Integral Reinforcement Learning for Linear Continuous-Time Zero-Sum Games With Completely Unknown Dynamics

Hongliang Li, *Student Member, IEEE*, Derong Liu, *Fellow, IEEE*, and Ding Wang

***Abstract*—In this paper, we develop an integral reinforcement learning algorithm based on policy iteration to learn online the Nash equilibrium solution for a two-player zero-sum differential game with completely unknown linear continuous-time dynamics. This algorithm is a fully model-free method solving the game algebraic Riccati equation forward in time. The developed algorithm updates value function, control and disturbance policies simultaneously. The convergence of the algorithm is demonstrated to be equivalent to Newton's method. To implement this algorithm, one critic network and two action networks are used to approximate the game value function, control and disturbance policies, respectively, and the least squares method is used to estimate the unknown parameters. The effectiveness of the developed scheme is demonstrated in the simulation by designing an H_∞ state feedback controller for a power system.**

***Note to Practitioners*—Noncooperative zero-sum differential game provides an ideal tool to study multiplayer optimal decision and control problems. Existing approaches usually solve the Nash equilibrium solution by means of offline iterative computation, and require the exact knowledge of the system dynamics. However, it is difficult to obtain the exact knowledge of the system dynamics for many real-world industrial systems. The algorithm developed in this paper is a fully model-free method which solves the zero-sum differential game problem forward in time by making use of online measured data. This method is not affected by errors between an identification model and a real system, and responds fast to changes of the system dynamics. Exploration signals are required to satisfy the persistence of excitation condition to update the value function and the policies, and these signals do not affect the convergence of the learning process. The least squares method is used to obtain the approximate solution for the zero-sum games with unknown dynamics. The developed algorithm is applied to a load-frequency controller design for a power system whose parameters are not known *a priori*. In future research, we will extend the results to zero-sum and nonzero-sum differential games with completely unknown nonlinear continuous-time dynamics.**

***Index Terms*—Adaptive critic designs, adaptive dynamic programming, approximate dynamic programming, reinforcement learning, policy iteration, zero-sum games.**

Manuscript received August 29, 2013; revised November 29, 2013; accepted January 12, 2014. Date of publication January 31, 2014; date of current version June 30, 2014. This paper was recommended for publication by Associate Editor D. Garg and Editor H. Ding upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of Chinaunder Grant 61034002, Grant 61233001, Grant 61273140, and Grant 61304086, and in part by the Early Career Development Award of SKLMCCS.

The authors are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: hongliang.li@ia.ac.cn; derong.liu@ia.ac.cn; ding.wang@ia.ac.cn).

Digital Object Identifier 10.1109/TASE.2014.2300532

I. Introduction

ADAPTIVE DYNAMIC PROGRAMMING (ADP) [1]–[4] and reinforcement learning (RL) [5] have received significantly increasing attention as machine learning and optimization methods. These algorithms can solve the optimal control problem forward in time by making use of online measured data, while the exact knowledge of the system dynamics is not required. For discrete-time dynamical systems, ADP and RL have obtained great success in both theories [6]–[15] and applications [16]–[25]. He *et al.* [26], [27] proposed a dual critic network design that contains the internal goal representation to help approximate the value function.

For continuous-time dynamical systems, Doya [28] presented RL framework without a prior discretization of time, state and control. Vamvoudakis and Lewis [29] proposed a synchronous policy iteration (PI) algorithm for learning online the continuous-time optimal control with known dynamics, where both action and critic neural networks were simultaneously tuned. Zhang *et al.* [30] extended the synchronous PI algorithm to the optimal tracking problem for unknown nonlinear systems, and added a robust term to compensate for the neural network approximation errors. Bhasin *et al.* [31] presented an actor-critic-identifier structure to implement the PI algorithm without the requirement of complete knowledge of the dynamics. However, the technique by identifying the system parameters responds slowly to parameter variations from the plant. A pioneering work was that Vrabie *et al.* derived an integral RL algorithm to obtain direct adaptive optimal control for partially unknown linear [32] and nonlinear systems [33]. It is more meaningful to develop adaptive optimal control algorithms for completely unknown systems without identification. Mehta and Meyn [34] established connections between Q-learning and nonlinear optimal control of continuous-time models, and proposed continuous-time Q-learning for completely unknown systems. Lee *et al.* [35] derived an integral Q-learning for linear continuous-time systems without the knowledge of the system dynamics. Jiang and Jiang [36] presented a computational adaptive optimal control algorithm for linear continuous-time systems with completely unknown system dynamics. The algorithms in [35] and [36] have similar properties, and they are similar to the action-dependent heuristic dynamic programming for unknown discrete-time systems [16].

Game theory provides an ideal environment to study multiplayer optimal decision and control problems. Two-player noncooperative zero-sum differential game [37] has received much attention since it provides the solution of the H_∞ optimal control [38]. The Nash equilibrium solution is usually obtained by means of offline iterative computation, and the exact knowledge of the system dynamics is required. For linear discrete-time systems, Al-Tamimi *et al.* [39] solved online the zero-sum game using ADP methods, and proposed a model-free Q-learning iterative algorithm [40]. Kim and Lewis [41] presented a model-free H_∞ control algorithm by using Q-learning with linear matrix inequalities.

For nonlinear continuous-time systems, Abu-Khalaf *et al.* [42], [43] derived a two-player PI to design an H_∞ suboptimal state feedback controller. Zhang *et al.* [44] used four action networks and two critic networks to obtain the saddle point solution of the game. Vamvoudakis and Lewis [45] presented an online synchronous PI to solve the two-player zero-sum game with known dynamics. In [46], the Hamilton–Jacobi–Isaacs (HJI) equation was solved online using a novel single approximator-based scheme to achieve optimal regulation and tracking control of affine nonlinear continuous-time systems. In [47], a neural-network-based online simultaneous policy update algorithm with only one iterative loop was proposed to solve the HJI equation for partially unknown systems, and the convergence was established by proving that it was mathematically equivalent to Newton's method.

When the system has linear dynamics and the performance index is quadratic, finding the Nash equilibrium of the zero-sum game problem reduces to solving the game algebraic Riccati equation (GARE). Vrabie and Lewis [48] proposed an online data-based ADP algorithm based on the idea of integral RL for two-player zero-sum differential games without requiring the knowledge of internal system dynamics. The algorithm leads to the equilibrium solution of the Nash game while only one of the players actively learns to optimize its policy and the other passively plays based on fixed policies. There are two iterative loops, thus the method is often time-consuming [49]. Wu and Luo [49] proposed an online simultaneous policy update algorithm for H_∞ state feedback control to improve the efficiency by updating policies of both control player and disturbance player simultaneously, where only one iterative loop was involved. It should be mentioned that the methods above only relax the requirement of exact knowledge on the internal system dynamics.

However, for many practical problems, it is difficult for us to obtain the knowledge of the system dynamics. Hence, the ADP methods mentioned above cannot be directly applied to zero-sum games with completely unknown dynamics. In this paper, we develop an online integral RL algorithm to learn the Nash equilibrium solution for a two-player zero-sum linear differential game with completely unknown dynamics. It results in a fully model-free method solving the GARE forward in time for the first time, where both internal and drift system dynamics are not required. The developed algorithm updates value function, control and disturbance policies simultaneously. The convergence of the algorithm is demonstrated to be equivalent to Newton's method. To implement this algorithm, one critic network and two action networks are used to approximate the game value function, control and disturbance policies, respectively, and the least squares method is used to estimate the unknown parameters.

The rest of this paper is organized as follows. Section II provides the formulation of a two-player zero-sum differential game. In Section III, we first develop a model-free integral RL for zero-sum games, then provide the convergence analysis, and finally give the least squares method to estimate the unknown parameters. Section IV presents a simulation example in power systems to demonstrate the effectiveness of the developed algorithm and is followed by concluding remarks in Section V.

Notations: $\mathbb{R}^+$, $\mathbb{R}^n$, and $\mathbb{R}^{n\times m}$ are the set of positive real numbers, the n-dimensional Euclidean space and the set of all real $n \times m$ matrices, respectively. $\|\cdot\|$ denotes the vector norm or matrix norm in $\mathbb{R}^n$ or $\mathbb{R}^{n\times m}$. I_n denotes the n-dimensional identity matrix. Denote $\mathbb{Z}_+$ the set of nonnegative integers. Use $\mathrm{vec}(X)$ for $X \in \mathbb{R}^{n\times m}$ as a vectorization map from a matrix into an mn-dimensional column vector which stacks the column of X on top of one another. For $X \in \mathbb{R}^{n\times m}$ and $Y \in \mathbb{R}^{n\times m}$, we let $X \otimes Y$ be a Kronecker product of X and Y. The superscript T is used for the transpose. $\nabla_x f(x,y) \triangleq \partial f(x,y)/\partial x$ denotes a gradient operator notation.

II. Problem Formulation

Consider a class of linear continuous-time dynamical systems described by

$$\dot{x} = Ax + B_1 u + B_2 w \tag{1}$$

where $x \in \mathbb{R}^n$ is the system state with initial state x_0, $u \in \mathbb{R}^m$ is the control input, and $w \in \mathbb{R}^q$ is the external disturbance input with $w \in L_2[0,\infty)$. $A \in \mathbb{R}^{n\times n}$, $B_1 \in \mathbb{R}^{n\times m}$, and $B_2 \in \mathbb{R}^{n\times q}$ are unknown system matrices.

Define the infinite horizon performance index

$$\begin{aligned} J(x_0,u,w) &= \int_0^\infty (x^\mathsf{T} Qx + u^\mathsf{T} Ru - \gamma^2 w^\mathsf{T} w)\mathrm{d}\tau \\ &\triangleq \int_0^\infty r(x,u,w)\mathrm{d}\tau \end{aligned} \tag{2}$$

with $Q = Q^\mathsf{T} \geq 0$, $R = R^\mathsf{T} > 0$, and a prescribed constant $\gamma \geq \gamma^* \geq 0$, where γ^* denotes the smallest γ for which the system (1) is stabilized. For feedback policy $u(x)$ and disturbance policy $w(x)$, we define the value function of the policies as

$$V(x_t,u,w) = \int_t^\infty (x^\mathsf{T} Qx + u^\mathsf{T} Ru - \gamma^2 w^\mathsf{T} w)\mathrm{d}\tau. \tag{3}$$

Then, we define the two-player zero-sum differential game as

$$\begin{aligned} V^*(x_0) &= \min_u \max_w J(x_0,u,w) \\ &= \min_u \max_w \int_0^\infty (x^\mathsf{T} Qx + u^\mathsf{T} Ru - \gamma^2 w^\mathsf{T} w)\mathrm{d}\tau \end{aligned}$$

where the control policy player u seeks to minimize the performance index, while the disturbance policy player w desires to maximize it. The goal is to find the saddle point (u^*, w^*) which satisfies the following inequalities:

$$J(x_0,u,w^*) \geq J(x_0,u^*,w^*) \geq J(x_0,u^*,w)$$

for any state feedback control policy u and disturbance policy w.

We use notations $u = -Kx$ and $w = Lx$ for the state feedback control policy and the disturbance policy, respectively. Then, the value function (3) can be represented as $V(x_t) = x_t^{\mathsf{T}} P x_t$, where the matrix P is determined by K and L. The saddle point can be obtained by solving the following continuous-time GARE [37]:

$$A^{\mathsf{T}} P^* + P^* A + Q - P^* B_1 R^{-1} B_1^{\mathsf{T}} P^* + \gamma^{-2} P^* B_2 B_2^{\mathsf{T}} P^* = 0. \tag{4}$$

Defining P^* as the unique positive definite solution of (4), the saddle point of the zero-sum game is

$$u^* = -K^* x = -R^{-1} B_1^{\mathsf{T}} P^* x \tag{5}$$

$$w^* = L^* x = \gamma^{-2} B_2^{\mathsf{T}} P^* x \tag{6}$$

and the game value function is

$$V^*(x_0) = x_0^{\mathsf{T}} P^* x_0.$$

The solution of the H_∞ control problem can be obtained by solving the saddle point of the equivalent two-player zero-sum game problem. The following H_∞ norm (the L_2-gain) is used to measure the performance of the control system.

Definition 1: Let $\gamma \geq 0$ be certain prescribed level of disturbance attenuation. The system (1) is said to have L_2-gain less than or equal to γ if

$$\int_0^\infty (x^{\mathsf{T}} Q x + u^{\mathsf{T}} R u) \mathrm{d}\tau \leq \gamma^2 \int_0^\infty (w^{\mathsf{T}} w) \mathrm{d}\tau \tag{7}$$

for all $w \in L_2[0, \infty)$.

The H_∞ control problem is to find a state feedback control policy u such that the closed-loop system is asymptotically stable and satisfies the condition (7). For every $\gamma \geq \gamma^*$, the GARE has a unique positive definite solution [37].

A stabilizing control policy exists with the following standard assumptions: The pair (A, B_1) is stabilizable and the pair $(A, Q^{1/2})$ is observable.

III. Main Results

The methods for solving the GARE of a linear zero-sum differential game can be classified into: online or offline algorithms and algorithms with one or two iterative loops. The offline methods [43] require complete knowledge of the system dynamics. The algorithms with two iterative loops [48] have lower efficiency than those with one iterative loop [49]. The online simultaneous policy update algorithm proposed in [49] has only one iterative loop, but it requires the input gain matrix. On the other hand, an integral Q-learning algorithm [35], [36] was proposed to solve the adaptive optimal control of linear continuous-time systems with completely unknown dynamics. In this section, we first develop an online model-free integral RL algorithm for the linear continuous-time zero-sum game with completely unknown systems. We then provide the convergence analysis, and finally present the online implementation using the least squares method.

A. Model-Free Integral RL for Zero-Sum Games

In this part, we will develop an online model-free integral RL algorithm for the linear continuous-time zero-sum differential game with completely unknown dynamics. First, we assume an initial stabilizing control matrix K_0 to be known. Define $V_i(x) = x^{\mathsf{T}} P_i x$, $u_i(x) = -K_i x$, and $w_i(x) = L_i x$ as the value function, control policy and disturbance policy, respectively, for each iterative step $i \geq 0$.

To relax the assumptions of exact knowledge of A, B_1, and B_2, we use e_1 and e_2 to denote the exploration signals added to the control policy u_i and disturbance policy w_i, respectively. The exploration signals are assumed to be any nonzero measurable signal which is bounded by $e_M > 0$, i.e., $\|e_1\| \leq e_M$, $\|e_2\| \leq e_M$. Then, the original system (1) becomes

$$\dot{x} = Ax + B_1(u_i + e_1) + B_2(w_i + e_2). \tag{8}$$

The derivative of the value function with respect to time is calculated as

$$\dot{V}_i(x) = -x^{\mathsf{T}} Q x - x^{\mathsf{T}} K_i^{\mathsf{T}} R K_i x + \gamma^2 x^{\mathsf{T}} L_i^{\mathsf{T}} L_i x + 2 x^{\mathsf{T}} K_{i+1}^{\mathsf{T}} R e_1 + 2\gamma^2 x^{\mathsf{T}} L_{i+1}^{\mathsf{T}} e_2. \tag{9}$$

Integrating (9) from t and $t+T$ with any time interval $T > 0$, we have

$$x_{t+T}^{\mathsf{T}} P_i x_{t+T} - x_t^{\mathsf{T}} P_i x_t = -\int_t^{t+T} r(x, u_i, w_i) \mathrm{d}\tau + 2\int_t^{t+T} x^{\mathsf{T}} K_{i+1}^{\mathsf{T}} R e_1 \mathrm{d}\tau + 2\gamma^2 \int_t^{t+T} x^{\mathsf{T}} L_{i+1}^{\mathsf{T}} e_2 \mathrm{d}\tau$$

where the values of the state at time t and $t+T$ are denoted with x_t and x_{t+T}. Therefore, we obtain the online model-free integral RL algorithm (Algorithm 1) for zero-sum differential games.

Algorithm 1: Online Model-Free Integral RL for Zero-Sum Games

Step 1. Give an initial stabilizing policy $u_1 = -K_1 x$ and $w_1 = L_1 x$. Set $i = 1$ and $P_0 = 0$.

Step 2. (Policy Evaluation and Improvement)
For the system (8) with policies $u_i = -K_i x$ and $w_i = L_i x$, and exploration signals e_1 and e_2, solve the following equation for P_i, K_{i+1} and L_{i+1}

$$x_t^T P_i x_t = x_{t+T}^{\mathsf{T}} P_i x_{t+T} + \int_t^{t+T} r(x, u_i, w_i) \mathrm{d}\tau - 2\int_t^{t+T} x^{\mathsf{T}} K_{i+1}^{\mathsf{T}} R e_1 \mathrm{d}\tau - 2\gamma^2 \int_t^{t+T} x^{\mathsf{T}} L_{i+1}^{\mathsf{T}} e_2 \mathrm{d}\tau. \tag{10}$$

Step 3. If $\|P_i - P_{i-1}\| \leq \xi$ (ξ is a prescribed small positive real number), stop and output P_i; else, set $i = i + 1$ and go to Step 2.

Remark 1: Equation (10) plays an important role in relaxing the assumption of the knowledge of system dynamics, since A, B_1, and B_2 do not appear in (10). Only online data measured along the system trajectories are required to run this algorithm. The exploration signals can satisfy the persistence of excitation (PE) condition to efficiently update the value function and the policies.

Remark 2: Our method avoids the identification of A, B_1, and B_2 whose information is embedded in the online measured data. In other words, the lack of knowledge about the system dynamics does not have any impact on our method to obtain the Nash equilibrium. Thus, our method will not be affected by the errors between the identification model and the real system, and it can respond fast to changes of the system dynamics.

Remark 3: This algorithm is actually a PI method, but the policy evaluation and improvement are performed at the same time. Compared with the model-based method [45] and partially model-free method [49], our algorithm is a fully model-free method which does not require knowledge of the system dynamics. Different from the iterative method with inner loop on disturbance policy and outer loop on control policy [45], and the method with only one iterative loop by updating control and disturbance policies simultaneously [49], the developed method here updates the value function, control and disturbance policies at the same time. Hence, our method will have higher efficiency.

Remark 4: The PE condition in adaptive control is very similar to the exploration and exploitation in RL. To guarantee the PE condition, the state may need to be reset during the iterative process, but it results in technical problems for stability analysis of the closed-loop system. An alternative way is to add exploration noises. The solution obtained by our method is exactly the same as the one determined by the GARE by considering the effects of exploration noises.

Next, we will show the relationship between the developed algorithm and Q-learning algorithm by extending the concept of Q-function to zero-sum games that are continuous in time, state and action space.

The optimal continuous-time Q-function for zero-sum games is defined as the following quadratic form:

$$\begin{aligned} &Q^*(x,u,w)\\ &= [x^\top u^\top w^\top] H^* [x^\top u^\top w^\top]^\top \\ &= [x^\top u^\top w^\top] \begin{bmatrix} H_{11}^* & H_{12}^* & H_{13}^* \\ H_{21}^* & H_{22}^* & H_{23}^* \\ H_{31}^* & H_{32}^* & H_{33}^* \end{bmatrix} \begin{bmatrix} x \\ u \\ w \end{bmatrix} \\ &= [x^\top u^\top w^\top] \begin{bmatrix} A^\top P^* + P^* A + Q & P^* B_1 & P^* B_2 \\ B_1^\top P^* & R & 0 \\ B_2^\top P^* & 0 & -\gamma^2 \end{bmatrix} \begin{bmatrix} x \\ u \\ w \end{bmatrix}. \end{aligned} \tag{11}$$

It can be seen that the matrix H^* is associated with P^* in GARE. By solving $\nabla_u Q^*(x,u,w) = 0$ and $\nabla_w Q^*(x,u,w) = 0$, we can obtain

$$u^* = -(H_{22}^*)^{-1}(H_{12}^*)^\top x \tag{12}$$

$$w^* = -(H_{33}^*)^{-1}(H_{13}^*)^\top x \tag{13}$$

which are the same as the (5) and (6). Since we have

$$Q^*(x_0, u^*, w^*) = V^*(x_0)$$

the relationship between P^* and H^* can be represented as

$$P^* = [\, I_n \quad -K^\top \quad L^\top \,] \, H^* \begin{bmatrix} I_n \\ -K \\ L \end{bmatrix}.$$

According to (11), we can obtain

$$H_{11}^* = H_{12}^*(H_{22}^*)^{-1}H_{21}^* + H_{13}^*(H_{33}^*)^{-1}H_{31}^*$$

and thus H_{11}^* is a redundant term. Define

$$H^i = \begin{bmatrix} H_{11}^i & H_{12}^i & H_{13}^i \\ H_{21}^i & R & 0 \\ H_{31}^i & 0 & -\gamma^2 \end{bmatrix}$$

where $H_{21}^i = (H_{12}^i)^\top$ and $H_{31}^i = (H_{13}^i)^\top$. Then, we can develop the following online integral Q-learning algorithm (Algorithm 2).

Algorithm 2: Online Integral Q-Learning for Zero-Sum Games

Step 1. Give an initial stabilizing policy $u_1 = -K_1 x$ and $w_1 = L_1 x$.

Step 2. Set $i = 0$, $H_{11}^0 = 0$, $H_{12}^0 = K_1^\top R$, and $H_{13}^0 = \gamma^2 L_1^\top$.

Step 3. (Policy Evaluation)
Let $i = i + 1$. For the system (8) with policies $u_i = -K_i x$ and $w_i = L_i x$, and exploration signals e_1 and e_2, solve the following equation for H_{11}^i, H_{12}^i and H_{13}^i

$$\begin{aligned} x_t^\top H_{11}^i x_t = x_{t+T}^\top H_{11}^i x_{t+T} + \int_t^{t+T} r(x, u_i, w_i) \mathrm{d}\tau \\ -2 \int_t^{t+T} x^\top H_{12}^i e_1 \mathrm{d}\tau - 2 \int_t^{t+T} x^\top H_{13}^i e_2 \mathrm{d}\tau. \end{aligned} \tag{14}$$

Step 4. (Policy Improvement)
Update the following parameters

$$K_{i+1} = R^{-1} \left(H_{12}^i\right)^\top, L_{i+1} = \gamma^{-2} \left(H_{13}^i\right)^\top.$$

Step 5. If $\|H_{11}^i - H_{11}^{i-1}\| + \|H_{12}^i - H_{12}^{i-1}\| + \|H_{13}^i - H_{13}^{i-1}\| \le \zeta$ (ζ is a prescribed small positive real number), stop and output H_i; else, go to Step 3.

Remark 5: It can be seen that the developed model-free integral RL (Algorithm 1) is equivalent to the integral Q-learning (Algorithm 2) for zero-sum games. As PI methods, the algorithms developed above require an initial stabilizing control policy which is usually obtained by experience. We can also obtain $B_1 = (H_{11}^i)^{-1} H_{12}^i$ and $B_2 = (H_{11}^i)^{-1} H_{13}^i$.

B. Convergence Analysis of Model-Free Integral RL for Zero-Sum Games

In this part, we will provide a convergence analysis of the developed algorithms for two-player zero-sum differential games. It can be shown that the developed model-free integral RL and Q-learning algorithms are equivalent to Newton's method.

Theorem 1: For an initial stabilizing control policy $u_1 = -K_1 x$, the sequences of $\{P_i\}_{i=1}^{\infty}$, $\{K_i\}_{i=1}^{\infty}$, and $\{L_i\}_{i=1}^{\infty}$ obtained by solving (10) in Algorithm 1 converge to the optimal solution P^* of GARE, the saddle point K^*, and L^*, respectively, as $i \to \infty$.

Proof: First, for an initial stabilizing control policy $u_1 = -K_1 x$, we can prove that the developed Algorithm 1 is equivalent to the following Lyapunov equation:

$$A_i^{\top} P_i + P_i A_i = -M_i \tag{15}$$

where

$$\begin{aligned} A_i &= A - B_1 K_i + B_2 L_i, \\ M_i &= Q + K_i^{\top} R K_i - \gamma^2 L_i^{\top} L_i. \end{aligned}$$

With the control policy $u_i = -K_i x$, the disturbance policy $w_i = L_i x$, and the exploration signals e_1 and e_2, the closed-loop system (1) becomes

$$\dot{x} = A_i x + B_1 e_1 + B_2 e_2$$

where $A_i = A - B_1 K_i + B_2 L_i$. Considering the Lyapunov function $V_i(x) = x^{\top} P_i x$, its derivative can be calculated as

$$\begin{aligned} \dot{V}_i(x) &= \dot{x}^{\top} P_i x + x^{\top} P_i \dot{x} \\ &= x^{\top} A_i^{\top} P_i x + x^{\top} P_i A_i x \\ &\quad + (B_1 e_1 + B_2 e_2)^{\top} P_i x + x^{\top} P_i (B_1 e_1 + B_2 e_2) \\ &= x^{\top} \left(A_i^{\top} P_i + P_i A_i \right) x + 2 x^{\top} K_{i+1}^{\top} R e_1 + 2\gamma^2 x^{\top} L_{i+1}^{\top} e_2. \end{aligned} \tag{16}$$

Integrating (16) from t and $t + T$ yields

$$\begin{aligned} V_i(x_{t+T}) - V_i(x_t) &= \int_t^{t+T} x^{\top} \left(A_i^{\top} P_i + P_i A_i \right) x \mathrm{d}\tau \\ +2 \int_t^{t+T} x^{\top} K_{i+1}^{\top} R e_1 \mathrm{d}\tau &+ 2 \int_t^{t+T} \gamma^2 x^{\top} L_{i+1}^{\top} e_2 \mathrm{d}\tau. \end{aligned} \tag{17}$$

According to (10), we have

$$\begin{aligned} V_i(x_{t+T}) - V_i(x_t) &= -\int_t^{t+T} r(x, u_i, w_i) \mathrm{d}\tau \\ +2 \int_t^{t+T} x^{\top} K_{i+1}^{\top} R e_1 \mathrm{d}\tau &+ 2 \int_t^{t+T} \gamma^2 x^{\top} L_{i+1}^{\top} e_2 \mathrm{d}\tau. \end{aligned} \tag{18}$$

Therefore, considering (17) and (18), we have

$$\begin{aligned} x^{\top} \left(A_i^{\top} P_i + P_i A_i \right) x &= -r(x, u_i, w_i) \\ &= -x^{\top} \left(Q + K_i^{\top} R K_i - \gamma^2 L_i^{\top} L_i \right) x \end{aligned}$$

i.e.,

$$A_i^{\top} P_i + P_i A_i = -M_i$$

where

$$M_i = Q + K_i^{\top} R K_i - \gamma^2 L_i^{\top} L_i.$$

Then, according to the results in [49], the sequence $\{P_i\}_{i=1}^{\infty}$ generated by (15) is equivalent to Newton's method and converges to the optimal solution P^* of GARE, as $i \to \infty$. Furthermore, the sequences $\{K_i\}_{i=1}^{\infty}$ and $\{L_i\}_{i=1}^{\infty}$ converge to the saddle point K^* and L^*, respectively, as $i \to \infty$. ■

Remark 6: In [45], the value function of the inner loop is monotonically increasing, while the value function of the outer loop is monotonically decreasing. The monotonicity of game value function using our algorithm is monotonically decreasing as Newton's method, and the game value function has quadratic convergence. We can find that the exploration signals do not affect the convergence of the learning process.

The next theorem will show the convergence of the model-free integral Q-learning algorithm for zero-sum games.

Theorem 2: For an initial stabilizing control policy $u_1 = -K_1 x$, the sequence $\{H^i\}_{i=1}^{\infty}$ obtained by solving (14) in Algorithm 2 converges to the optimal solution H^*, i.e., $Q_i \to Q^*$, as $i \to \infty$.

Proof: Because the iteration process of H^i with solving (14) in Algorithm 2 is equivalent to that of P_i with solving (10) in Algorithm 1, then as $i \to \infty$

$$H^i \to \begin{bmatrix} A^{\top} P^* + P^* A + Q & P^* B_1 & P^* B_2 \\ B_1^{\top} P^* & R & 0 \\ B_2^{\top} P^* & 0 & -\gamma^2 \end{bmatrix} = H^*.$$

■

C. Online Implementation of Model-Free Integral RL for Zero-Sum Games

In this part, an online implementation of Algorithm 1 is developed based on ADP with the least squares method. Algorithm 2 can be implemented by the same way. Here, parametric structures are used to approximate the game value function, control policy, and disturbance policy.

Given a stabilizing control policy $u_i = -K_i x$, a triplet (P_i, K_{i+1}, L_{i+1}) with $P_i = P_i^{\top} > 0$, can be uniquely determined by (10). We define the following two operators: $P \in \mathbb{R}^{n \times n} \to \hat{P} \in \mathbb{R}^{(1/2)n \times (n+1)}$, $x \in \mathbb{R}^n \to \bar{x} \in \mathbb{R}^{(1/2)n \times (n+1)}$, where

$$\begin{aligned} \hat{P} &= [p_{11}, 2p_{12}, \ldots, 2p_{1n}, p_{22}, 2p_{23}, \ldots, 2p_{(n-1)n}, p_{nn}]^{\top} \\ \bar{x} &= \left[x_1^2, x_1 x_2, \ldots, x_1 x_n, x_2^2, x_2 x_3, \ldots, x_{n-1} x_n, x_n^2 \right]^{\top}. \end{aligned}$$

Hence, we have

$$\begin{aligned} x_{t+(k-1)T}^{\top} P_i x_{t+(k-1)T} - x_{t+kT}^{\top} P_i x_{t+kT} \\ = \left(\bar{x}_{t+(k-1)T} - \bar{x}_{t+kT} \right)^{\top} \hat{P} \end{aligned}$$

where $k \in \mathbb{Z}_+$ and $k \geq 1$. Using Kronecker product $\otimes$, we can obtain

$$\begin{aligned} x^{\top} K_{i+1}^{\top} R e_1 &= (x \otimes e_1)^{\top} (I_n \otimes R) \mathrm{vec}(K_{i+1}) \\ x^{\top} L_{i+1}^{\top} e_2 &= (x \otimes e_2)^{\top} \mathrm{vec}(L_{i+1}). \end{aligned}$$

Using the expressions established above, (10) can be rewritten in a general compact form as

$$\psi_k^{\mathsf{T}} \begin{bmatrix} \hat{P}_i \\ \mathrm{vec}(K_{i+1}) \\ \mathrm{vec}(L_{i+1}) \end{bmatrix} = \theta_k, \quad \forall i \in \mathbb{Z}_+ \tag{19}$$

where

$$\theta_k = \int_{t+(k-1)T}^{t+kT} r(x, u_i, w_i)\mathrm{d}\tau$$

$$\psi_k = \left[\left(\bar{x}_{t+(k-1)T} - \bar{x}_{t+kT}\right)^{\mathsf{T}}, 2\int_{t+(k-1)T}^{t+kT} (x \otimes e_1)^{\mathsf{T}} \mathrm{d}\tau (I_n \otimes R), \right.$$
$$\left. 2\gamma^2 \int_{t+(k-1)T}^{t+kT} (x \otimes e_2)^{\mathsf{T}} \mathrm{d}\tau \right]^{\mathsf{T}}$$

where the measurement time is from $t+(k-1)T$ to $t+kT$. Since (19) is only a one-dimensional equation, we cannot guarantee the uniqueness of the solution. We will use the least squares method to solve this problem, where the parameter vector is determined in a least squares sense over a compact set Ω.

For any positive integer N, denote $\Phi = [\psi_1, \ldots, \psi_N]$ and $\Theta = [\theta_1, \ldots, \theta_N]^{\mathsf{T}}$. Then, we have the following N-dimensional equation:

$$\Phi^{\mathsf{T}} \begin{bmatrix} \hat{P}_i \\ \mathrm{vec}(K_{i+1}) \\ \mathrm{vec}(L_{i+1}) \end{bmatrix} = \Theta, \quad \forall i \in \mathbb{Z}_+.$$

If Φ^{T} has full column rank, the parameters can be solved by

$$\begin{bmatrix} \hat{P}_i \\ \mathrm{vec}(K_{i+1}) \\ \mathrm{vec}(L_{i+1}) \end{bmatrix} = (\Phi\Phi^{\mathsf{T}})^{-1}\Phi\Theta. \tag{20}$$

Therefore, we need to have the number of collected points N at least $N_{\min} = \mathrm{rank}(\Phi)$, i.e.,

$$N_{\min} = \frac{n(n+1)}{2} + nm + nq$$

which will guarantee $(\Phi\Phi^{\mathsf{T}})^{-1}$ exist.

The least squares problem in (20) can be solved in real time by collecting enough data points generated from the system (8). The flowchart of this algorithm is shown in Fig. 1. The solution can be obtained using the batch least squares, the recursive least squares algorithms, or the gradient descent algorithms.

Remark 7: The sequence $\{P_i\}_{i=1}^{\infty}$ calculated by the least squares method converges to the approximate solution of GARE. The PE condition is required in adaptive control to perform system identification. Several types of exploration signal have been used, such as piecewise constant exploration signals [35], sinusoidal signals with different frequencies [36], random noise [39], [44], and exponentially decreasing probing noise [45].

IV. Simulation Study

In this section, we will demonstrate the effectiveness of the developed algorithm by designing an H_∞ state feedback controller for a power system.

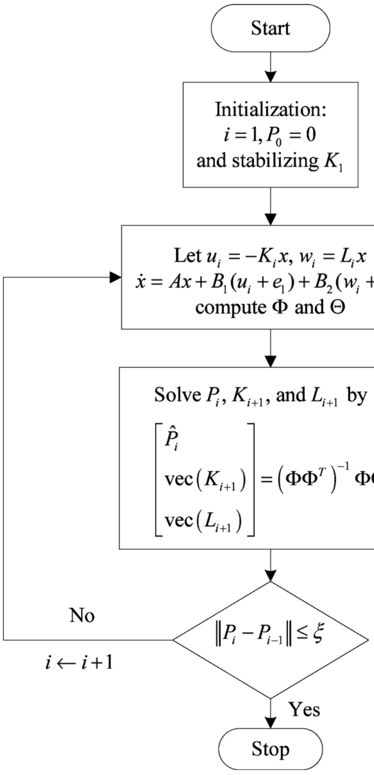
Start
Initialization: $i=1, P_0=0$ and stabilizing K_1
Let $u_i=-K_i x$, $w_i=L_i x$
$\dot{x}=Ax+B_1(u_i+e_1)+B_2(w_i+e_2)$
compute Φ and Θ
Solve P_i, K_{i+1}, and L_{i+1} by
$\begin{bmatrix} \hat{P}_i \\ \mathrm{vec}(K_{i+1}) \\ \mathrm{vec}(L_{i+1}) \end{bmatrix} = (\Phi\Phi^T)^{-1}\Phi\Theta$
$\|P_i - P_{i-1}\| \le \xi$
No
$i \leftarrow i+1$
Yes
Stop

Fig. 1. Flowchart of Algorithm 1.

Consider the following linear model of a power system that was studied in [48]:

$$\begin{aligned} \dot{x} &= Ax + B_1 u + B_2 w \\ &= \begin{bmatrix} -0.0665 & 8 & 0 & 0 \\ 0 & -3.663 & 3.663 & 0 \\ -6.86 & 0 & -13.736 & -13.736 \\ 0.6 & 0 & 0 & 0 \end{bmatrix} x \\ &\quad + \begin{bmatrix} 0 \\ 0 \\ 13.736 \\ 0 \end{bmatrix} u + \begin{bmatrix} -8 \\ 0 \\ 0 \\ 0 \end{bmatrix} w \end{aligned} \tag{21}$$

where the state vector is $x = [\triangle f \; \triangle P_g \; \triangle X_g \; \triangle E]^{\mathsf{T}}$, $\triangle f$ (Hz) is the incremental frequency deviation, $\triangle P_g$ (p.u. MW) is the incremental change in generator output, $\triangle X_g$ (p.u. MW) is the incremental change in governor value position, and $\triangle E$ is the incremental change in integral control. We assume that the dynamics of the system (21) is unknown. The matrices Q and R in the performance index are identity matrices of appropriate dimensions, and $\gamma = 3.5$. Using the system model (21), the matrix in the optimal value function of the zero-sum game is

$$P^* = \begin{bmatrix} 0.8335 & 0.9649 & 0.1379 & 0.8005 \\ 0.9649 & 1.4751 & 0.2358 & 0.8046 \\ 0.1379 & 0.2358 & 0.0696 & 0.0955 \\ 0.8005 & 0.8046 & 0.0955 & 2.6716 \end{bmatrix}.$$

Now, we will use the developed online model-free integral RL algorithm to solve this problem. The initial state is selected as $x_0 = [0.1, 0.2, 0.2, 0.1]^{\mathsf{T}}$. The simulation is conducted using data obtained along the system trajectory at every 0.01 s. The

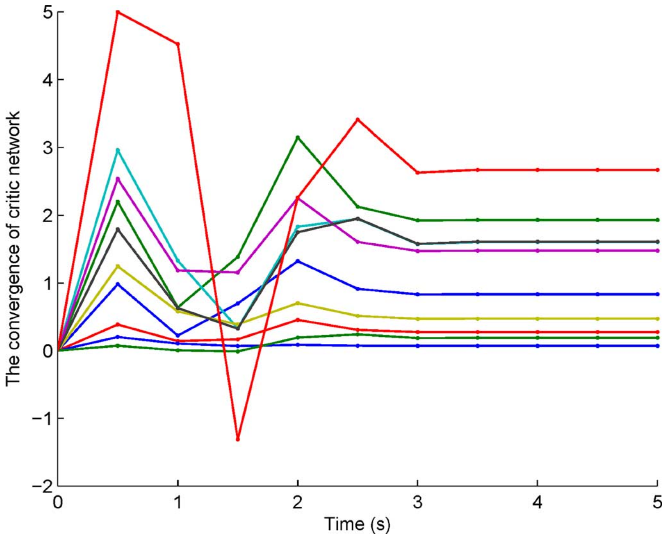

Fig. 2. Convergence of the game value function matrix P_i.

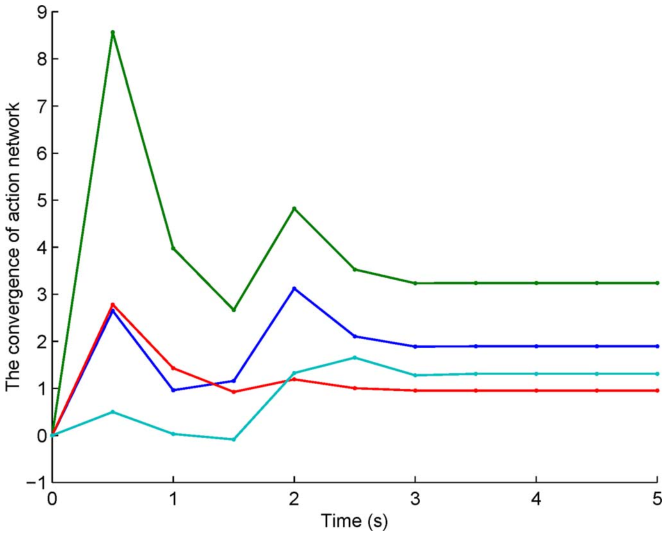

Fig. 3. Convergence of the control action network parameters K_i.

least squares problem is solved after 50 data samples are acquired, and thus the parameters of the control policy is updated every 0.5 s. The parameters of the critic network, the control action network and the disturbance action network are all initialized to zero. Similar to [44], the PE condition is ensured by adding small zero-mean Gaussian noises with variances to the control and disturbance inputs.

Fig. 2 presents the evolution of the parameters of the game value function during the learning process. It is clear that Algorithm 1 is convergent after ten iterative steps. The obtained approximate game value function is given by the matrix

$$P_{10} = \begin{bmatrix} 0.8335 & 0.9649 & 0.1379 & 0.8005 \\ 0.9649 & 1.4752 & 0.2359 & 0.8047 \\ 0.1379 & 0.2359 & 0.0696 & 0.0956 \\ 0.8005 & 0.8047 & 0.0956 & 2.6718 \end{bmatrix}$$

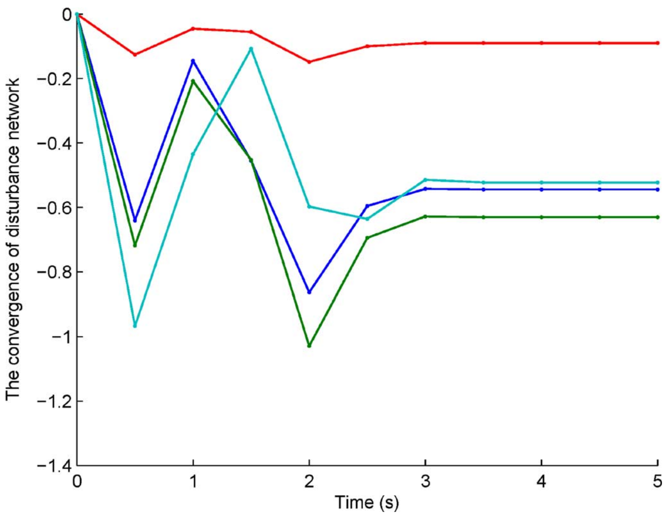

Fig. 4. Convergence of the disturbance action network parameters L_i.

and $\|P_{10} - P^*\| = 2.9375 \times 10^{-4}$. We can find that the solution obtained by the online model-free integral RL algorithm is quite close to the exact one obtained by solving GARE. Figs. 3 and 4 show the convergence process of the control and disturbance action network parameters. The obtained H_∞ state feedback control policy is $u_{11} = -[1.8941, 3.2397, 0.9563, 1.3126]x$.

V. Conclusion

In this paper, we developed an integral RL algorithm based on PI to solve online the Nash equilibrium solution for a two-player zero-sum differential game with completely unknown linear continuous-time dynamics. This led to a fully model-free method solving the GARE forward in time. The developed algorithm updates the value function, control and disturbance policies simultaneously. The convergence of the algorithm is demonstrated to be equivalent to Newton's method. One critic network and two action networks are used to approximate the game value function, control and disturbance policies, and the least squares method is given to estimate the unknown parameters. We demonstrate the effectiveness of the developed scheme by designing an H_∞ state feedback controller for a power system. In future research, we will extend the results to zero-sum and nonzero-sum differential games with completely unknown nonlinear continuous-time dynamics.

References

[1] P. J. Werbos, "Intelligence in the brain: A theory of how it works and how to build it," *Neural Netw.*, vol. 22, no. 3, pp. 200–212, Apr. 2009.

[2] F. Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 39–47, May 2009.

[3] F. L. Lewis and D. Vrabie, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Syst. Mag.*, vol. 32, no. 6, pp. 76–105, Dec. 2012.

[4] F. L. Lewis and D. Liu, *Approximate Dynamic Programming and Reinforcement Learning for Feedback Control*. Hoboken, NJ: Wiley, 2012.

[5] R. S. Sutton and A. G. Barto, *Reinforcement Learning : An Introduction*. Cambridge, MA: MIT Press, 1998.

[6] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 943–949, Aug. 2008.

[7] H. Zhang, Y. Luo, and D. Liu, "Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1490–1503, Sep. 2009.

[8] 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. 12, pp. 1854–1862, Dec. 2011.

[9] D. Wang, D. Liu, Q. Wei, D. Zhao, and N. Jin, "Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming," *Automatica*, vol. 48, no. 8, pp. 1825–1832, Aug. 2012.

[10] D. Liu, D. Wang, D. Zhao, Q. Wei, and N. Jin, "Neural-network-based optimal control for a class of unknown discrete-time nonlinear systems using globalized dual heuristic programming," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 3, pp. 628–634, Jul. 2012.

[11] H. Li and D. Liu, "Optimal control for discrete-time affine nonlinear systems using general value iteration," *IET Control Theory Appl.*, vol. 6, no. 18, pp. 2725–2736, Dec. 2012.

[12] Q. Wei and D. Liu, "An iterative ϵ-optimal control scheme for a class of discrete-time nonlinear systems with unfixed initial state," *Neural Netw.*, vol. 32, no. 6, pp. 236–244, Aug. 2012.

[13] D. Liu, D. Wang, and X. Yang, "An iterative adaptive dynamic programming algorithm for optimal control of unknown discrete-time nonlinear systems with constrained inputs," *Inf. Sci.*, vol. 220, pp. 331–342, Jan. 2013.

[14] D. Liu, H. Li, and D. Wang, "Neural-network-based zero-sum game for discrete-time nonlinear systems via iterative adaptive dynamic programming algorithm," *Neurocomputing*, vol. 110, pp. 92–100, June 2013.

[15] 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.

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

[17] D. Liu, Y. Zhang, and H. Zhang, "A self-learning call admission control scheme for CDMA cellular networks," *IEEE Trans. Neural Netw.*, vol. 16, no. 5, pp. 1219–1228, Sep. 2005.

[18] D. Liu, H. Javaherian, O. Kovalenko, and T. Huang, "Adaptive critic learning techniques for engine torque and air-fuel ratio control," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 988–993, Aug. 2008.

[19] G. G. Yen and P. G. Delima, "Improving the performance of globalized dual heuristic programming for fault tolerant control through an online learning supervisor," *IEEE Trans. Autom. Sci. Eng.*, vol. 2, no. 2, pp. 121–131, Apr. 2005.

[20] R. Ganesan, T. K. Das, and K. M. Ramachandran, "A multiresolution analysis-assisted reinforcement learning approach to run-by-run control," *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 2, pp. 182–193, Apr. 2007.

[21] W. S. Lin and J. W. Sheu, "Optimization of train regulation and energy usage of metro lines using an adaptive-optimal-control algorithm," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 4, pp. 855–864, Oct. 2011.

[22] S. K. Pradhan and B. Subudhi, "Real-time adaptive control of a flexible manipulator using reinforcement learning," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 2, pp. 237–249, Apr. 2012.

[23] Q. Kang, M. Zhou, J. An, and Q. Wu, "Swarm intelligence approaches to optimal power flow problem with distributed generator failures in power networks," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 2, pp. 343–353, Apr. 2013.

[24] B. Sun, P. B. Luh, Q. Jia, Z. Jiang, F. Wang, and C. Song, "Building energy management: Integrated control of active and passive heating, cooling, lighting, shading, and ventilation systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 3, pp. 588–602, Jul. 2013.

[25] Z. Xu, Q. Jia, H. Guan, and J. Shen, "Smart management of multiple energy systems in automotive painting shop," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 3, pp. 603–614, Jul. 2013.

[26] 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, Feb. 2012.

[27] Z. Ni, H. He, and J. Wen, "Adaptive learning in tracking control based on the dual critic network design," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 6, pp. 913–928, Jun. 2013.

[28] K. Doya, "Reinforcement learning in continuous time and space," *Neural Comput.*, vol. 12, no. 1, pp. 219–245, 2000.

[29] 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, May 2010.

[30] H. Zhang, L. Cui, X. Zhang, and Y. Luo, "Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2226–2236, Dec. 2011.

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

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

[33] D. Vrabie and F. L. Lewis, "Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems," *Neural Netw.*, vol. 22, no. 3, pp. 237–246, Apr. 2009.

[34] P. Mehta and S. Meyn, "Q-learning and pontryagins minimum principle," in *Proc. IEEE Conf. Decision Control*, Shanghai, China, Dec. 2009, pp. 3598–3605.

[35] J. Y. Lee, J. B. Park, and Y. H. Choi, "Integral Q-learning and explorized policy iteration for adaptive optimal control of continuous-time linear systems," *Automatica*, vol. 48, no. 11, pp. 2850–2859, Nov. 2012.

[36] 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, Oct. 2012.

[37] T. Basar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, 2nd ed. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1999.

[38] T. Basar and P. Bernhard, *H_∞ Optimal Conrol and Related Minimax Design Problems: A Dynamic Game Approach*, 2nd ed. Boston, MA, USA: Birkhäuser, 1995.

[39] A. Al-Tamimi, M. Abu-Khalaf, and F. L. Lewis, "Adaptive critic designs for discrete-time zero-sum games with application to H_∞ control," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 1, pp. 240–247, Feb. 2007.

[40] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Model-free Q-learning designs for linear discrete-time zero-sum games with application to H_∞ control," *Automatica*, vol. 43, no. 3, pp. 473–481, Mar. 2007.

[41] J.-H. Kim and F. L. Lewis, "Model-free H_∞ control design for unknown linear discrete-time systems via Q-learning with LMI," *Automatica*, vol. 46, no. 8, pp. 1320–1326, Aug. 2010.

[42] M. Abu-Khalaf, F. L. Lewis, and J. Huang, "Policy iterations and the Hamilton-Jacobi-Isaacs equation for H_∞ state feedback control with input saturation," *IEEE Trans. Autom. Control*, vol. 51, no. 12, pp. 1989–1995, Dec. 2006.

[43] M. Abu-Khalaf, F. L. Lewis, and J. Huang, "Neurodynamic progarmming and zero-sum games for constrained control systems," *IEEE Trans. Neural Netw.*, vol. 19, no. 7, pp. 1243–1252, Jul. 2008.

[44] H. Zhang, Q. Wei, and D. Liu, "An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games," *Automatica*, vol. 47, no. 1, pp. 207–214, Jan. 2011.

[45] K. G. Vamvoudakis and F. L. Lewis, "Online solution of nonlinear two-player zero-sum games using synchronous policy iteration," *Int. J. Robust Nonlinear Control*, vol. 22, no. 13, pp. 1460–1483, 2011.

[46] T. Dierks and S. Jagannathan, "Optimal control of affine nonlinear continuous-time systems using an online Hamilton-Jacobi-Isaacs formulation," in *Proc. IEEE Conf. Decision Control*, Atlanta, GA, USA, Dec. 2010, pp. 3048–3053.

[47] H. 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.

[48] D. Varbie and F. L. Lewis, "Adaptive dynamic programming for online solution of a zero-sum differential game," *J. Control Theory Appl.*, vol. 9, no. 3, pp. 353–360, 2011.

[49] H. Wu and B. Luo, "Simultaneous policy update algorithms for learning the solution of linear continuous-time H_∞ state feedback control," *Inf. Sci.*, vol. 222, no. 10, pp. 472–485, Feb. 2013.

Hongliang Li (S'13) received the B.S. degree in mechanical engineering and automation from the Beijing University of Posts and Telecommunications, Beijing, China, in 2010. He is currently working toward the Ph.D. degree at the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing.

He is also with the University of Chinese Academy of Sciences, Beijing, China. His current research interests include machine learning, neural networks, adaptive dynamic programming, optimization and game theory.

Derong Liu (S'91–M'94–SM'96–F'05) received the Ph.D. degree in electrical engineering from the University of Notre Dame, Notre Dame, IN, USA, in 1994.

He was a Staff Fellow with the General Motors Research and Development Center, Warren, MI, USA, from 1993 to 1995. He was an Assistant Professor with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, USA, from 1995 to 1999. He joined the University of Illinois at Chicago, Chicago, IL, USA, in 1999, and became a Full Professor of Electrical and Computer Engineering and of Computer Science in 2006. He was selected for the 100 Talents Program by the Chinese Academy of Sciences in 2008. He has published 14 books (six research monographs and eight edited volumes).

Dr. Liu is a Fellow of the INNS. He is currently the Editor-in-Chief of the IEEE Transactions on Neural Networks and Learning Systems. He received the Michael J. Birck Fellowship from the University of Notre Dame in 1990, the Harvey N. Davis Distinguished Teaching Award from the Stevens Institute of Technology in 1997, the Faculty Early Career Development CAREER Award from the National Science Foundation in 1999, the University Scholar Award from the University of Illinois from 2006 to 2009, and the Overseas Outstanding Young Scholar Award from the National Natural Science Foundation of China in 2008.

Ding Wang received the B.S. degree in mathematics from Zhengzhou University of Light Industry, Zhengzhou, China, the M.S. degree in operational research and cybernetics from Northeastern University, Shenyang, China, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2007, 2009, and 2012, respectively.

He is currently an Assistant Professor with The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His current research interests include adaptive dynamic programming, neural networks and learning systems, and complex systems and intelligent control.