Neurocomputing 238 (2017) 377–386

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Data-driven adaptive dynamic programming for continuous-time fully cooperative games with partially constrained inputs

Qichao Zhang[a,b], Dongbin Zhao[a,b,*], Yuanheng Zhu[a,b]

[a] The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[b] The University of Chinese Academy of Sciences, Beijing 100049, China

ARTICLE INFO

Article history:
Received 15 July 2016
Revised 20 January 2017
Accepted 23 January 2017
Available online 8 February 2017

Communicated by Prof. H. Zhang

Keywords:
Adaptive dynamic programming
Optimal control
Neural network
Fully cooperative games
Data-driven
Constrained input

ABSTRACT

In this paper, the fully cooperative game with partially constrained inputs in the continuous-time Markov decision process environment is investigated using a novel data-driven adaptive dynamic programming method. First, the model-based policy iteration algorithm with one iteration loop is proposed, where the knowledge of system dynamics is required. Then, it is proved that the iteration sequences of value functions and control policies can converge to the optimal ones. In order to relax the exact knowledge of the system dynamics, a model-free iterative equation is derived based on the model-based algorithm and the integral reinforcement learning. Furthermore, a data-driven adaptive dynamic programming is developed to solve the model-free equation using generated system data. From the theoretical analysis, we prove that this model-free iterative equation is equivalent to the model-based iterative equations, which means that the data-driven algorithm can approach the optimal value function and control policies. For the implementation purpose, three neural networks are constructed to approximate the solution of the model-free iteration equation using the off-policy learning scheme after the available system data is collected in the online measurement phase. Finally, two examples are provided to demonstrate the effectiveness of the proposed scheme.

1. Introduction

Recently, a newly developed technique, multi-agent reinforcement learning (MARL) which integrates the developments of reinforcement learning (RL) and game theory, has been widely applied to various fields including robotic control, traffic light control, battery management, distributed sensor network, etc [1–3]. In MARL, an agent is usually a computational entity which can perceive its environment, make decisions, and act upon its environment through actuators. Generally, the agent is not isolated but connected to its neighbour agents for multi-agent systems (MAS), and the mutual links between each agent can be expressed through a communication diagram. Their behaviors are adopted to optimize some performance indexes based on their own information and the shared one from their neighbors to affect the environment together. However, due to the complexity and variability of the environment, it is difficult to design the agents' behaviors relying on the prior knowledge. That is, it is necessary to learn appropriate behaviors for each agent.

For a single-agent environment, RL provides a method to learn to behave in an unknown or known environment. Through interactions with the environment, the agent adapts its behaviors continually based on a received reward signal, to finally achieve an optimal or near-optimal policy that maximizes the long-term accumulated reward. The accumulated reward is known as the value function [4]. In most cases, RL considers the Markov decision process (MDP). And some well-understood RL algorithms with good convergence such as Q-learning, Sarsa and adaptive dynamic programming (ADP) are proposed and widely used to tackle the single-agent RL task without full information of system dynamics [5–13].

For the multi-agent case, game theory provides a powerful tool to address most of challenges for RL in MAS, such that how to coordinate the agent's behaviors when the others change their strategies, how to guarantee the convergence properties of online algorithms in such a nonstationary environment, and so on. The generalization of the MDP for the multi-player case is the stochastic game [14,15], where each player can be regarded as an agent. It is well-known that many MARL algorithms such as team-Q [16], Minimax-Q [17], and Nash-Q [18] have been proposed for

* Corresponding author at: The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.
E-mail addresses: zhangqichao2013@163.com (Q. Zhang), dongbin.zhao@ia.ac.cn (D. Zhao), zyh7716155@163.com (Y. Zhu).

http://dx.doi.org/10.1016/j.neucom.2017.01.076
0925-2312/

the stochastic game. Relying on the roles and tasks of players, the stochastic game can be divided into several types. If the players have the same value function and coordinate to fulfill a common task, e.g., the wireless network agents in [19], the vehicle stability control in [20], and traffic signal light agents in [21], this kind of game is called a fully cooperative (FC) game. Otherwise, if the agents only pursue their own interest and compete with each other, this kind of game is known as zero-sum (ZS) game [22]. From the perspective of minimax optimization problem, the H_∞ control can be formulated as a two-player ZS game [23,24]. In some tasks, the agents are neither fully cooperative nor fully competitive, and these mixed games are called nonzero-sum (NZS) games. Different from the above centralized games, the distributed cooperative control of the multi-agent graphical games is investigated in [25,26], where the value iteration and Riccati design algorithms are used respectively. However, most of above mentioned methods are suitable for the discrete-time MDP environment.

For the continuous-time MDP environment, many model-based and model-free methods based on ADP have been applied to the ZS and NZS games [27–33], in which the knowledge of system dynamics is known or should be identified. For the multi-agent graphical games, the distributed synchronization controller based on model-based RL algorithms can also be found in [34,35] for linear and nonlinear continuous-time systems, respectively. For the partially unknown multi-agent graphical games, the RL-based distributed cooperative controller is designed for multi-input and multi-output nonlinear systems without the knowledge of internal system dynamics in [36]. It should be mentioned that the training of identifers which is used in above model-free methods is usually time-consuming and the introduced identification errors are usually adverse to find the optimal control policies. Vrabie and Lewis propose the integral reinforcement learning (IRL) to solve the ZS game without the knowledge of internal dynamics in [37]. Note that multiple iterative loops are required in the proposed algorithms, which is often time-consuming. Motivated by that, an online simultaneous policy update algorithm with only one iterative loop and an off-policy reinforcement learning for the partially unknown ZS game are developed in [38,39], respectively. In [40], the tracking HJI equation is solved by an actor-critic structure with only one neural network for the wheeled mobile robot without the knowledge of the system's drift dynamics. For completely unknown systems, an exciting work is given to solve the optimal control problem of uncertain nonlinear systems in [41]. A robust ADP is performed without any system dynamics and identification process. In [42], a model free approach is proposed based on IRL with safe explorations for the nonlinear optimal control problem. Then, this kind of data-based or data driven method is extended to the optimal control for linear ZS game in [43] and the H_∞ control problem in [44]. Unfortunately, the FC game with completely unknown dynamics in continuous-time MDP environment is rarely mentioned.

On the other hand, the mentioned ADP methods do not take into account the input constraints caused by actuator saturation. However, failure to account for actuator saturation often severely destroys the system performance, or may even lead to instability. In [45], embracing IRL and experience replay, an online ADP algorithm is proposed to design the optimal controller of partially-unknown constrained control systems. For ZS games with saturation constraints, Abu-Khalaf et al. [46] design a suitable quasi-norm controller of continuous-time nonlinear systems. In [47], a policy iteration (PI) algorithm with the actor-critic-disturbance structure is performed to solve the associated Hamilton–Jacobi–Isaacs (HJI) equation of constrained ZS game with a non-quadratic performance index. For NZS games, Yasini et al. [48] integrate concurrent learning with RL to solve the corresponding Hamilton-Jacobi (HJ) equations, where the restrictive persistence of excitation (PE) condition is relaxed. Unfortunately, the unknown nonlinear FC games with constrained input are rarely studied using data-driven ADP methods in the literatures.

In this paper, we focus on the continuous-time unknown FC game with partially constrained inputs using the data-driven iteration ADP method. The contribution of this paper emphasizes in two parts:

1. This paper extends the existing work in [39,44] for the H_∞ control problem to the continuous-time nonlinear FC games. An two-phase data-driven iterative ADP is proposed with the online measurement phase and off-policy learning phase, where the system dynamics and the identification procedure in [33,49] are neither required. It also differs from the work in [15,16], which focus on the FC games in discrete-time MAP environment.
2. In contrast to the existing data-driven method in [39,41–43], the proposed data-driven iterative ADP takes into account the situation of partially constrained input caused by part of the actuator saturation, which is much more difficult and complicated in some sense.

The rest of this paper is organized as follows: Section 2 introduces the problem formulation. In Section 3, a two-phase data-driven iteration ADP method that needs no system dynamics for the FC game with partially constrained inputs is given, and the convergence analysis is given. Simulation results and the conclusion are presented in Sections 4 and 5, respectively.

2. Preliminary

2.1. Problem statement

Consider the following two-player FC game system

$$\dot{x} = f(x) + g_1(x)u_1 + g_2(x)u_2 \tag{1}$$

where $x \in \Re^n$ is the state vector, $u_i = u_i(x) \in \Re^{m_i}$ is the control input for each player i, $f(x) \in \Re^n$ and $g_i(x) \in \Re^{n\times m_i}, i = 1, 2$, are continuous unknown nonlinear functions with $f(0) = 0$. Note that $u_2 = [u_2^1, \ldots, u_2^{m_2}]^{\mathrm{T}}$ is constrained by a positive constant, i.e., $|u_2^l| \le \lambda, l = 1, \ldots, m_2$, where λ is the actuator bound. Assumed that $f(x) + g_1(x)u_1 + g_2(x)u_2$ is Lipschitz continuous on a compact set $\Omega \in \Re^n$ containing the origin, and the system is stabilizable on Ω.

For general optimal control of the NZS games [29,33], the cost functions associated with each player are given by

$$\begin{aligned} J_i(x_0, u_1, u_2) &= \int_0^\infty r_i(x, u_1, u_2)\mathrm{d}t \\ &= \int_0^\infty \{x^{\mathrm{T}}Q_i x + \Upsilon_i(u_1) + \Upsilon_i(u_2)\}\mathrm{d}t, i = 1, 2 \end{aligned} \tag{2}$$

where $r_i(x, u_1, u_2)$ is the utility function, and $Q_i > 0$. In general, we choose $\Upsilon_i(u_1) = u_1^{\mathrm{T}}R_{i1}u_1$ and $\Upsilon_i(u_2) = u_2^{\mathrm{T}}R_{i2}u_2, i = 1, 2$ for the unconstrained control inputs.

Clearly, there is an individual cost function associated with all control inputs for each player of the NZS games, where each player wants to optimize his own performance index. However, for the optimal control of the FC games, all players take actions together as a team. They have precisely the same cost function, which is given by

$$\begin{aligned} J(x_0, u_1, u_2) &= \int_0^\infty r(x, u_1, u_2)\mathrm{d}t \\ &= \int_0^\infty \{x^{\mathrm{T}}Qx + U_1(u_1) + U_2(u_2)\}\mathrm{d}t \end{aligned} \tag{3}$$

where $r(x, u_1, u_2)$ is the utility function for the FC game (1), $U_1(u_1) = u_1^{\mathrm{T}}R_1u_1$ and $U_2(u_2) = u_2^{\mathrm{T}}R_2u_2$ for the unconstrained inputs u_1 and u_2.

Remark 1. If we choose $Q_1 = Q_2 = Q$, $R_{11} = R_{21} = R_1$ and $R_{21} = R_{22} = R_2$ for (2) and (3), the cost function (3) of the FC game is in line with (2) of the NZS game. In other words, the FC game can be regarded as a special case of the NZS game. In order to extend the data-driven ADP algorithm, the FC game is considered.

Remark 2. For the output feedback optimal control problem with the system output $y = h(x)$ [39] or $y = Cx$ [50], where $h(x)$ and C are a continuous function and a system matrix with appropriate dimensions, the cost function (3) is designed based on the system output accordingly. That is, the term $x^{\mathrm{T}}Qx$ in (3) is usually replaced by $y^{\mathrm{T}}Qy$. Then, the output feedback optimal control problem can be solved based on RL algorithms.

Motivated by [46], for the constrained control input u_2, a generalized non-quadratic functional $U_2(u_2) = 2\int_0^{u_2} \lambda \tanh^{-\mathrm{T}}(s/\lambda)R_2 \mathrm{d}s$ is employed. Note that Q, R_1 are positive definite matrices with appropriate dimensions and $R_2 = \mathrm{diag}(\gamma_1, \ldots, \gamma_{m_2}) > 0$ for simplicity of analysis.

Assume that the system (1) is zero-state observable. The goal of the two-player FC game is to find the feedback control policies $\{u_1(x), u_2(x)\}$ that minimizes the following common value function defined as

$$V(x, u_1, u_2) \triangleq \int_t^{\infty} \left\{ x^{\mathrm{T}}Qx + u_1^{\mathrm{T}}R_1u_1 + 2\int_0^{u_2} \lambda \tanh^{-\mathrm{T}}(s/\lambda)R_2 \mathrm{d}s \right\} \mathrm{d}\tau \tag{4}$$

Definition 1. [(Admissible control):] A feedback control policy pair $\mu = \{\mu_1, \mu_2\}$ is defined as admissible to (4) on the compact set Ω, denoted by $\mu_i \in \Psi_i(\Omega), i = 1, 2$, if $\mu_i(x)$ is continuous on Ω with $\mu_i(0) = 0$, μ stabilizes the system (1) on Ω, and the value function (4) is finite $\forall x_0 \in \Omega$.

Thus, for given system (1) with admissible control inputs u_1 and u_2, we aim to find the so-called coordination equilibria $\{u_1^*, u_2^*\}$ which gains the optimal value function

$$V^*(x) \triangleq V(u_1^*, u_2^*) = \min_{u_1, u_2} V(u_1, u_2)$$

which means

$$V(u_1^*, u_2^*) \le \min\{V(u_1^*, u_2), V(u_1, u_2^*)\}$$

In other words, no player has any incentive to change his policy from the coordination equilibria, so the best policy for one player is the best one for the other [16]. Let $V^*(x) \in C^1(\Omega)$. $C^1(\Omega)$ denotes a function space on Ω with first derivatives are continuous.

A differential equivalent to the value function in (4) is a Lyapunov-like equation

$$r(x, u_1, u_2) + \nabla V^{\mathrm{T}}(f + g_1u_1 + g_2u_2) = 0, V(0) = 0 \tag{5}$$

where $\nabla V = \partial V/\partial x$. Define the Hamiltonian function

$$H(x, \nabla V, u_1, u_2) \triangleq r(x, u_1, u_2) + \nabla V^{\mathrm{T}}(f + g_1u_1 + g_2u_2)$$

Furthermore, applying the stationary conditions $\partial H(x, \nabla V^*, u_1, u_2)/\partial u_i = 0$, we can obtain the optimal control policies

$$u_1^*(x) = -\frac{1}{2}R_1^{-1}g_1^{\mathrm{T}}(x)\nabla V^*(x) \tag{6}$$

$$u_2^*(x) = -\lambda \tanh\left((1/2\lambda)R_2^{-1}g_2^{\mathrm{T}}(x)\nabla V^*(x)\right) \tag{7}$$

Based on (6) and (7), the Lyapunov-like Eq. (5) can be presented as

$$0 = x^{\mathrm{T}}Qx - \frac{1}{4}(\nabla V^*(x))^{\mathrm{T}}g_1R_1^{-1}g_1^{\mathrm{T}}\nabla V^*(x) + (\nabla V^*(x))^{\mathrm{T}}f + \lambda^2\bar{R}_2 \ln(\underline{1} - \tanh^2(D^*)) \tag{8}$$

where $\bar{R}_2 = [s_1, \ldots, s_{m_2}] \in \Re^{1\times m_2}$, $D^* = (1/2\lambda)R_2^{-1}g_2^{\mathrm{T}}(x)\nabla V^*(x)$, and $\underline{1}$ is a column vector with all elements equal to one.

2.2. Model-based PI method

To obtain the optimal control policies (6) and (7), we have to solve the Lyapunov-like Eq. (8). However, it is difficult to give an analytic solution of (8) due to the inherent nonlinearity. Motivated by the online PI algorithm with one iteration loop for the NZS games in [28], a model-based iterative method is given to approach the solution of (8).

1. *(Policy evaluation)* given an initial admissible control policy pair $\{u_1^0(x), u_2^0(x)\}$, find $V^k(x)$ successively approximated by solving the following equation with $V^k(0) = 0$

$$r(x, u_1^k, u_2^k) + (\nabla V^{k+1})^{\mathrm{T}}(f + g_1u_1^k + g_2u_2^k) = 0, k = 0, 1, \ldots, \tag{9}$$

2. *(Policy improvement)* update the control policies simultaneously by

$$\begin{aligned} u_1^{k+1}(x) &= -\frac{1}{2}R_1^{-1}g_1^{\mathrm{T}}(x)\nabla V^{k+1}(x) \\ u_2^{k+1}(x) &= -\lambda \tanh\left(D^{k+1}\right) \end{aligned} \tag{10}$$

where $D^{k+1} = (1/2\lambda)R_2^{-1}g_2^{\mathrm{T}}(x)\nabla V^{k+1}(x)$, and k is the iterative index.

Theorem 1. *Let $V^k(x) \in C^1(\Omega)$, $V^k(x) \ge 0$, $V^k(0) = 0$ and $u_i^k(x) \in \Psi_i(\Omega), i = 1, 2$. If $(V^{k+1}(x), u_i^k(x))$ and $(V^{k+2}(x), u_i^{k+1}(x))$ both satisfy the Lyapunov-like Eq. (5) with the boundary condition $V^{k+1}(0) = 0, V^{k+2}(0) = 0$, then we have*

1. *the obtained control policies $u_i^{k+1}(x), i = 1, 2$ in (10) are admissible for system (1) on Ω;*
2. *$V^*(x) \le V^{k+2}(x) \le V^{k+1}(x), \forall x \in \Omega$;*
3. *$\lim_{k\to\infty} V^k(x) = V^*(x)$;*
4. *$\lim_{k\to\infty} u_i^k(x) = u_i^*(x), i = 1, 2$.*

Proof. For the first part of Theorem 1, taking the derivative of $V^{k+1}(x)$ with respect to time along the system $f + g_1u_1^{k+1} + g_2u_2^{k+1}$ trajectory, we have

$$\dot{V}^{k+1} \triangleq (\nabla V^{k+1})^{\mathrm{T}}f + (\nabla V^{k+1})^{\mathrm{T}}g_1u_1^{k+1} + (\nabla V^{k+1})^{\mathrm{T}}g_2u_2^{k+1} \tag{11}$$

Based on (9) and (10), we can get

$$\begin{aligned} (\nabla V^{k+1})^{\mathrm{T}}f &= -x^{\mathrm{T}}Qx - (u_1^k)^{\mathrm{T}}R_1u_1^k - (\nabla V^{k+1})^{\mathrm{T}}g_1u_1^k \\ &\quad - 2\int_0^{u_2^k} \lambda \tanh^{-\mathrm{T}}(s/\lambda)R_2 \mathrm{d}s - (\nabla V^{k+1})^{\mathrm{T}}g_2u_2^k \\ (\nabla V^{k+1})^{\mathrm{T}}g_1 &= -2(u_1^{k+1})^{\mathrm{T}}R_1 \\ (\nabla V^{k+1})^{\mathrm{T}}g_2 &= -2\lambda \tanh^{-\mathrm{T}}(u_2^{k+1}/\lambda)R_2 \end{aligned} \tag{12}$$

Substitute into (11)

$$\begin{aligned} \dot{V}^{k+1} &= -x^{\mathrm{T}}Qx - (u_1^k)^{\mathrm{T}}R_1u_1^k + 2(u_1^{k+1})^{\mathrm{T}}R_1u_1^k - 2(u_1^{k+1})^{\mathrm{T}}R_1 \\ &\quad \times u_1^{k+1} + 2[\lambda \tanh^{-\mathrm{T}}(u_2^{k+1}/\lambda)R_2(u_2^k - u_2^{k+1}) \\ &\quad - \int_0^{u_2^k} \lambda \tanh^{-\mathrm{T}}(s/\lambda)R_2 \mathrm{d}s] \\ &= -x^{\mathrm{T}}Qx - \|\theta(u_1^{k+1} - u_1^k)\|^2 - (u_1^{k+1})^{\mathrm{T}}R_1u_1^{k+1} \\ &\quad + 2\left[\varrho^{\mathrm{T}}(u_2^{k+1})R_2(u_2^k - u_2^{k+1}) - \int_0^{u_2^k} \varrho^{\mathrm{T}}(s)R_2 \mathrm{d}s\right] \end{aligned}$$

where $R_1 = \theta^{\mathrm{T}}\theta$ and $\varrho^{\mathrm{T}}(u_2^k) = \lambda \tanh^{-\mathrm{T}}(u_2^k/\lambda)$.

Based on $R_2 = \mathrm{diag}(\gamma_1, \ldots, \gamma_{m_2}) > 0$, we have

$$\dot{V}^{k+1} = -x^{\mathrm{T}}Qx - \|\theta^{T}(u_1^{k+1} - u_1^k)\|^2 - (u_1^{k+1})^{\mathrm{T}}R_1u_1^{k+1} + 2\sum_{\zeta=1}^{m_2}\gamma_\zeta\left[\varrho(u_{2,\zeta}^{k+1})(u_{2,\zeta}^k - u_{2,\zeta}^{k+1}) - \int_0^{u_{2,\zeta}^k}\varrho(s_\zeta)\mathrm{d}s_\zeta\right]$$

where $u_2^k = [u_{2,1}^k, \ldots, u_{2,m_2}^k]^{\mathrm{T}}$.

Since $\tanh(\cdot)$ is a monotonic odd function, $\varrho^{\mathrm{T}}(u_2^k) = \tanh^{-\mathrm{T}}(u_2^k/\lambda)$ is monotone and odd. So does $\varrho(u_{2,\zeta}^k)$. Then, we have the term $\varrho(u_{2,\zeta}^{k+1})(u_{2,\zeta}^k - u_{2,\zeta}^{k+1}) - \int_0^{u_{2,\zeta}^k}\varrho(s_\zeta)\mathrm{d}s_\zeta$ is always negative according to the geometrical meaning. This implies that $\dot{V}^k(x+1) < 0$ and $V^{k+1}(x)$ is a Lyapunov function for $u_i^{k+1}, i = 1, 2$ on Ω. Since the nonlinear functions g_1, g_2 are continuous and $V^{k+1}(0) = 0$, then we have u_i^{k+1} in (10) is continuous with $u_i^{k+1}(0) = 0$. According to the Definition 1, the control policies $u_i^{k+1}(x)$ are admissible for (1) on Ω.

For the second part of Theorem 1, considering $V(x)$ defined in (4) along the system $f + g_1u_1^{k+1} + g_2u_2^{k+1}$ trajectory, we have

$$V^{k+2}(x) - V^{k+1}(x) = -\int_t^\infty\left\{\frac{\partial(V^{k+2} - V^{k+1})^{\mathrm{T}}}{\partial x}(f + g_1u_1^{k+1} + g_2u_2^{k+1})\right\}\mathrm{d}\tau \tag{13}$$

Since $(V^{k+2}(x), u_i^{k+1}(x))$ satisfies (9), we can obtain

$$(\nabla V^{k+2})^{\mathrm{T}}f = -x^{\mathrm{T}}Qx - (u_1^{k+1})^{\mathrm{T}}R_1u_1^{k+1} - (\nabla V^{k+2})^{\mathrm{T}}g_1u_1^{k+1} - 2\int_0^{u_2^{k+1}}\lambda\tanh^{-\mathrm{T}}(s/\lambda)R_2\mathrm{d}s - (\nabla V^{k+2})^{\mathrm{T}}g_2u_2^{k+1} \tag{14}$$

Substituting (12) and (14) into (13), we can get

$$\begin{aligned}V^{k+2}(x) - V^{k+1}(x) &= -\int_t^\infty\left\{(u_1^{k+1})^{\mathrm{T}}R_1u_1^{k+1} - 2(u_1^{k+1})^{\mathrm{T}}R_1u_1^k + (u_1^k)^{\mathrm{T}}R_1u_1^k - 2\left[\varrho^{\mathrm{T}}(u_2^{k+1})R_2(u_2^{k+1} - u_2^k) - \int_{u_2^k}^{u_2^{k+1}}\varrho^{\mathrm{T}}(s)R_2\mathrm{d}s\right]\right\}\mathrm{d}\tau \\ &= -\int_t^\infty\left\{\|\theta(u_1^{k+1} - u_1^k)\|^2 - 2\left[\varrho^{\mathrm{T}}(u_2^{k+1})R_2(u_2^{k+1} - u_2^k) - \int_{u_2^k}^{u_2^{k+1}}\varrho^{\mathrm{T}}(s)R_2\mathrm{d}s\right]\right\}\mathrm{d}\tau\end{aligned} \tag{15}$$

Since $\varrho^{\mathrm{T}}(u_2^k)$ is monotone and odd, we can deduce that $V^{k+2}(x) - V^{k+1}(x) \le 0$. Furthermore, we have $V^*(x) \le V^{k+2}(x)$ by contradiction.

Because $\{V^k\}_{k=0}^\infty$ is a monotonically decreasing sequence with the lower bounded $V^*(x)$, then V^k converges pointwise to V^∞. Because of the uniqueness of $V(x)$ with $x \in \Omega$ [51], we can get that $V^\infty = V^*$, which means that $\lim_{k\to\infty} V^k(x) = V^*(x)$. According to (10), it can be deduced that $\lim_{k\to\infty} u_i^{k+1}(x) = u_i^*(x), i = 1, 2$. The proof is completed. □

Remark 3. This method can be seen as an extension of the iterative method to solve the constrained HJB equation in [52]. Note that this iterative method with only one iteration loop is a model-based PI which involves two steps: policy evaluation by (9) and policy improvement by (10). It can be seen that the complete knowledge of system dynamics is required.

3. Data-driven constrained optimal control

Since the system dynamics are unknown for the partially constrained optimal control problem of FC games under consideration, a two-phase data-driven iterative ADP method is given using generated system data rather than accurate system dynamics. In this paper, the off-policy learning scheme used in [39,53] is adopted.

3.1. Data-driven iterative ADP

In this section, the data-driven iterative ADP approach is derived based on the model-based PI method and IRL. Given two arbitrary admissible control policies $u_i' \in \Psi_i(\Omega), i = 1, 2$ which can stabilize the system (1) on the compact set Ω, the derivative of $V^{k+1}(x)$ with respect to time for the $\{k+1\}$th iteration equals $\mathrm{d}V^{k+1}/\mathrm{d}t = (\nabla V^{k+1})^{\mathrm{T}}(f + g_1u_1' + g_2u_2')$. Based on (9) and (10), we have

$$\begin{aligned}\frac{\mathrm{d}V^{k+1}}{\mathrm{d}t} &= (\nabla V^{k+1})^{\mathrm{T}}\big(g_1(u_1' - u_1^k) + g_2(u_2' - u_2^k)\big) - r(x, u_1^k, u_2^k) \\ &= -2(u_1^{k+1})^{\mathrm{T}}R_1(u_1' - u_1^k) + 2\lambda(D^{k+1})^{\mathrm{T}}R_2(u_2' - u_2^k) - r(x, u_1^k, u_2^k)\end{aligned} \tag{16}$$

According to IRL, integrating both sides of (16) on the interval $[t, t + \Delta t]$, the following equation is formulated as

$$V^{k+1}(x(t)) - V^{k+1}(x(t+\Delta t)) - \int_t^{t+\Delta t}2(u_1^{k+1})^{\mathrm{T}}R_1(u_1' - u_1^k)\mathrm{d}\tau + \int_t^{t+\Delta t}2\lambda(D^{k+1})^{\mathrm{T}}R_2(u_2' - u_2^k)\mathrm{d}\tau = \int_t^{t+\Delta t}r(x, u_1^k, u_2^k)\mathrm{d}\tau \tag{17}$$

where V^{k+1} is an unknown function and u_1^{k+1}, D^{k+1} are unknown function vectors to be solved.

Remark 4. The main idea of the data-driven iterative ADP is to solve the model-free Eq. (17) instead of the model-based iterative Eqs. (9) and (10). Note that the system dynamics, i.e., $f(x)$ and $g_i(x), i = 1, 2$, are not required in the iterative Eq. (17). Instead, the available system data, i.e., u_1^k and u_2^k, are utilized during the iteration process. In fact, the knowledge of system dynamics is embedded in the available system data. Therefore, an online measurement phase to collect the available system data is required before the off-policy learning scheme which is used to approach the solution of Eq. (17).

Thus, the unknown function $(V^{k+1}, u_1^{k+1}, D^{k+1})$ is iterated following (17). Motivated by [39], as the iteration step k increases, the convergence of the generated solution sequence $\{(V^k, u_1^{k+1}, D^{k+1})\}$ to the optimal one is proved as follows.

Theorem 2. *Let $V^{k+1}(x) \in C^1(\Omega), V^{k+1}(x) \ge 0$, $V^{k+1}(0) = 0$ and $u_i^{k+1}(x) \in \Psi_i(\Omega), i = 1, 2$. Then $(V^{k+1}, u_1^{k+1}, D^{k+1})$ is the solution of (17) for $\forall u_i' \in \Psi_i(\Omega), i = 1, 2$ if and only if it is a solution of the model-based iterative Eqs. (9) and (10).*

Proof. If we can prove $(V^{k+1}, u_1^{k+1}, D^{k+1})$ is a unique solution of (17) for $\forall u_i' \in \Psi_i(\Omega), i = 1, 2$, the model-free Eq. (17) is equivalent to the model-based iterative Eqs. (9) and (10). The proof is by contradiction.

From the derivation of (17), we can conclude that the solution $(V^{k+1}, u_1^{k+1}, D^{k+1})$ of (17) also satisfies the Eqs. (9)–(16). Suppose that there is another solution $(\hbar_V(x), \hbar_{u_1}(x), \hbar_D(x))$ of (17) with $\hbar_V(0) = 0, \hbar_{u_1} \in \Psi_1(\Omega)$ and $\hbar_{u_2} = -\lambda\tanh(\hbar_D(x)) \in \Psi_2(\Omega)$. Thus, $(\hbar_V(x), \hbar_{u_1}(x), \hbar_D(x))$ also satisfies (16), i.e.,

$$\frac{\mathrm{d}\hbar_V}{\mathrm{d}t} = -2(\hbar_{u_1})^{\mathrm{T}}R_1(u_1' - u_1^k) + 2\lambda(\hbar_D)^{\mathrm{T}}R_2(u_2' - u_2^k) - r(x, u_1^k, u_2^k) \tag{18}$$

Substituting (18) from (16), we have

$$\frac{\mathrm{d}}{\mathrm{d}t}(V^{k+1} - \hbar_V) = -2(u_1^{k+1} - \hbar_{u_1})^{\mathrm{T}}R_1(u_1' - u_1^k) + 2\lambda(D^{k+1} - \hbar_D)^{\mathrm{T}}R_2(u_2' - u_2^k) \tag{19}$$

for $\forall u_i' \in \Psi_i(\Omega)$. If we choose the admissible control policies $u_1' = u_1^k$ and $u_2' = u_2^k = -\lambda \tanh(D^k)$, then we can obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}(V^{k+1} - \hbar_V) = 0 \tag{20}$$

which means that the value of $V^{k+1} - \hbar_V$ is a real constant for $\forall x \in \Omega$. According to the boundary conditions $V^{k+1}(0) = 0$, $\hbar_V(0) = 0$ and (20), we can deduce that $V^{k+1} - \hbar_V = 0$, i.e., $V^{k+1} = \hbar_V$ for $\forall x \in \Omega$.

Combined (19) and (20), we can deduce

$$(u_1^{k+1} - \hbar_{u_1})^{\mathrm{T}} R_1 (u_1' - u_1^k) = \lambda (D^{k+1} - \hbar_D)^{\mathrm{T}} R_2 (u_2' - u_2^k) \tag{21}$$

holds for $\forall u_i' \in \Psi_i(\Omega)$.

Now, we want to prove that Eq. (21) is set up for $\forall u_i' \in \Psi_i(\Omega)$ if and only if $u_1^{k+1} = \hbar_{u_1}$ and $D^{k+1} = \hbar_D$ are satisfied simultaneously.

First, we should prove that if $u_1^{k+1} = \hbar_{u_1}$ and $D^{k+1} = \hbar_D$ are satisfied simultaneously, Eq. (21) is established for $\forall u_i' \in \Psi_i(\Omega)$. Note that the admissible control policies u_1' and u_2' in (21) can be chosen arbitrarily in $\Psi_1(\Omega)$ and $\Psi_2(\Omega)$. If we choose $u_1' = u_1^k \in \Psi_1(\Omega)$, then

$$0 = \lambda (D^{k+1} - \hbar_D)^{\mathrm{T}} R_2 (u_2' - u_2^k)$$

holds for $\forall u_2' \in \Psi_2(\Omega)$. That is, $D^{k+1} = \hbar_D$ holds for $\forall x \in \Omega$. In a similar way, it can be acquired that $u_1^{k+1} = \hbar_{u_1}$ holds for $\forall x \in \Omega$. In fact, if $u_1^{k+1} = \hbar_{u_1}$ and $D^{k+1} = \hbar_D$ hold, Eq. (21) is always equal to zero for $\forall u_i' \in \Psi_i(\Omega)$.

Then, we should prove that Eq. (21) is not established for $\forall u_i' \in \Psi_i(\Omega)$ for the other cases in addition to $u_1^{k+1} = \hbar_{u_1}$ and $D^{k+1} = \hbar_D$, i.e., $u_1^{k+1} = \hbar_{u_1}$ and $D^{k+1} \neq \hbar_D$, $u_1^{k+1} \neq \hbar_{u_1}$ and $D^{k+1} = \hbar_D$, $u_1^{k+1} \neq \hbar_{u_1}$ and $D^{k+1} \neq \hbar_D$. Note that Eq. (21) is established for $\forall u_i' \in \Psi_i(\Omega)$ means that it should be satisfied for any $u_i' \in \Psi_i(\Omega)$.

If $u_1^{k+1} = \hbar_{u_1}$ and $D^{k+1} \neq \hbar_D$ hold, Eq. (21) is not established obviously for $u_2' \neq u_2^k, u_2' \in \Psi_2(\Omega)$.

If $D^{k+1} \neq \hbar_D$, $u_1^{k+1} \neq \hbar_{u_1}$ hold, Eq. (21) is not established obviously for $u_1' \neq u_1^k, u_1' \in \Psi_1(\Omega)$.

If $u_1^{k+1} \neq \hbar_{u_1}$ and $D^{k+1} \neq \hbar_D$ hold, Eq. (21) is no longer valid for $u_1' = u_1^k$ and $u_2' \neq u_2^k$ or for $u_1' \neq u_1^k$ and $u_2' = u_2^k$.

Hence, we can conclude that Eq. (21) is set up for $\forall u_i' \in \Psi_i(\Omega)$ if and only if $u_1^{k+1} = \hbar_{u_1}$ and $D^{k+1} = \hbar_D$ hold. That is to say, $(V^k, u_1^{k+1}, D^{k+1})$ is a unique solution of (17) for $\forall u_i' \in \Psi_i(\Omega), i = 1, 2$. The proof is completed. □

Remark 5. According to Theorem 2, we know that (17) is equivalent to the iterative Eqs. (9) and (10). Since the solution V^k of (9)–(10) can converge to the optimal one V^*, we can say the data-driven iterative ADP can also approach the solution V^* of Eq. (8). It differs from the off-policy RL in [39,44], which concentrates on the two-player ZS game. In addition, a complicated situation that partial control inputs are constrained is considered in this paper.

Remark 6. For simplicity of analysis, a two-player FC game is considered in this paper. In fact, for multi-player FC game with partially constrained inputs, all the constrained inputs can be integrated into a constrained input vector while the others can be integrated in a unconstrained input vector. Then the system dynamics of multi-player FC game can be expressed as the formula (1) in this paper. Hence, this data-driven iterative ADP can be extended to multi-player FC game in theory.

3.2. Implementation based on neural network

For implementation purpose, neural networks are constructed to approach the solution of (17). According to the Weierstrass high-order approximation theorem [54], the solution $\left(V_{k+1}(x), u_1^{k+1}(x), D^{k+1}(x)\right)$ of (17) based on NNs can be written as

$$\begin{aligned} V^{k+1}(x) &= w_{c,k+1}^{\mathrm{T}} \phi_c(x) + \varepsilon_{c,k+1} \\ u_1^{k+1}(x) &= w_{u,k+1}^{\mathrm{T}} \phi_u(x) + \varepsilon_{u,k+1} \\ D^{k+1}(x) &= w_{d,k+1}^{\mathrm{T}} \phi_d(x) + \varepsilon_{d,k+1} \end{aligned}$$

where $\phi_c : \Re^n \to \Re^{K_c}$, $\phi_u : \Re^n \to \Re^{K_u}$ and $\phi_d : \Re^n \to \Re^{K_d}$ are linearly independent basis function vectors, $w_{c,k+1} \in \Re^{K_c}$, $w_{u,k+1} \in \Re^{K_u \times m_1}$ and $w_{d,k+1} \in \Re^{K_d \times m_2}$ are the unknown coefficient vector and matrices with K_c, K_u and K_d the numbers of hidden neurons, $\varepsilon_{c,k+1}$, $\varepsilon_{u,k+1}$ and $\varepsilon_{d,k+1}$ are the reconstruction errors with appropriate dimensions for $\forall k$. It is shown in [52] that as $K_c \to \infty$, $K_u \to \infty$ and $K_d \to \infty$, the reconstruction errors $\varepsilon_{c,k+1}$, $\varepsilon_{u,k+1}$ and $\varepsilon_{d,k+1}$ converge to zero.

Let $\hat{w}_{c,k+1}, \hat{w}_{u,k+1}$ and $\hat{w}_{d,k+1}$ be the estimations of the unknown coefficients $w_{c,k+1}, w_{u,k+1}$ and $w_{d,k+1}$, respectively. Then the actual output of the three NNs can be presented as

$$\begin{aligned} \hat{V}^{k+1}(x) &= \hat{w}_{c,k+1}^{\mathrm{T}} \phi_c(x) \\ \hat{u}_1^{k+1}(x) &= \hat{w}_{u,k+1}^{\mathrm{T}} \phi_u(x) \\ \hat{D}^{k+1}(x) &= \hat{w}_{d,k+1}^{\mathrm{T}} \phi_d(x) \end{aligned} \tag{22}$$

Define a strictly increasing time sequence $\{t_j\}_{j=0}^q$ for a large interval with the number of collected data points $q > 0$. Using $(\hat{V}^{k+1}(x), \hat{u}_1^{k+1}(x), \hat{D}^{k+1}(x))$ instead of $(V^{k+1}(x), u_1^{k+1}(x), D^{k+1}(x))$ in Eq. (17), due to the existence of the truncation error of the estimated solution, the residual error is given by

$$\begin{aligned} e_j^{k+1} =& \hat{V}^{k+1}(x(t_j)) - \hat{V}^{k+1}(x(t_{j+1})) - \int_{t_j}^{t_{j+1}} r(x, u_1^k, u_2^k)\mathrm{d}\tau \\ & - \int_{t_j}^{t_{j+1}} 2(\hat{u}_1^{k+1})^{\mathrm{T}} R_1 (u_1' - u_1^k)\mathrm{d}\tau \\ & + \int_{t_j}^{t_{j+1}} 2\lambda (\hat{D}^{k+1})^{\mathrm{T}} R_2 (u_2' - u_2^k)\mathrm{d}\tau \\ =& [\phi_c(x(t_j)) - \phi_c(x(t_{j+1}))]^{\mathrm{T}} \hat{w}_{c,k+1} \\ & - 2\int_{t_j}^{t_{j+1}} \phi_u^{\mathrm{T}}(x) \hat{w}_{u,k+1} R_1 (u_1' - \hat{w}_{u,k}^{\mathrm{T}} \phi_u(x))\mathrm{d}\tau + 2\lambda \\ & \times \int_{t_j}^{t_{j+1}} \left(\phi_d^{\mathrm{T}}(x) \hat{w}_{d,k+1} R_2 (u_2' + \lambda \tanh(\hat{w}_{d,k}^{\mathrm{T}} \phi_d(x)))\right)\mathrm{d}\tau \\ & - \int_{t_j}^{t_{j+1}} \Big\{ x^{\mathrm{T}} Q x + \phi_u^{\mathrm{T}}(x) \hat{w}_{u,k} R_1 \hat{w}_{u,k}^{\mathrm{T}} \phi_u(x) \\ & + \int_0^{-\lambda \tanh(\hat{w}_{d,k}^{\mathrm{T}} \phi_d(x))} \left(\lambda \tanh^{-1}(s/\lambda)\right)^{\mathrm{T}} R_2 \mathrm{d}s \Big\} \mathrm{d}\tau \end{aligned} \tag{23}$$

By Kronecker product $\otimes$, a compact form of (23) is given by

$$e_j^{k+1} = \rho_j^{\mathrm{T}}(\bar{W}_k) \bar{W}_{k+1} - \pi_j(\bar{W}_k) \tag{24}$$

where $\bar{W}_{k+1} = [\hat{w}_{c,k+1}^{\mathrm{T}}, \mathrm{vec}(\hat{w}_{u,k+1})^{\mathrm{T}}, \mathrm{vec}(\hat{w}_{d,k+1})^{\mathrm{T}}]^{\mathrm{T}} \in \Re^{\bar{K}}$ is named the estimated weighting function vector with $\bar{K} = K_c + m_1 K_u + m_2 K_d$, $\bar{W}_k$ can also be expressed in the same way, $\mathrm{vec}(\cdot)$ denotes the vectorization of a matrix formed by stacking the columns of the matrix into a single column vector, the iterative index $k \in \{0, 1, \ldots\}$, the time sequence index $j \in \{0, \ldots, q\}$, and $\rho_j(\bar{W}_k)$, $\pi_j(\bar{W}_k)$ are defined as

$$\rho_j(\bar{W}_k) = \begin{bmatrix} \phi_c(x(t_j)) - \phi_c(x(t_{j+1})) \\ \int_{t_j}^{t_{j+1}} -2R_1 (u_1' - \hat{w}_{u,k}^{\mathrm{T}} \phi_u(x)) \otimes \phi_u \mathrm{d}\tau \\ \int_{t_j}^{t_{j+1}} 2\lambda R_2 (u_2' + \lambda \tanh(\hat{w}_{d,k}^{\mathrm{T}} \phi_d)) \otimes \phi_d \mathrm{d}\tau \end{bmatrix}$$

$$\pi_j(\bar{W}_k)=\int_{t_j}^{t_{j+1}}\Big\{x^{\mathrm{T}}Qx+\phi_u^{\mathrm{T}}(x)\hat{w}_{u,k}R_1\hat{w}_{u,k}^{\mathrm{T}}\phi_u(x)+\int_0^{-\lambda\tanh(\hat{w}_{d,k}^{\mathrm{T}}\phi_d(x))}\left(\lambda\tanh^{-1}(s/\lambda)\right)^{\mathrm{T}}R_2\mathrm{d}s\Big\}\mathrm{d}\tau$$

To guarantee the convergence of $\bar{W}_{k+1}$, the PE assumption which is usually needed in adaptive control algorithms is given.

Assumption 1. [41]: Let the signal $\rho_j(\bar{W}_k)$ be persistently existed, that is there exist $q_0>0$ and $\delta>0$ such that for all $q\le q_0$, we have

$$\frac{1}{q}\sum_{k=0}^{q-1}\rho_j(\bar{W}_k)\rho_j^{\mathrm{T}}(\bar{W}_k)\ge\delta I_{\bar{K}}$$

where $I_{\bar{K}}$ is the identity matrix of appropriate dimensions.

Based on the least-squares (LS) principle, it is desired to determine the estimated weighting function vector $\bar{W}_{k+1}$ by minimizing $\min_{\bar{W}_{k+1}}\sum_{j=0}^{q}(e_j^{k+1})^2$. According to (24), the solution to this LS problem yields

$$\bar{W}_{k+1}=[P^{\mathrm{T}}(\bar{W}_k)P(\bar{W}_k)]^{-1}P^{\mathrm{T}}(\bar{W}_k)\Pi(\bar{W}_k)\tag{25}$$

where

$$P(\bar{W}_k)=[\rho_0(\bar{W}_k),\ldots,\rho_q(\bar{W}_k)]^{\mathrm{T}}\tag{26}$$

$$\Pi(\bar{W}_k)=[\pi_0(\bar{W}_k),\ldots,\pi_q(\bar{W}_k)]^{\mathrm{T}}\tag{27}$$

Similar with [39], this iterative ADP is actually off-policy learning method. Note that $\rho(\bar{W}_k)$ and $\pi(\bar{W}_k)$ can be computed with a suitable initial policies weights $w_{u,0}$ and $w_{d,0}$ and collected system data. Then the algorithm is iterated using the expression (25). Accordingly, the unknown function $\hat{V}_k(x)$ and function vectors $\hat{u}_1^{k+1}(x)$ and $\hat{D}^{k+1}(x)$ can be approximately computed by (22) with the convergent $\bar{W}_{k+1}$. That is Eq. (17) is solved iteratively.

Remark 7. In general, the exploration noise is added to the given control inputs to guarantee the PE condition, i.e., $u_i'=u_i+e_i, i=1,2$, where u_i are arbitrary admissible control policies, and e_i are the exploration noise. Furthermore, the addition of exploration noise can make the sampling data set $P(\bar{W}_k)$ richer. In order to compute the inverse of matrix $P^{\mathrm{T}}(\bar{W}_k)P(\bar{W}_k)$ in (25), the matrix $P(\bar{W}_k)$ should be full column rank. To attain this goal in real implementation, the number of collected data points q is generally satisfied $q\ge\mathrm{rank}(P(\bar{W}_k))$, i.e., $q\ge\bar{K}=K_c+m_1K_u+m_2K_d$.

Then the flowchart of this data-driven iterative ADP algorithm is given in Fig. 1. Note that the proposed algorithm includes two phase. The first step is the online measurement phase, where the system data is collected under the given control inputs u_1' and u_2'. The second step is the off-policy learning phase. With the collected data and the iterative expression (25), the estimated weighting vector $\bar{W}_{k+1}$ can converge to the optimal one. Then the optimal controllers can be obtained.

3.3. Convergence analysis

In this section, the convergence of the developed data-driven iterative ADP algorithm is proved under the NN approximators (22).

Theorem 3. *Suppose that Assumption 1 holds, for $\forall\epsilon>0$, there exist integer $k^*>0$, $K_c^*>0, K_u^*>0$ and $K_d^*>0$, such that if $k>k^*$, $K_c>K_c^*, K_u>K_u^*$ and $K_d>K_d^*$, then*

1) $|\hat{V}^k(x)-V^k(x)|\le\epsilon, \|\hat{u}_1^{k+1}-u_1^{k+1}\|\le\epsilon, \|\hat{D}^{k+1}-D^{k+1}\|\le\epsilon$

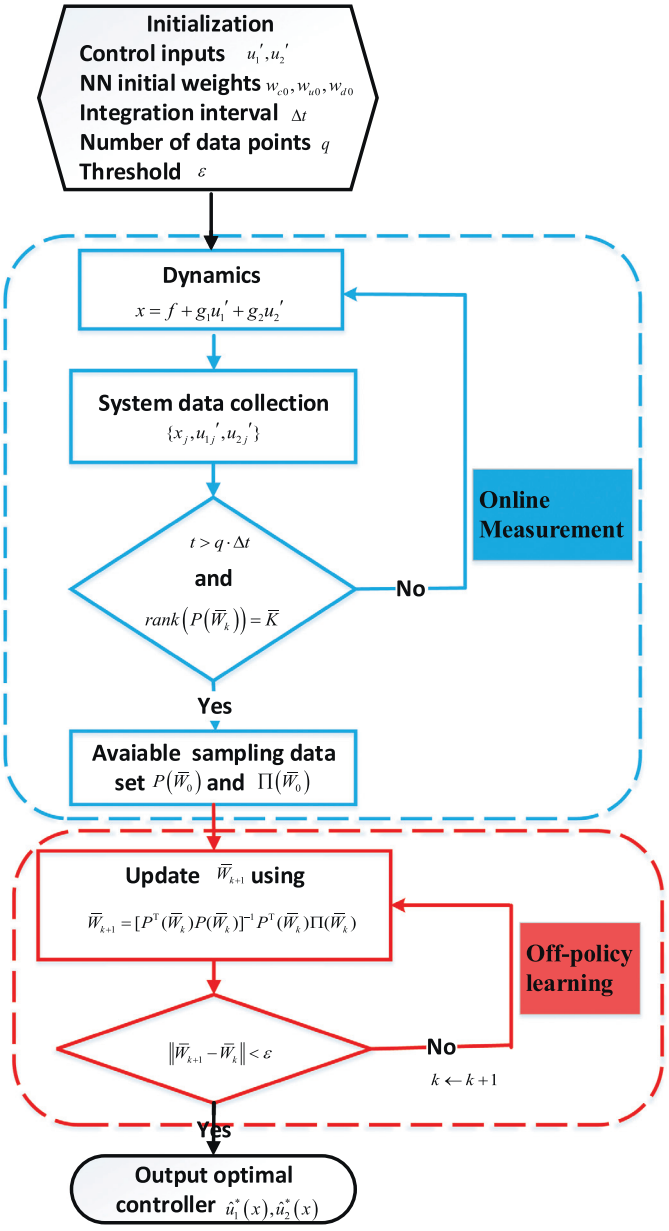


Fig. 1. The flowchart of the data-driven iterative ADP.

2) $|\hat{V}^k(x)-V^*(x)|\le\epsilon, \|\hat{u}_1^{k+1}-u_1^*\|\le\epsilon, \|\hat{D}^{k+1}-D^*\|\le\epsilon$

hold for all $x\in\Omega$,

Proof 1. The similar proof procedures is presented in [39,53]. So we omit it here.

Remark 8. Different from the convergence of algorithms for the general constrained-input systems in [53] and the zero-sum game systems in [39], Theorem 3 is focused on the convergence analysis for two-player FC game systems with partially constrained inputs in this paper. Compared with [39] that requires the partial system dynamics, the knowledge of system dynamics is not required in the proposed algorithm.

Remark 9. Due to the existence of the constrained input, a generalized non-quadratic function is introduced in the value function (4). Because the value function is a non-quadratic function, the Lyapunov-like Eq. (5) can not result in the algebraic Riccati

equation (ARE). Until now there has been no known analytical solutions to optimal control problems for linear systems with constrained inputs [55]. Note that the proposed data-driven iterative ADP method in this paper can also be used to approach the optimal solution to the optimal problem for the linear FC game systems with partially constrained inputs.

4. Simulation

To demonstrate the effectiveness of the developed algorithm, we choose two examples for numerical experiment.

4.1. Example 1

Consider the continuous-time linear systems given by

$$\dot{x} = Ax + B_1(x)u_1 + B_2(x)u_2$$

where

$$A = \begin{bmatrix} -0.0665 & 8 & 0 & 0 \\ 0 & -3.663 & 3.663 & 0 \\ -6.86 & 0 & -13.736 & -73.736 \\ 0.6 & 0 & 0 & 0 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} 0 \\ 0 \\ 13.763 \\ 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} -8 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

and $x = [x_1, x_2, x_3, x_4]^{\mathrm{T}} \in \Re^4$ is the state available for measurement, and u_1, $u_2 \in \Re$ are the control inputs with $\|u_2\| \le 10$. According to [55], one way to test the validity of the ADP algorithm for the linear constrained-input system is to select a large actuator bound, which can guarantee the constrained input does not exceed the large bound. Then, the optimal value function will be same as the value function of the ARE for the linear system with unconstrained inputs, if the proposed algorithm is effective. Hence, we choose a large enough actuator bound for the control input u_2 (i.e., $\|u_2\| \le 10$).

Let Q, R_1 and R_2 be the identity matrices with appropriate dimensions. For the unconstrained case, the optimal value function can be described as $V^* = x^{\mathrm{T}}Px$. By solving the corresponding ARE with the MATLAB command LQR, we can obtain the optimal matrix P

$$P = \begin{bmatrix} 0.1346 & 0.0709 & 0 & 0.1479 \\ 0.0709 & 0.2192 & 0.0331 & 0.0436 \\ 0 & 0.0331 & 0.0362 & -0.0164 \\ 0.1479 & 0.0436 & -0.0164 & 1.9528 \end{bmatrix}$$

Accordingly, we choose the complete basis function vector for the critic NN and two actor NNs as

$$\phi_c(x) = [x_1^2, x_2^2, x_3^2, x_4^2, x_1x_2, x_1x_4, x_2x_3, x_2x_4, x_3x_4]^{\mathrm{T}}$$

$$\phi_u(x) = \phi_d(x) = [x_1, x_2, x_3, x_4]^{\mathrm{T}}$$

Therefore, the ideal weight vector of the critic NN is

$$w_c = \big[0.1346,\ \ 0.2192,\ \ 0.0362,\ \ 1.9528, 0.1418,\ \ 0.2958,\ \ 0.0662,\ \ 0.0872,\ \ -0.328\ \big]^{\mathrm{T}}$$

Set the initial state $x_0 = [0.1, 0.2, 0.2, 0.1]^{\mathrm{T}}$, the probing control inputs $u_1' = \sin(10t) + \sin(9.3t) + \sin(5.2t) + 1.02$ and $u_2' = -10\tanh(0.2\sin(11t)^2 + \sin(7.8t)^3 + \sin(9.5t)^4)$, and the convergence threshold $\varepsilon = 10^{-6}$. The integral time interval is chosen as 0.01s. We choose the length index $q = 200$, which means the online measurement phase is terminated after 2 s. The initial weights of the three NNs are both initialized to zero. The convergence curves of $w_{c,k+1}$, $w_{u,k+1}$ and $w_{d,k+1}$ are shown in Figs. 2–4. After 8 iterations, the critic weights $w_{c,k+1}$ converge to

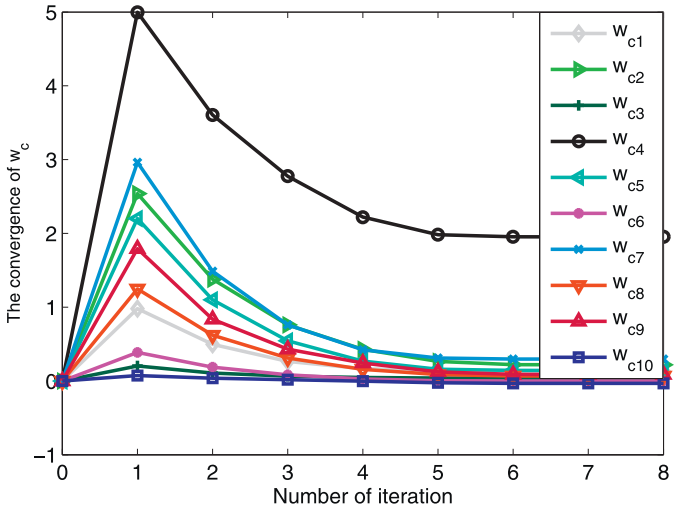


Fig. 2. The convergence curves of w_c^{k+1}.

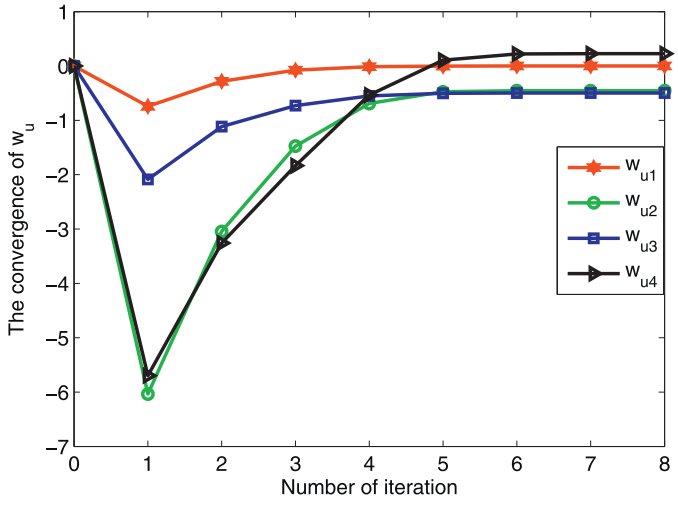


Fig. 3. The convergence curves of w_u^{k+1}.

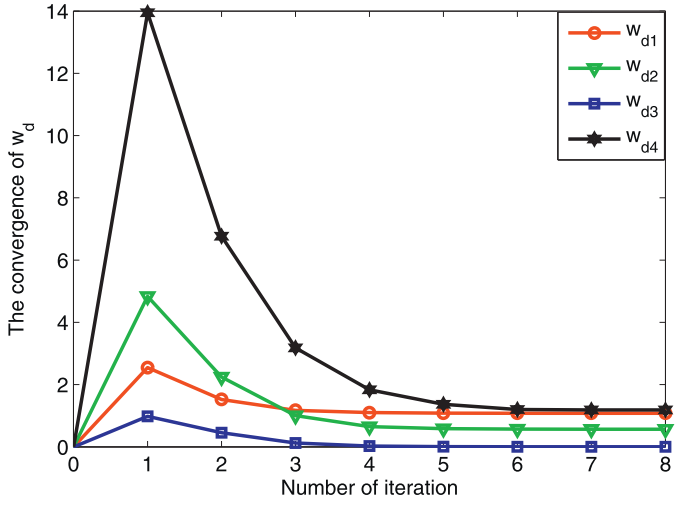


Fig. 4. The convergence curves of w_d^{k+1}.

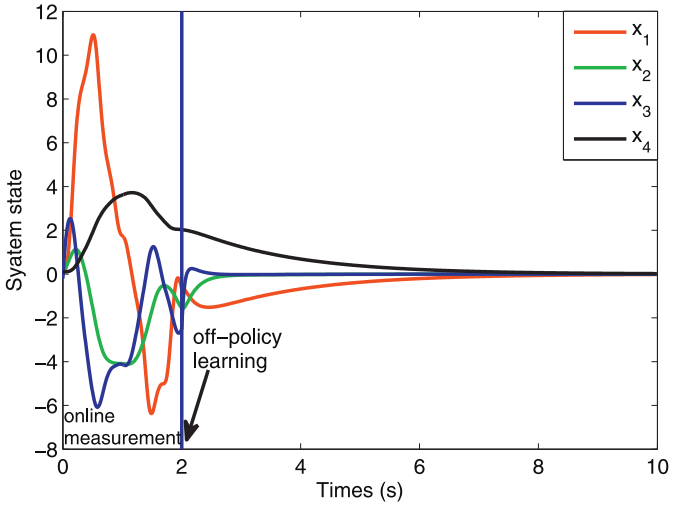


Fig. 5. Trajectories of system state.

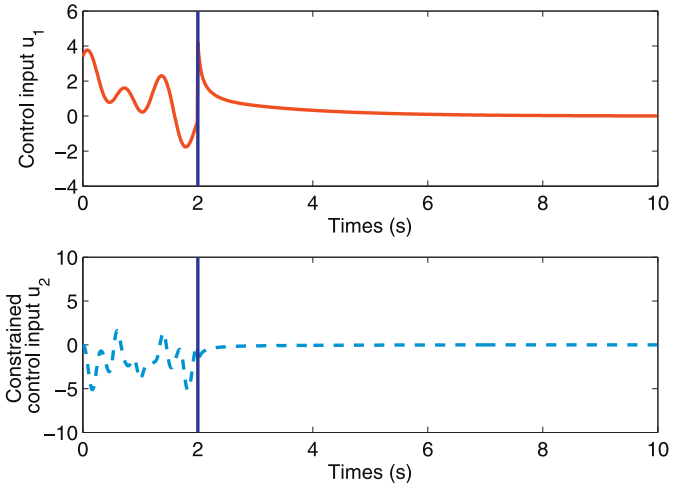


Fig. 6. The curves of u_1 and u_2 for partially constrained case.

$\hat{w}_c = [0.1345, 0.2191, 0.0362, 1.9530, 0.1419, 0.2959, 0.0663, 0.0865, -0.0332]^{\mathrm{T}}$, which are nearly the ideal values above. The trajectories of system state, the control input u_1 and the constrained input u_2 are shown in Fig. 5 and Fig. 6, respectively. We can see the system state is stable under the obtained optimal controllers, and the constrained input u_2 does not exceed the actuator bound.

4.2. Example 2

The continuous-time nonlinear FC game is give by

$$\dot{x} = f(x) + g_1(x)u_1 + g_2(x)u_2$$

where

$$f(x) = \begin{bmatrix} -0.25x_1 \\ 0.5x_2{}^3 - 0.25x_1{}^2x_2 - 0.5x_2 \end{bmatrix}$$

$$g_1(x) = \begin{bmatrix} 0 \\ x_1 \end{bmatrix}, g_2(x) = \begin{bmatrix} 0 \\ x_2 \end{bmatrix}$$

$x = [x_1, x_2]^{\mathrm{T}} \in \Re^2$ and $u_1, u_2 \in \Re$ are state and control variables, respectively.

Select $Q = I$, $R_1 = 2I$ and $R_2 = I$. In this example, we assume the control input u_2 is constrained by $\|u_2\| \le 0.5$. Then $U_2(u_2)$ defined in the value functional is

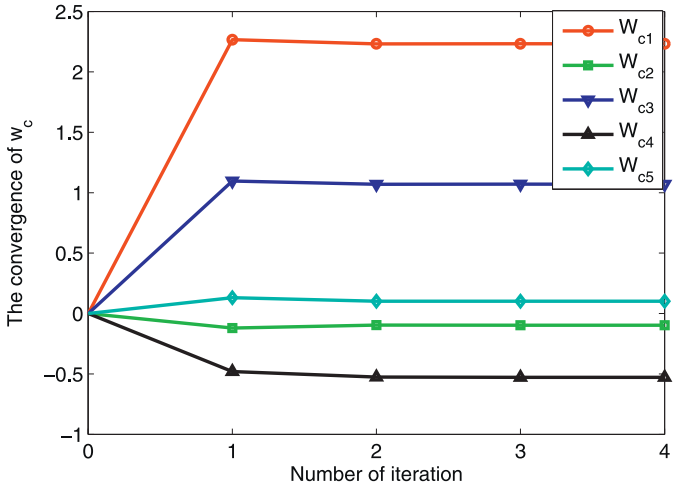


Fig. 7. The convergence curves of w_c^{k+1}.

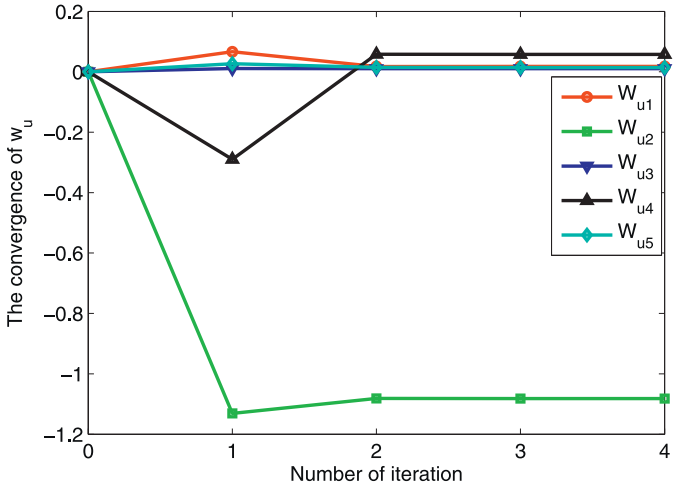


Fig. 8. The convergence curves of w_u^{k+1}.

$$U(u_2) = 2\int_0^{u_2} \left(0.5\tanh^{-1}(s/0.5)\right)^{\mathrm{T}} R_2 \mathrm{d}s$$

Based on the approximation Eq. (22), we can experimentally choose three three-layer feedforward NNs to approach the solution with the same basis function vector given by

$$\phi_c(x) = \phi_u(x) = \phi_d(x) = [x_1^2, x_1x_2, x_2^2, x_1^4, x_2^4]^{\mathrm{T}}$$

Set the initial state $x_0 = [0.5, -0.5]^{\mathrm{T}}$, the probing control inputs u_1'=$1.1(\sin(\pi t) + \sin(1.5\pi t) + \sin(1.8\pi t) + \sin(2\pi t) + 1.02)$, $u_2' = -0.5\tanh(2(\sin(1.1\pi t) + \sin(1.4\pi t) + \sin(1.2\pi t) + \sin(2.9\pi t) + \sin(3.2\pi t) - 1.78))$, and the convergence threshold $\varepsilon = 10^{-6}$. The integration is conducted at every 0.1s. We choose the length index $q = 50$. Then, the off-policy iterative learning begins at the time of 5s to obtain the optimal control policies using the collected available data. The initial iterative weights of NNs are chosen as $w_c^0 = w_u^0 = w_d^0 = [0, 0, 0, 0, 0]^{\mathrm{T}}$. The convergence curves of $w_{c,k+1}$, $w_{u,k+1}$ and $w_{d,k+1}$ are shown in Figs. 7–9. After 4 iterations, the curves of $w_{c,k+1}$ of the value function and $w_{u,k+1}$, $w_{d,k+1}$ of the control inputs can be converged. The trajectories of system state, the control input u_1 and the constrained input u_2 are shown in Fig. 10 and Fig. 11, respectively. Compared with the curve of control input u_1 without saturation, the control input u_2 in Fig. 11 is bounded with $\|u_2\| \le 0.5$. Meanwhile, the control inputs for the unconstrained case is shown in Fig. 12, where the upper and lower bounds of the unconstrained control input u_2 exceed the

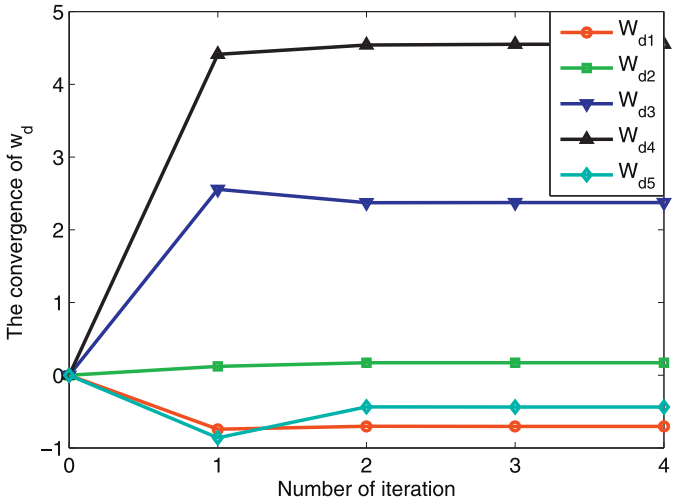


Fig. 9. The convergence curves of w_d^{k+1}.

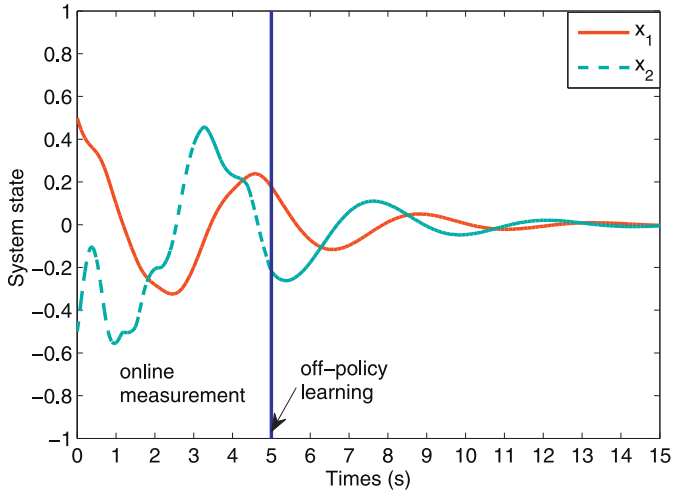


Fig. 10. Trajectories of system state.

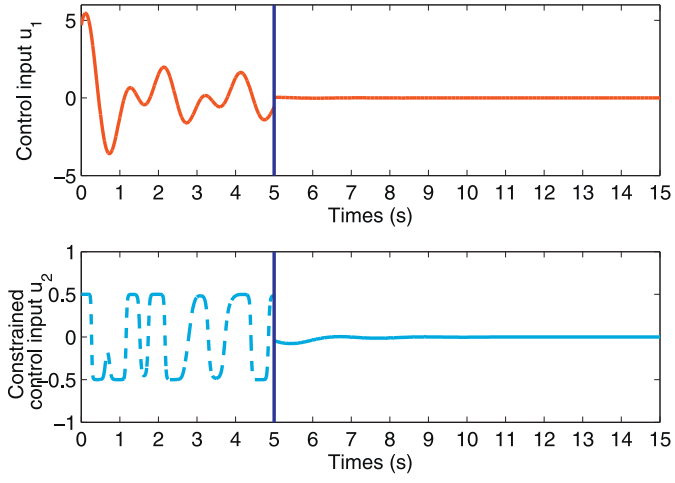


Fig. 11. The curves of u_1 and u_2 for partially constrained case.

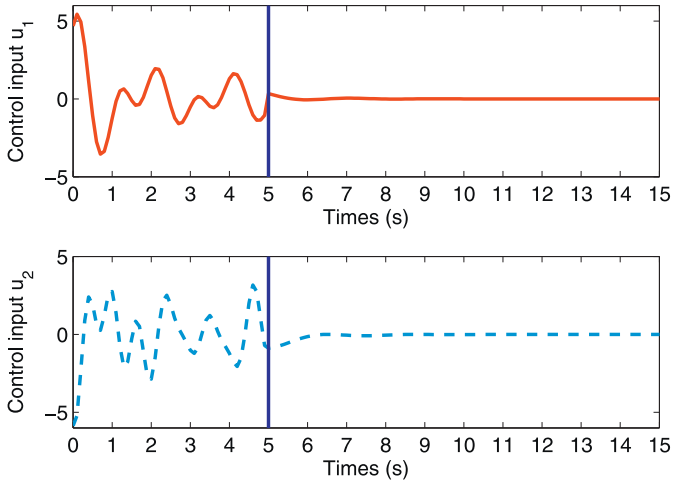


Fig. 12. The curves of u_1 and u_2 for unconstrained case.

constraint value. These simulation results verify the effectiveness of the developed control scheme for the FC game with partially constrained inputs.

5. Conclusion

The continuous-time unknown FC game with partially constrained inputs is solved by a model-free ADP algorithm based on generated system data. This algorithm is composed of two phase: online measurement and off-policy learning. Three neural networks are constructed in the off-policy learning phase to approach the optimal solution of the model-free iterative equation based on real system data. The application on two simple numerical systems demonstrates the effectiveness of the developed data-driven ADP algorithm. Our future work is to extend the data-driven ADP algorithm to the NZS game.

Acknowledgment

This work is supported by National Natural Science Foundation of China (NSFC) under Grants 61273136, 61573353, 61533017, 61603382 and the National Key Research and Development Plan under Grants No. 2016YFB0101000.

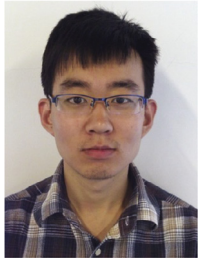
References

[1] P. Stone, M. Veloso, Multiagent systems: a survey from a machine learning perspective, Auton. Robots 8 (3) (2000) 345–383.
[2] M. Wiering, et al., Multi-agent reinforcement learning for traffic light control, in: Proceedings of ICML, 2000, pp. 1151–1158.
[3] Q. Wei, D. Liu, G. Shi, A novel dual iterative-learning method for optimal battery management in smart residential environments, IEEE Trans. Ind. Electron. 62 (4) (2015) 2509–2518.
[4] D. Zhao, Y. Zhu, MEC–a near-optimal online reinforcement learning algorithm for continuous deterministic systems, IEEE Trans. Neural Netw. Learn. Syst. 26 (2) (2015) 346–356.
[5] B. Luo, D. Liu, T. Huang, D. Wang, Model-free optimal tracking control via critic-only Q-learning, IEEE Trans. Neural Netw. Learn. Syst. 27 (10) (2016) 2134–2144.
[6] B. Luo, H.-N. Wu, Approximate optimal control design for nonlinear one-dimensional parabolic PDE systems using empirical eigenfunctions and neural network, IEEE Trans. Syst. Man Cybern. Part B Cybern. 42 (6) (2012) 1538–1549.
[7] B. Luo, H.-N. Wu, T. Huang, D. Liu, Data-based approximate policy iteration for affine nonlinear continuous-time optimal control design, Automatica 50 (12) (2014) 3281–3290.
[8] D. Zhao, Z. Zhang, Y. Dai, Self-teaching adaptive dynamic programming for Gomoku, Neurocomputing 78 (1) (2012) 23–29.
[9] D. Zhao, B. Wang, D. Liu, A supervised actor-critic approach for adaptive cruise control, Soft Comput. 17 (11) (2013) 2089–2099.
[10] D. Wang, D. Liu, Q. Wei, D. Zhao, N. Jin, Optimal control of unknown nonaffine nonlinear discrete-time systems based on adaptive dynamic programming, Automatica 48 (8) (2012) 1825–1832.

[11] H. Jiang, H. Zhang, Y. Luo, J. Wang, Optimal tracking control for completely unknown nonlinear discrete-time Markov jump systems using data-based reinforcement learning method, Neurocomputing 194 (2016) 176–182.
[12] B. Luo, H.-N. Wu, H.-X. Li, Adaptive optimal control of highly dissipative nonlinear spatially distributed processes with neuro-dynamic programming, IEEE Trans. Neural Netw. Learn. Syst. 26 (4) (2015) 684–696.
[13] D. Zhao, Z. Xia, D. Wang, Model-free optimal control for affine nonlinear systems with convergence analysis, IEEE Trans. Autom. Sci. Eng. 12 (4) (2015) 1461–1468.
[14] L. Busoniu, R. Babuska, B. De Schutter, A comprehensive survey of multiagent reinforcement learning, IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. 38 (2) (2008) 156–172.
[15] Z. Zhang, D. Zhao, Clique-based cooperative multiagent reinforcement learning using factor graphs, IEEE/CAA J. Autom. Sin. 3 (1) (2015) 248–256.
[16] M.L. Littman, Value-function reinforcement learning in Markov games, Cognit. Syst. Res. 2 (1) (2001) 55–66.
[17] M.L. Littman, Markov games as a framework for multi-agent reinforcement learning, in: Proceedings of the Eleventh International Conference on Machine Learning, 157, 1994, pp. 157–163.
[18] J. Hu, M.P. Wellman, et al., Multiagent reinforcement learning: theoretical framework and an algorithm., in: Proceedings of ICML, 98, Citeseer, 1998, pp. 242–250.
[19] Z. Khan, S. Glisic, L. DaSilva, J. Lehtomaki, et al., Modeling the dynamics of coalition formation games for cooperative spectrum sharing in an interference channel, IEEE Trans. Comput. Intell. AI Games 3 (1) (2011) 17–30.
[20] S.H. Tamaddoni, S. Taheri, M. Ahmadian, Optimal preview game theory approach to vehicle stability controller design, Veh. Syst. Dyn. 49 (12) (2011) 1967–1979.
[21] D. Zhao, Y. Dai, Z. Zhang, Computational intelligence in urban traffic signal control: a survey, IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 42 (4) (2012) 485–494.
[22] D. Liu, H. Li, D. Wang, Neural-network-based zero-sum game for discrete–time nonlinear systems via iterative adaptive dynamic programming algorithm, Neurocomputing 110 (2013) 92–100.
[23] T. Basar, G.J. Olsder, G. Clsder, T. Basar, T. Baser, G.J. Olsder, Dynamic Noncooperative Game Theory, SIAM, Philadelphia, PA, 1995.
[24] Q. Zhang, D. Zhao, Y. Zhu, Event-triggered H_∞ control for continuous-time nonlinear system via concurrent learning, IEEE Trans. Syst. Man Cybern. Syst. (2016), doi:10.1109/TSMC.2016.2531680.
[25] M.I. Abouheaf, F.L. Lewis, K.G. Vamvoudakis, S. Haesaert, B. Robert, Multi-agent discrete-time graphical games and reinforcement learning solutions, Automatica 50 (2014) 3038–3053.
[26] K. Hengster-Movric, K. You, F.L. Lewis, L. Xie, Synchronization of discrete-time multi-agent systems on graphs using Riccati design, Automatica 49 (2013) 414–423.
[27] H. Zhang, Q. Wei, D. Liu, An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games, Automatica 47 (1) (2011) 207–214.
[28] D. Liu, H. Li, D. Wang, Online synchronous approximate optimal learning algorithm for multi-player non-zero-sum games with unknown dynamics, IEEE Trans. Syst. Man Cybern. Syst. 44 (8) (2014) 1015–1027.
[29] H. Zhang, L. Cui, Y. Luo, Near-optimal control for nonzero-sum differential games of continuous-time nonlinear systems using single-network ADP, IEEE Trans. Cybern. 43 (1) (2013) 206–216.
[30] Q. Wei, R. Song, P. Yan, Data-driven zero-sum neuro-optimal control for a class of continuous-time unknown nonlinear systems with disturbance using ADP, IEEE Trans. Neural Netw. Learn. Syst. 27 (2) (2016) 444–458.
[31] Q. Zhang, D. Zhao, D. Wang, Event-based robust control for uncertain nonlinear systems using adaptive dynamic programming, IEEE Trans. Neural Netw. Learn. Syst. (2016), doi:10.1109/TNNLS.2016.2614002.
[32] K.G. Vamvoudakis, F.L. Lewis, Multi-player non-zero-sum games: online adaptive learning solution of coupled Hamilton–Jacobi equations, Automatica 47 (8) (2011) 1556–1569.
[33] D. Zhao, Q. Zhang, D. Wang, Y. Zhu, Experience replay for optimal control of nonzero-sum game systems with unknown dynamics, IEEE Trans. Cybern. 46 (3) (2016) 854–865.
[34] Q. Wei, D. Liu, F.L. Lewis, Optimal distributed synchronization control for continuous-time heterogeneous multi-agent differential graphical games, Inf. Sci. 317 (2015) 96–113.
[35] K.G. Vamvoudakis, F.L. Lewis, G.R. Hudas, Multi-agent differential graphical games: online adaptive learning solution for synchronization with optimality, Automatica 48 (2012) 1598–1611.
[36] T.L. Nguyen, Adaptive dynamic programming-based design of integrated neural network structure for cooperative control of multiple MIMO nonlinear systems, Neurocomputing (2016). http://dx.doi.org/10.1016/j.neucom.2016.05.044.
[37] D. Vrabie, F. Lewis, Adaptive dynamic programming for online solution of a zero-sum differential game, J. Control Theory Appl. 9 (3) (2011) 353–360.
[38] H.-N. Wu, 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. 23 (12) (2012) 1884–1895.
[39] B. Luo, H.-N. Wu, T. Huang, Off-policy reinforcement learning for H_∞ control design, IEEE Trans. Cybern. 45 (1) (2015) 65–76.
[40] T.L. Nguyen, T.T. Nguyen, M.T. Hoang, Reinforcement learning-based intelligent tracking control for wheeled mobile robot, Trans. Inst. Meas. Control 36 (7) (2014) 868–877.
[41] Y. Jiang, Z.-P. Jiang, Robust adaptive dynamic programming and feedback stabilization of nonlinear systems, IEEE Trans. Neural Netw. Learn. Syst. 25 (5) (2014) 882–893.
[42] J.Y. Lee, J.B. Park, Y.H. Choi, Integral reinforcement learning for continuous-time input-affine nonlinear systems with simultaneous invariant explorations, IEEE Trans. Neural Netw. Learn. Syst. 26 (5) (2015) 916–932.
[43] H. Li, D. Liu, D. Wang, Integral reinforcement learning for linear continuous–time zero-sum games with completely unknown dynamics, IEEE Trans. Autom. Sci. Eng. 11 (3) (2014) 706–714.
[44] B. Luo, T. Huang, H.-N. Wu, X. Yang, Data-driven H_∞ control for nonlinear distributed parameter systems, IEEE Trans. Neural Netw. Learn. Syst. 26 (11) (2015) 2949–2961.
[45] H. Modares, F.L. Lewis, M.-B. Naghibi-Sistani, Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems, Automatica 50 (1) (2014) 193–202.
[46] M. Abu-Khalaf, F.L. Lewis, J. Huang, Neurodynamic programming and zero-sum games for constrained control systems, IEEE Trans. Neural Netw. 19 (7) (2008) 1243–1252.
[47] H. Modares, F.L. Lewis, M.-B.N. Sistani, Online solution of nonquadratic two–player zero-sum games arising in the h control of constrained input systems, Int. J. Adapt. Control Signal Process. 28 (3–5) (2014) 232–254.
[48] S. Yasini, M.B.N. Sitani, A. Kirampor, Reinforcement learning and neural networks for multi-agent nonzero-sum games of nonlinear constrained-input systems, Int. J. Mach. Learn. Cybern. 7 (6) (2014) 967–980.
[49] M. Johnson, R. Kamalapurkar, S. Bhasin, W.E. Dixon, Approximate-player nonzero-sum game solution for an uncertain continuous nonlinear system, IEEE Trans. Neural Netw. Learn. Syst. 22 (8) (2014) 1645–1658.
[50] W. Gao, Y. Jiang, Z. Jiang, T. Chai, Output-feedback adaptive optimal control of interconnected systems based on robust adaptive dynamic programming, Automatica 72 (2016) 37–45.
[51] F.L. Lewis, V.L. Syrmos, Optimal Control, John Wiley & Sons, New York, NY, 1995.
[52] M. Abu-Khalaf, F.L. Lewis, Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach, Automatica 41 (5) (2005) 779–791.
[53] B. Luo, H.-N. Wu, T. Huang, D. Liu, Reinforcement learning solution for HJB equation arising in constrained optimal control problem, Neural Netw. 71 (2015) 150–158.
[54] B.A. Finlayson, The Method of Weighted Residuals and Variational Principles, Academic Press, New York, 1990.
[55] X. Yang, D. Liu, B. Luo, C. Li, Data-based robust adaptive control for a class of unknown nonlinear constrained-input systems via integral reinforcement learning, Inf. Sci. 336 (2016) 731–747.

Qichao Zhang: received the B.S. degree of automation from Northeastern Electric Power University, Jilin, China and the M.S. degree from Northeast University in 2012 and 2014, respectively. 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, China. His research interests include reinforcement learning, game theory and multi-agent systems.

Dongbin Zhao (M'06-SM'10) received the B.S., M.S., Ph.D. degrees from Harbin Institute of Technology, Harbin, China, in 1994, 1996, and 2000, respectively. He was a postdoctoral fellow at Tsinghua University, Beijing, China, from 2000 to 2002. He has been a professor at Institute of Automation, Chinese Academy of Sciences since 2002, and also a professor with the University of Chinese Academy of Sciences, China. From 2007 to 2008, he was also a visiting scholar at the University of Arizona. He has published 4 books, and over 50 international journal papers. His current research interests are in the area of computational intelligence, adaptive dynamic programming, deep reinforcement learning, robotics, intelligent transportation systems, and smart grids.



Dr. Zhao is the Associate Editor of *IEEE Transactions on Neural Networks and Learning Systems* (2012-), *IEEE Computation Intelligence Magazine* (2014-), etc. He serves as the Chair of *Adaptive Dynamic Programming and Reinforcement Learning Technical Committee* (2015-), *Multimedia Subcommittee* (2015-) of IEEE Computational Intelligence Society (CIS). He works as several guest editors of international journals. He is involved in organizing several international conferences.