the iterative value function $\hat{V}_{k+1}(x)$ that satisfies (40) and (41) is equivalent to finding the weight $\hat{W}_k$ that satisfies (44) and (47). However, the weight $\hat{W}_k$ that satisfies (44) and (47) might not be unique. Therefore, we present the following quadratically constrained quadratic program (QCQP) to determine the weight $\hat{W}_k$ in each iteration:

$$\min_{\hat{W}_k}\ \left\|\hat{W}_k\right\|^2 \quad \text{such that}\ \begin{cases}\hat{W}_k^T A_j\hat{W}_k+\hat{W}_k^T b_{j,k}+c_{j,k}\le 0\\ \hat{W}_k^T d_{j,k}+\gamma_{j,k}\le 0.\end{cases} \quad (49)$$

To this end, the model-free inexact ADP method can be summarized in Algorithm 3.

Corollary 3: The off-policy IRL equation (34) has the same solution for the value function as the Bellman equation (19), and the same updated control policy as described in Corollary 2.

Proof: Dividing both sides of the off-policy IRL Bellman equation (34) by T and taking the limit as $T\to 0$, one can obtain

$$\left[\frac{\partial\hat{V}_k(x)}{\partial x}\right]^T\big[f(x)+g(x)u_k+g(x)(u-u_k)\big]+Q(x)+\|u_k\|_R+2u_{k+1}^T R(u-u_k)=\varepsilon_k. \quad (50)$$

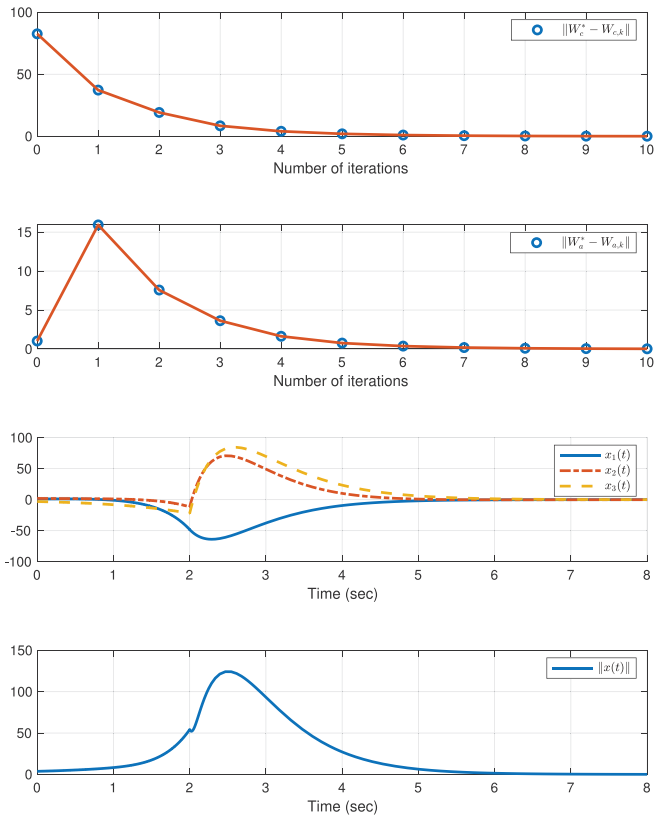


Fig. 1. Model-free inexact ADP for a linear system. The first two figures illustrate the convergence of the actor and critic networks. The latter two figures show that after learning using the collected data during the first 2 s, the approximate optimal policy stabilizes the system.

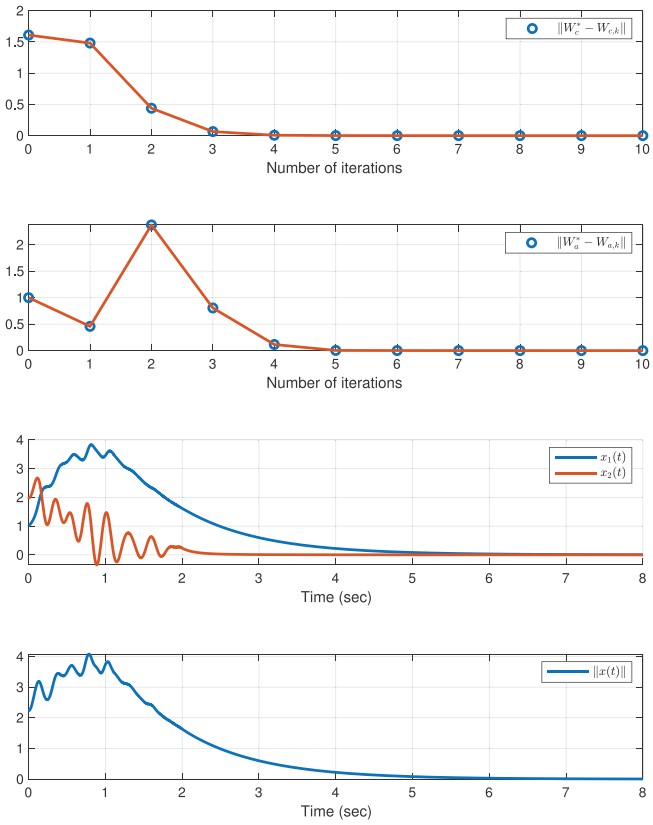


Fig. 2. Model-free inexact ADP for nonlinear system. The first two figures illustrate the convergence of the actor and critic networks. The latter two figures show that after learning using the collected data during the first 2 s, the approximate optimal policy stabilizes the system.

Inserting the policy improvement rule into (50) yields the inexact policy evaluation (19). This completes the proof. ■

VI. Simulations

A. Linear System

Consider the continuous-time linear system given by

$$\dot{x} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} u, \ t \geq 0.$$

We consider the reward function as $L(x, u) = x^T Q x + u^T R u$ with Q and R being identity matrices of appropriate dimensions. Based on LQR, the optimal control and the optimal value function can be determined as

$$u^\star(x) = \begin{bmatrix} -8.3056 & -2.2827 & -4.6607 \\ -8.5707 & -2.7323 & -2.2827 \end{bmatrix} x$$

$$V^\star(x) = x^T \begin{bmatrix} 31.0746 & 8.5707 & 8.3056 \\ 8.5707 & 2.7323 & 2.2827 \\ 8.3056 & 2.2827 & 4.6607 \end{bmatrix} x.$$

The behavior policy for data collection is designed as a linear combination of sinusoidal signals. For the inexact ADP learning, we employ the quadratic polynomial basis function for the critic network. Given ten iterations, the results are shown in Fig. 1. One can observe that for the inexact ADP algorithm, both the iterative quadratic value function and the iterative linear feedback policy converge to the optimal value function and optimal control policy, respectively.

B. Nonlinear System

Consider the following nonlinear dynamical system [56]:

$$\dot{x}_1 = -x_1 + x_1 x_2^2$$
$$\dot{x}_2 = -x_2 + x_1 u, \ t \geq 0$$

with the reward function selected as, $L(x, u) = 2(x_1^2 + x_2^2) + 0.5u^2$. The optimal control and the optimal value function can be determined as $u^\star(x) = -2x_1 x_2$ and $V^\star(x) = x_1^2 + x_2^2$, respectively. Based on the quadratic form of the optimal value function, we select the quadratic polynomial basis function for the critic network. After learning with the data collected from the online system running phase, the results are shown in Fig. 2. One can observe that within eight iterations, both the iterative quadratic value function and the iterative linear feedback policy converge to the optimal value function and optimal control policy, respectively.

VII. Conclusion

This article leveraged the Hamiltonian-driven framework, where the Hamiltonian of admissible control policies plays

an important role in the iterative performance improvement of the control policy. First, the HJB equation was represented in terms of the min-Hamiltonian function in order to apply the quasi-Newton iterative method to solve it. A novel PI algorithm with bounded approximation error in each iteration was discussed in detail. A sufficient condition was derived for the iterative value gradient update with approximation error in each iteration to guarantee the convergence of the inexact ADP algorithm.

Future work will focus on extending the work to networked systems.

REFERENCES

[1] J. A. E. Bryson and Y.-C. Ho, *Applied Optimal Control: Optimization, Estimation and Control*. Abingdon, U.K.: Taylor & Francis, 1975.

[2] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 1st ed. Belmont, MA, USA: Athena Sci., 1995.

[3] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*, 3rd ed. Hoboken, NJ, USA: Wiley, 2012.

[4] J. L. Speyer and D. H. Jacobson, *Primer on Optimal Control Theory*, vol. 20. Philadelphia, PA, USA: SIAM, 2010.

[5] D. Liberzon, *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton, NJ, USA: Princeton Univ. Press, 2012.

[6] L. S. Pontryagin, *Mathematical Theory of Optimal Processes*. Boca Raton, FL, USA: CRC Press, 1987.

[7] R. Bellman, *Dynamic Programming*. Princeton, NJ, USA: Princeton Univ. Press, 1957.

[8] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," in *Handbook of Intelligent Control*, D. A. White and D. A. Sofge, Eds. New York, NY, USA: Van Nostrand Reinhold Company, 1992, pp. 493–526.

[9] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. London, U.K.: Athena Sci., 1996.

[10] P. J. Werbos, *The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting*. New York, NY, USA: Wiley, 1994.

[11] D. V. Prokhorov and D. C. Wunsch, "Adaptive critic designs," *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 997–1007, Sep. 1997.

[12] R. Padhi, N. Unnikrishnan, X. Wang, and S. Balakrishnan, "A single network adaptive critic (SNAC) architecture for optimal control synthesis for a class of nonlinear systems," *Neural Netw.*, vol. 19, no. 10, pp. 1648–1660, 2006.

[13] H. He, Z. Ni, and J. Fu, "A three-network architecture for on-line learning and optimization based on adaptive dynamic programming," *Neurocomputing*, vol. 78, no. 1, pp. 3–13, 2012.

[14] J. Si and Y.-T. Wang, "Online learning control by association and reinforcement," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 264–276, Mar. 2001.

[15] Z. Ni, H. He, X. Zhong, and D. V. Prokhorov, "Model-free dual heuristic dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1834–1839, Aug. 2015.

[16] Y. Yang, K. G. Vamvoudakis, H. Modares, Y. Yin, and D. C. Wunsch, "Safe intermittent reinforcement learning with static and dynamic event generators," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 12, pp. 5441–5455, Dec. 2020.

[17] B. Luo, T. Huang, and D. Liu, "Periodic event-triggered suboptimal control with sampling period and performance analysis," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1253–1261, Mar. 2021.

[18] Y. Yang, K. G. Vamvoudakis, H. Modares, Y. Yin, and D. C. Wunsch, "Hamiltonian-driven hybrid adaptive dynamic programming," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Jan. 13, 2020, doi: 10.1109/TSMC.2019.2962103.

[19] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, no. 2, pp. 477–484, 2009.

[20] H. Modares and F. L. Lewis, "Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning," *IEEE Trans. Autom. Control*, vol. 59, no. 11, pp. 3051–3056, Nov. 2014.

[21] H. N. Wu and B. Luo, "Neural network based online simultaneous policy update algorithm for solving the HJI equation in nonlinear H_∞ control," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 12, pp. 1884–1895, Dec. 2012.

[22] 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.

[23] T. Dierks and S. Jagannathan, "Online optimal control of affine nonlinear discrete-time systems with unknown internal dynamics by using time-based policy update," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 7, pp. 1118–1129, Jul. 2012.

[24] D. Wang, H. He, and D. Liu, "Adaptive critic nonlinear robust control: A survey," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3429–3451, Oct. 2017.

[25] Q. Zhang, D. Zhao, and D. Wang, "Event-based robust control for uncertain nonlinear systems using adaptive dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 37–50, Jan. 2018.

[26] S. Bhasin, R. Kamalapurkar, M. Johnson, K. Vamvoudakis, F. Lewis, and W. Dixon, "A novel actor–critic–identifier architecture for approximate optimal control of uncertain nonlinear systems," *Automatica*, vol. 49, no. 1, pp. 82–92, 2013.

[27] Y. Jiang and Z.-P. Jiang, "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics," *Automatica*, vol. 48, no. 10, pp. 2699–2704, 2012.

[28] T. Bian, Y. Jiang, and Z.-P. Jiang, "Adaptive dynamic programming and optimal control of nonlinear nonaffine systems," *Automatica*, vol. 50, no. 10, pp. 2624–2632, 2014.

[29] T. Bian and Z. P. Jiang, "Reinforcement learning and adaptive optimal control for continuous-time nonlinear systems: A value iteration approach," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 8, 2021, doi: 10.1109/TNNLS.2020.3045087.

[30] Y. Yang, B. Kiumarsi, H. Modares, and C. Xu, "Model-free λ-policy iteration for discrete-time linear quadratic regulation," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Aug. 11, 2021, doi: 10.1109/TNNLS.2021.3098985.

[31] H. Modares, F. L. Lewis, and Z. P. Jiang, "H_∞ tracking control of completely unknown continuous-time systems via off-policy reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2550–2562, Oct 2015.

[32] B. Luo, H. N. Wu, and T. Huang, "Off-policy reinforcement learning for H_∞ control design," *IEEE Trans. Cybern.*, vol. 45, no. 1, pp. 65–76, Jan. 2015.

[33] Q. Zhang and D. Zhao, "Data-based reinforcement learning for nonzero-sum games with unknown drift dynamics," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 2874–2885, Aug. 2019.

[34] L. N. Tan, "Event-triggered distributed H_∞ control of physically interconnected mobile Euler–Lagrange systems with slipping, skidding and dead zone," *IET Control Theory Appl.*, vol. 14, no. 3, pp. 438–451, 2020.

[35] L. N. Tan, "Distributed H_∞ optimal tracking control for strict-feedback nonlinear large-scale systems with disturbances and saturating actuators," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 11, pp. 4719–4731, Aug. 2020.

[36] B. Luo, Y. Yang, and D. Liu, "Policy iteration Q-learning for data-based two-player zero-sum game of linear discrete-time systems," *IEEE Trans. Cybern.*, vol. 51, no. 7, pp. 3630–3640, Jul. 2021.

[37] T. Bian, Y. Jiang, and Z. Jiang, "Adaptive dynamic programming for stochastic systems with state and control dependent noise," *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 4170–4175, Dec. 2016.

[38] T. Bian and Z.-P. Jiang, "Continuous-time robust dynamic programming," *SIAM J. Control Optim.*, vol. 57, no. 6, pp. 4150–4174, 2019.

[39] Z.-P. Jiang, T. Bian, and W. Gao, "Learning-based control: A tutorial and some recent results," *Found. Trends Syst. Control*, vol. 8, no. 3, pp. 176–284, 2020.

[40] Y. Yang, Z. Guo, H. Xiong, D. Ding, Y. Yin, and D. C. Wunsch, "Data-driven robust control of discrete-time uncertain linear systems via off-policy reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 12, pp. 3735–3747, Dec. 2019.

[41] B. Kiumarsi, F. L. Lewis, and Z.-P. Jiang, "H_∞ control of linear discrete-time systems: Off-policy reinforcement learning," *Automatica*, vol. 78, pp. 144–152, Apr. 2017.

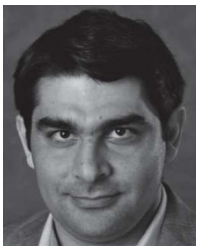
[42] P. Yan, D. Wang, H. Li, and D. Liu, "Error bound analysis of Q-function for discounted optimal control problems with policy iteration," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1207–1216, Jul. 2017.

[43] F. Y. Wang, N. Jin, D. Liu, and Q. Wei, "Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with ε-error bound," *IEEE Trans. Neural Netw.*, vol. 22, no. 1, pp. 24–36, Jan. 2011.
[44] D. Liu and Q. Wei, "Finite-approximation-error-based optimal control approach for discrete-time nonlinear systems," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 779–789, Apr. 2013.
[45] Q. Wei, F. Y. Wang, D. Liu, and X. Yang, "Finite-approximation-error-based discrete-time iterative adaptive dynamic programming," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2820–2833, Dec. 2014.
[46] A. Heydari, "Stability analysis of optimal adaptive control using value iteration with approximation errors," *IEEE Trans. Autom. Control*, vol. 63, no. 9, pp. 3119–3126, Sep. 2018.
[47] A. Heydari, "Theoretical and numerical analysis of approximate dynamic programming with approximation errors," *J. Guid. Control Dyn.*, vol. 39, no. 2, pp. 301–311, 2016.
[48] D. Liu, H. Li, and D. Wang, "Error bounds of adaptive dynamic programming algorithms for solving undiscounted optimal control problems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 6, pp. 1323–1334, Jun. 2015.
[49] D. Kleinman, "On an iterative technique for Riccati equation computations," *IEEE Trans. Autom. Control*, vol. AC-13, no. 1, pp. 114–115, Feb. 1968.
[50] A. J. van der Schaft, "L_2-gain analysis of nonlinear systems and nonlinear state-feedback H_∞ control," *IEEE Trans. Autom. Control*, vol. 37, no. 6, pp. 770–784, Jun. 1992.
[51] 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.
[52] R. W. Beard, "Improving the closed-loop performance of nonlinear systems," Ph.D. dissertation, Dept. Comput. Sci., Rensselaer Polytechn. Inst., Troy, NY, USA, 1995.
[53] 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 Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 10, pp. 1513–1525, Oct. 2013.
[54] R. W. Beard, G. N. Saridis, and J. T. Wen, "Galerkin approximations of the generalized Hamilton–Jacobi–Bellman equation," *Automatica*, vol. 33, no. 12, pp. 2159–2177, 1997.
[55] K. R. Davidson and A. P. Donsig, *Real Analysis and Applications: Theory in Practice*. New York, NY, USA: Springer, 2009.
[56] Y. Yang, D. Wunsch, and Y. Yin, "Hamiltonian-driven adaptive dynamic programming for continuous nonlinear dynamical systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 8, pp. 1929–1940, Aug. 2017.
[57] F. Feitzinger, T. Hylla, and E. Sachs, "Inexact Kleinman–Newton method for Riccati equations," *SIAM J. Matrix Anal. Appl.*, vol. 31, no. 2, pp. 272–288, 2009.
[58] Y. Jiang and Z. Jiang, "Global adaptive dynamic programming for continuous-time nonlinear systems," *IEEE Trans. Autom. Control*, vol. 60, no. 11, pp. 2917–2929, Nov. 2015.

Yongliang Yang (Member, IEEE) received the B.S. degree in electrical engineering from Hebei University, Baoding, China, in 2011, and the Ph.D. degree in electrical engineering from the University of Science and Technology Beijing (USTB), Beijing, China, in 2018.

From 2015 to 2017, he was a Visiting Scholar with the Missouri University of Science and Technology, Rolla, MO, USA, sponsored by China Scholarship Council. From 2018 to 2020, he was an Assistant Professor with USTB. From 2020 to 2021, he was a Postdoctoral Research Fellow with the State Key Laboratory of Internet of Things for Smart City, Faculty of Science and Technology, University of Macau, Macau, China. He is currently an Associate Professor with USTB. His research interests include reinforcement learning theory, robotics, distributed optimization, and control for cyber–physical systems.

Dr. Yang was a recipient of the Best Ph.D. Dissertation of China Association of Artificial Intelligence, the Best Ph.D. Dissertation of USTB, the Chancellor's Scholarship in USTB, the Excellent Graduates Awards in Beijing, and the UM Macao Talent Programme in Macau. He also serves as the Reviewer for several international journals and conferences, including *Automatica*, IEEE TRANSACTIONS ON AUTOMATIC CONTROL, IEEE TRANSACTIONS ON CYBERNETICS, and IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.



Hamidreza Modares (Senior Member, IEEE) received the B.Sc. degree from Tehran University, Tehran, Iran, in 2004, the M.Sc. degree from the Shahrood University of Technology (SUT), Shahrood, Iran, in 2006, and the Ph.D. degree from the University of Texas at Arlington (UTA), Arlington, TX, USA, in 2015.

From 2006 to 2009, he was with SUT, as a Senior Lecturer. From 2015 to 2016, he was a Faculty Research Associate with UTA. From 2016 to 2018, he was an Assistant Professor with the Department of Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, MO, USA. He is currently an Assistant Professor with the Department of Mechanical Engineering, Michigan State University, East Lansing, MI, USA. He has authored several journal and conference papers on the design of optimal controllers using reinforcement learning. His current research interests include cyber–physical systems, machine learning, distributed control, robotics, and renewable energy microgrids.

Dr. Modares was a recipient of the Best Paper Award from the 2015 IEEE International Symposium on Resilient Control Systems, the Stelmakh Outstanding Student Research Award from the Department of Electrical Engineering, UTA, in 2015, and the Summer Dissertation Fellowship from UTA in 2015. He is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.

Kyriakos G. Vamvoudakis (Senior Member, IEEE) was born in Athens, Greece. He received the Diploma (a 5 year degree, equivalent to a Master of Science) (with Highest Hons.) degree in electronic and computer engineering from the Technical University of Crete, Chania, Greece, in 2006, and the M.S. and Ph.D. degrees in electrical engineering from the University of Texas at Arlington (UTA), Arlington, TX, USA, in 2008 and 2011, respectively, under the guidance of F. L. Lewis.

From May 2011 to January 2012, he was working as an Adjunct Professor and a Faculty Research Associate with UTA and the Automation and Robotics Research Institute. From 2012 to 2016, he was a Project Research Scientist with the Center for Control, Dynamical Systems and Computation, University of California at Santa Barbara, Santa Barbara, CA, USA. He was an Assistant Professor with the Kevin T. Crofton Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, VA, USA, until 2018. He currently serves as an Assistant Professor with the Daniel Guggenheim School of Aerospace Engineering, Georgia Tech, Atlanta, GA, USA. His research interests include approximate dynamic programming, game theory, and optimal control. Recently, his research has focused on cyber–physical security, networked control, smart grid, and multiagent optimization.

Dr. Vamvoudakis is a recipient of the 2019 ARO YIP Award, the 2018 NSF CAREER Award, and of several international awards including, the 2016 International Neural Network Society Young Investigator (INNS) Award, the Best Paper Award for Autonomous/Unmanned Vehicles at the 27th Army Science Conference in 2010, the Best Presentation Award at the World Congress of Computational Intelligence in 2010, and the Best Researcher Award from the Automation and Robotics Research Institute, in 2011. He is an Associate Editor of the *IEEE Computational Intelligence Magazine* and the *Journal of Optimization Theory and Applications*, the Editor in Chief of the *Communications in Control Science and Engineering*, a Registered Electrical/Computer Engineer (PE), and a member of the Technical Chamber of Greece. He is a member of Tau Beta Pi, Eta Kappa Nu, and Golden Key Honor Societies and is listed in Who's Who in the World, Who's Who in Science and Engineering, and Who's Who in America. He has also served for various international program committees and has organized special sessions for several international conferences. He is currently a member of the Technical Committee on Intelligent Control of the IEEE Control Systems Society and the Technical Committee on Adaptive Dynamic Programming and Reinforcement Learning of the IEEE Computational Intelligence Society.

Wei He (Senior Member, IEEE) received the B.Eng. degree in automation and the M.Eng. degree in control science and engineering from the College of Automation Science and Engineering, South China University of Technology, Guangzhou, China, in 2006 and 2008, respectively, and the Ph.D. degree in control science and engineering from the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, in 2011.

He is currently a Full Professor with the School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, China. He has coauthored two books published in Springer and published over 100 international journal and conference papers. His current research interests include robotics, distributed parameter systems, and intelligent control systems.

Prof. He was a recipient of the Newton Advanced Fellowship from the Royal Society, U.K., in 2017, and the IEEE SMC Society Andrew P. Sage Best Transactions Paper Award, in 2017. He is serving as an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, IEEE/CAA JOURNAL OF AUTOMATICA SINICA, and *Neurocomputing*, and an Editor for the *Journal of Intelligent and Robotic Systems*. He is serving as the Chair of IEEE SMC Society Beijing Capital Region Chapter.

Cheng-Zhong Xu (Fellow, IEEE) received the B.Sc. and M.Sc. degrees in computer science and engineering from Nanjing University, Nanjing, China, in 1986 and 1989, respectively, and the Ph.D. degree in computer science and engineering from The University of Hong Kong, Hong Kong, in 1993.

He is currently the Dean of the Faculty of Science and Technology and the Interim Director of the Institute of Collaborative Innovation, University of Macau, Macau, China, and a Chair Professor of Computer and Information Science. He was a Professor with Wayne State University, Detroit, MI, USA, and the Director of the Institute of Advanced Computing, Shenzhen Institutes of Advanced Technologies, Chinese Academy of Sciences, Beijing, China, before he joined UM in 2019. He is a Chief Scientist of Key Project on Smart City of MOST, China, and a Principal Instigator of the Key Project on Autonomous Driving of FDCT, Macau SAR. He published two research monographs and more than 300 peer-reviewed papers in journals and conference proceedings; his papers received over 10K citations with an H-index of 54. He also received more than 100 patents or PCT patents and spun off a business "Shenzhen Baidou Applied Technology" with dedication to location-based services and technologies. His main research interests lie in parallel and distributed computing and cloud computing, in particular, with an emphasis on resource management for system's performance, reliability, availability, power efficiency, and security, and in big data and data-driven intelligence applications in smart city, and self-driving vehicles. The systems of particular interest include distributed systems and the Internet, servers, and cloud datacenters, scalable parallel computers, and wireless embedded devices and mobile edge systems.

Dr. Xu was a Best Paper Nominee or Awardee of the 2013 IEEE High Performance Computer Architecture, the 2013 ACM High Performance Distributed Computing, the IEEE Cluster'2015, the ICPP'2015, the GPC'2018, the UIC'2018, and the AIMS'2019. He received the most prestigious "President's Awards for Excellence in Teaching" of Wayne State University, in 2002. He has been the Chair of IEEE Technical Committee on Distributed Processing, from 2015 to 2020. He serves or served for a number of journal editorial boards, including IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, *Journal of Parallel and Distributed Computing*, *Science China: Information Science*, and *ZTE Communication*.

Donald C. Wunsch (Fellow, IEEE) received the B.S. degree in applied mathematics from the University of New Mexico, Albuquerque, NM, USA, in 1984, the M.S. degree in applied mathematics and the Ph.D. degree in electrical engineering from the University of Washington, Seattle, WA, USA, in 1987 and 1991, respectively, and the Executive M.B.A. degree from Washington University in St. Louis, St. Louis, MO, USA, in 2006.

He attended the Jesuit Core Honors Program with Seattle University, Seattle. He was with Texas Tech University, Lubbock, TX, USA, The Boeing Company, Seattle; Rockwell International, Albuquerque; and International Laser Systems, Albuquerque. He is currently the Mary K. Finley Missouri Distinguished Professor with the Missouri University of Science and Technology (Missouri S&T), Rolla, MO, USA, where he is also the Director of the Applied Computational Intelligence Laboratory, a multidisciplinary research group. His current research interests include clustering/unsupervised learning, biclustering, adaptive resonance and adaptive dynamic programming architectures, hardware, and applications, neurofuzzy regression, autonomous agents, games, and bioinformatics.

Dr. Wunsch was a recipient of the NSF CAREER Award, the 2015 INNS Gabor Award, and the 2019 Ada Lovelace Service Award. He has produced 21 Ph.D. recipients in computer engineering, electrical engineering, systems engineering, and computer science. He is an International Neural Networks Society (INNS) Fellow and the INNS President. He served as the IJCNN General Chair, and on several boards, including the St. Patricks School Board, the IEEE Neural Networks Council, the INNS, and the University of Missouri Bioinformatics Consortium, and the Chair of the Missouri S&T Information Technology and Computing Committee as well as the Student Design and Experiential Learning Center Board.