

Data-Driven Adaptive Dynamic Programming for Two-Player Nonzero-Sum Game

Qichao Zhang^{1,2}, Dongbin Zhao^{1,2,3}, Yafei Zhou³

1. The state Key Laboratory of Management and Control for Complex Systems,
Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China

2. University of Chinese Academy of Sciences, Beijing, 100049, China

3. Jiangsu Huimin Traffic Facility Co., Ltd., Jiangsu, 223001, China
E-mail: zhangqichao2013@163.com, dongbin.zhao@ia.ac.cn

Abstract: In this paper, we propose a data-driven adaptive dynamic programming approach to solve the Hamilton-Jacobi (HJ) equations for the two-player nonzero-sum (NZS) game with completely unknown dynamics. First, the model-based policy iteration (PI) algorithm is given, where the knowledge of system dynamics is required. To relax this requirement, a data-driven adaptive dynamic programming (ADP) is proposed in this paper to solve the unknown nonlinear NZS game with only online data. Neural network approximators are constructed to approach the solution of the HJ equations. The online data is collected under the two initial admissible control policies. Then, the NN weights are updated based on the least-squares method using the collected online data repeatedly, which is a kind of the off-policy learning scheme. Finally, a simulation example is provided to demonstrate the effectiveness of the proposed control scheme.

Key Words: Adaptive dynamic programming, data-driven, neural network, unknown dynamics, off-policy

1 INTRODUCTION

Game theory can be used to solve the optimal control problem of many practical systems with multiple controllers [1]. Relying on the roles and tasks of players, the stochastic game can be divided into several types such as zero-sum (ZS) game, nonzero-sum (NZS) game and fully cooperative (FC) game. Recently, many model-based and model-free methods based on ADP and game theory have been applied to the ZS and NZS games [2–7], where the system dynamics should be known or be identified. For the multi-agent graphical games, the distributed controller based on model-based reinforcement learning (RL) algorithms can also be found in [9, 10] for linear and nonlinear systems, respectively. It should be mentioned that the training of identifiers 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 et al. proposed the integral reinforcement learning (IRL) to solve the ZS game and NZS game without the knowledge of internal dynamics in [11, 12], where the time-consuming multiple iterative loops were required in the proposed algorithms. Motivated by that, Wu and Luo [13] developed an online simultaneous policy update algorithm with only one iterative loop for the partially unknown ZS game. In [8], the single-network ADP with experience replay algorithm was proposed for the NZS game with unknown dynamics, where the system identifier was established to reconstruct

the unknown NZS game.

For completely unknown systems, an exciting work was given to solve the optimal control problem of uncertain nonlinear systems in [14]. A robust ADP was performed without any system dynamics and identification process. In [15], a model free approach was proposed based on IRL with safe explorations for the nonlinear optimal control problem. Then, this kind of data-based or data driven method was extended to the optimal control for linear ZS game in [16] and the H_∞ control problem in [17]. Unfortunately, the NZS game with completely unknown dynamics in continuous-time MDP environment based on the system data is rarely mentioned.

Because the value function of each player contains all the control inputs for the NZS game, it is difficult to deduce the model-free iteration equations to replace the HJ equations based on the online data due to the existence of coupled relationship. That is to say that the system dynamics are still to be required. To solve this problem, two additional auxiliary NNs are introduced to relax the knowledge of system dynamics in this paper. Correspondingly, a data-driven ADP is proposed to solve the optimal control problem for the unknown NZS game based on off-policy learning scheme. This is the main contribution of this paper.

The rest of this paper is organized as follows: Section II introduces the problem formulation of the two-player continuous-time NZS game system, and the model-based PI algorithm for the NZS game is proposed. In Section III, a data-driven ADP is proposed with the NN approximators, where only the online data is required. Simulation results and the conclusion are presented in Sections IV and V, respectively.

This research is supported by National Natural Science Foundation of China (NSFC) under Grants No. 61573353, No. 61533017, by the National Key Research and Development Plan under Grants 2016YF-B0101000.

2 PRELIMINARY

2.1 Problem Statement

Consider the two-player nonzero-sum differential games given by

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

where $x \in \mathbb{R}^n$ is the state vector, $u_i \in \mathbb{R}^{m_i}$, $i = 1, 2$ are the control inputs, $f(\cdot) \in \mathbb{R}^n$, $g_i(\cdot) \in \mathbb{R}^{n \times m_i}$ are smooth nonlinear dynamics. Assume that $f(\cdot)$ and $g_i(\cdot)$ are unknown and Lipschitz continuous on a compact set $\Omega \subseteq \mathbb{R}^n$ with $f(0) = 0$. In order to facilitate the expression, we use x and u_i to represent $x(t)$ and $u_i(t)$ in the following presentation, respectively.

Define the cost functions associated with each player as

$$\begin{aligned} J_i(x_0, u_1, u_2) &= \int_0^\infty (Q_i(x) + \sum_{j=1}^2 u_j^T R_{ij} u_j) dt \\ &= \int_0^\infty r_i(x, u_1, u_2) dt, \quad i = 1, 2 \end{aligned} \quad (2)$$

where $r_i(x, u_1, u_2) = Q_i(x) + \sum_{j=1}^2 u_j^T R_{ij} u_j$, $Q_i(x) = x^T Q_i x$ is positive definite with $Q_i > 0$. $R_{ij} > 0$, $R_{ii} \geq 0$ are symmetric matrices, and $x_0 = x(0)$ denotes the initial state.

To begin with, let us introduce the concept of the admissible control [8].

Definition 1: A feedback control policy pair $u = \{u_1, u_2\}$ is defined as admissible with respect to (2) on the set Ω , denoted by $u_i \in \Psi(\Omega)$, if $u_i(x)$ is continuous on Ω with $u_i(0) = 0$, u stabilizes (1) on Ω , and (2) is finite $\forall x_0 \in \Omega$. Define the value functions for any admissible strategies $u_i(x) \in \Psi(\Omega)$ as

$$\begin{aligned} V_i(x, u_1, u_2) &= \int_t^\infty (Q_i(x(\tau)) + \sum_{j=1}^2 u_j^T(\tau) R_{ij} u_j(\tau)) d\tau \\ &= \int_t^\infty r_i(x(\tau), u_1(\tau), u_2(\tau)) d\tau, \quad i = 1, 2 \end{aligned} \quad (3)$$

The objective of the two-player nonzero-sum games is to find an optimal admissible control policy pair $\{u_1^*, u_2^*\}$ to minimize the cost functions associated with each player. This paper will focus on the so-called Nash equilibrium solution $\{u_1^*, u_2^*\}$ that is given by the following definition.

Definition 2: A two-tuple of policies $\{u_1^*, u_2^*\}$ with $u_i^* \in \Psi(\Omega)$ is said to constitute a Nash equilibrium for a two-player NZS game, if the following inequalities are satisfied for all $u_i \in \Psi(\Omega)$

$$\begin{aligned} J_1^*(u_1^*, u_2^*) &\leq J_1(u_1, u_2^*) \\ J_2^*(u_1^*, u_2^*) &\leq J_2(u_1^*, u_2) \end{aligned} \quad (4)$$

Assume that the value function (3) are continuously differentiable, the differential equivalent of (3) can be given by

$$0 = r_i(x, u_1, u_2) + (\nabla V_i)^T \left(f + \sum_{j=1}^2 g_j u_j \right), \quad i = 1, 2 \quad (5)$$

where ∇ denotes the partial derivative operator, such that $\nabla V_i = \partial V_i(x)/\partial x$.

Define the Hamiltonian functions of system (1) and value functions (3) are

$$\begin{aligned} H_i(x, \nabla V_i, u_1, u_2) &= r_i(x, u_1, u_2) \\ &+ (\nabla V_i)^T \left(f(x) + \sum_{j=1}^2 g_j(x) u_j \right), \quad i = 1, 2 \end{aligned} \quad (6)$$

The optimal value function $V_i^*(x)$ are defined as

$$V_i^*(x) = \min_{u_i} \int_t^\infty (Q_i(x(\tau)) + \sum_{j=1}^2 u_j^T(\tau) R_{ij} u_j(\tau)) d\tau$$

Applying the stationarity conditions $\partial H_i / \partial u_i = 0$, the optimal feedback control policy associated with the optimal value function can be obtained by

$$u_i^* = -\frac{1}{2} R_{ii}^{-1} g_i^T(x) \nabla V_i^*, \quad i = 1, 2 \quad (7)$$

Based on (6) and (7), the coupled Hamilton-Jacobi (HJ) equation can be presented as

$$\begin{aligned} 0 &= Q_i(x) + (\nabla V_i^*)^T f(x) - \frac{1}{2} (\nabla V_i^*)^T \sum_{j=1}^2 g_j(x) R_{jj}^{-1} g_j^T(x) \\ &\times (\nabla V_j^*) + \frac{1}{4} \sum_{j=1}^2 (\nabla V_j^*)^T g_j(x) R_{jj}^{-1} R_{ij} R_{jj}^{-1} g_j^T(x) \nabla V_j^* \end{aligned} \quad (8)$$

with $V_i^*(0) = 0$ and $i = 1, 2$.

2.2 Policy Iteration for Solving Coupled HJ Equations

To solve the optimal control problem, we aim to obtain the optimal control policy by solving the coupled HJ equations (8). As we know, it is hard to obtain the analytical solution of the HJ equations for nonlinear systems. Among the various proposed methods to approach the solution of the HJ equations, policy iteration is one of the most common methods, which can be described as follows [8].

Algorithm 1 (PI for NZS game)

1: **Policy Evaluation.** Given an initial admissible control policy $u_i^0(x)$, find $V_i^k(x)$ successively approximated by solving the following equation

$$0 = r_i(x, u_1^k, u_2^k) + (\nabla V_i^{k+1})^T \left(f(x) + \sum_{j=1}^2 g_j(x) u_j^k \right) \quad (9)$$

with $V_i^k(0) = 0$, $i = 1, 2$, $k = 0, 1, \dots$

2: **Policy Improvement.** Update the control policies simultaneously by

$$u_i^{k+1}(x) = -\frac{1}{2} R_{ii}^{-1} g_i^T(x) \nabla V_i^{k+1}(x) \quad (10)$$

where k is the iterative index.

According to [20], this algorithm can converge to the optimal cost and optimal control policy pair, i.e., $V^k(x) \rightarrow V^*(x)$ and $\{u_1^k(x), u_2^k(x)\} \rightarrow \{u_1^*(x), u_2^*(x)\}$ as $i \rightarrow \infty$.

From the policy iteration algorithm, we know that the system dynamics $f(x)$, $g_1(x)$ and $g_2(x)$ are both required. In the following, we will propose the data-based ADP algorithm for the NZS game to approach the optimal solution, where the knowledge of system dynamics is relaxed.

3 DATA-DRIVEN ADP for NZS GAME

3.1 Data-driven Iterative ADP

Consider an arbitrary control policy u'_i applied to (1). The derivative of $V_i^{k+1}(x)$ with respect to time along the system trajectory $f + g_1 u'_1 + g_2 u'_2$ equals to $dV_i^{k+1}/dt = (\nabla V_i^{k+1})^T (f + g_1 u'_1 + g_2 u'_2)$. Based on the IRL and policy iteration algorithm, we have the novel Bellman equation along the interval $[t - \Delta t, t]$

$$\begin{aligned} & V_1^{k+1}(x(t)) - V_1^{k+1}(x(t - \Delta t)) \\ &= - \int_{t-\Delta t}^t r_1(x, u_1^k, u_2^k) d\tau + \int_{t-\Delta t}^t -2(u_1^{k+1})^T \\ & \quad \times R_{11}(u'_1 - u_1^k) d\tau + \int_{t-\Delta t}^t (D_1^{k+1})^T (u'_2 - u_2^k) d\tau \end{aligned} \quad (11)$$

$$\begin{aligned} & V_2^{k+1}(x(t)) - V_2^{k+1}(x(t - \Delta t)) \\ &= - \int_{t-\Delta t}^t r_2(x, u_1^k, u_2^k) d\tau + \int_{t-\Delta t}^t -2(u_2^{k+1})^T \\ & \quad \times R_{22}(u'_1 - u_1^k) d\tau + \int_{t-\Delta t}^t (D_2^{k+1})^T (u'_1 - u_1^k) d\tau \end{aligned} \quad (12)$$

where ∇V_i^{k+1} , u_i^{k+1} , D_i^{k+1} , $i = 1, 2$ are the unknown function vectors to be solved, $\Delta t > 0$ is the integral interval. Different from the ZS game, two unknown functions $D_1^{k+1} = (\nabla V_1^{k+1})^T g_2$ and $D_2^{k+1} = (\nabla V_2^{k+1})^T g_1$ are introduced for the two-player NZS game. Due to the knowledge of g_i in D_i^{k+1} is unknown, two additional approximators are constructed to approach the unknown function D_i^{k+1} . Thus, the unknown function $(V_i^{k+1}, u_i^{k+1}, D_i^{k+1})$ for each player is iterated following (11) and (12), respectively. The data-driven iterative ADP can be described as follows.

Algorithm 2 (Data-driven iterative ADP)

- 1: Select admissible control policies u'_1 and u'_2 to collect the online system data along the integral interval.
 - 2: Let $k = 0$, select initial iterative control policies u_1^0 and u_2^0 .
 - 3: Let $k \geq 0$, solve the solution iteratively from (11) and (12).
-

Motivated by [17], as the iteration step k increases, the convergence of the generated solution sequence $(V_i^{k+1}, u_i^{k+1}, D_i^{k+1})$ to the optimal one is proved as follows.

Theorem 1 Let $V_i^{k+1}(x) \in C^1(\Omega)$, $C^1(\Omega)$ denotes a function space on Ω with first derivatives are continuous, $V_i^{k+1}(x) \geq 0$, $V_i^{k+1}(0) = 0$ and $u_i^{k+1}(x) \in \Psi(\Omega)$, $i = 1, 2$. Then (V_i^{k+1}, u_i^{k+1}) is the solution of (10)-(11) for $\forall u'_i \in \Psi(\Omega)$, $i = 1, 2$ if and only if it is a solution of the model-based iterative equations (9)-(10).

Proof: The mechanism is similar with [17], so we omit it here.

3.2 NN-based Iterative Algorithm

Next, the NN approximation is introduced to solve the model-free iterative equation approximately. According to the Weirstrass high-order approximation theorem, a smooth function can be uniformly approximated on a compact set by NNs,

$$\begin{aligned} V_i^{k+1}(x) &= w_{ci,k+1}^T \phi_{ci}(x) + \varepsilon_{ci,k+1} \\ u_i^{k+1}(x) &= w_{ui,k+1}^T \phi_{ui}(x) + \varepsilon_{ui,k+1} \\ D_i^{k+1}(x) &= w_{di,k+1}^T \phi_{di}(x) + \varepsilon_{di,k+1} \end{aligned} \quad (13)$$

where $\phi_{ci} : \mathbb{R}^n \rightarrow \mathbb{R}^{K_{ci}}$, $\phi_{ui} : \mathbb{R}^n \rightarrow \mathbb{R}^{K_{ui}}$ and $\phi_{di} : \mathbb{R}^n \rightarrow \mathbb{R}^{K_{di}}$ are linearly independent basis function vectors, $w_{ci,k+1} \in \mathbb{R}^{K_{ci}}$, $w_{ui,k+1} \in \mathbb{R}^{K_{ui} \times m_1}$ and $w_{di,k+1} \in \mathbb{R}^{K_{di} \times m_2}$ are the unknown coefficient vector and matrices with K_{ci} , K_{ui} and K_{di} the numbers of hidden neurons, $\varepsilon_{ci,k+1}$, $\varepsilon_{ui,k+1}$ and $\varepsilon_{di,k+1}$ are the reconstruction errors with appropriate dimensions for $i = 1, 2$. It is shown in [19] that as $K_{ci} \rightarrow \infty$, $K_{ui} \rightarrow \infty$ and $K_{di} \rightarrow \infty$, the reconstruction errors $\varepsilon_{ci,k+1}$, $\varepsilon_{ui,k+1}$ and $\varepsilon_{di,k+1}$ converge to zero.

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

$$\begin{aligned} \hat{V}_i^{k+1}(x) &= \hat{w}_{ci,k+1}^T \phi_{ci}(x) \\ \hat{u}_i^{k+1}(x) &= \hat{w}_{ui,k+1}^T \phi_{ui}(x) \\ \hat{D}_i^{k+1}(x) &= \hat{w}_{di,k+1}^T \phi_{di}(x) \end{aligned} \quad (14)$$

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 $(\hat{V}_i^{k+1}(x), \hat{u}_i^{k+1}(x), \hat{D}_i^{k+1}(x))$ instead of $(V_i^{k+1}(x), u_i^{k+1}(x), D_i^{k+1}(x))$ in the equations (11)-(12), due to the existence of the truncation error of the estimated solution, the residual errors for the two player are given by

$$\begin{aligned} e_{1,j}^{k+1} &= (\phi_{c1}(x(t_{j+1})) - \phi_{c1}(x(t_j)))^T \hat{w}_{c1,k+1} \\ &+ \int_{t_j}^{t_{j+1}} 2\phi_{u1}^T \hat{w}_{u1,k+1} R_{11}(u'_1 - \hat{w}_{u1,k}^T \phi_{u1}) d\tau \\ &- \int_{t_j}^{t_{j+1}} \phi_{d1}^T \hat{w}_{d1,k+1} (u'_2 - \hat{w}_{u2,k}^T \phi_{u2}) d\tau \\ &+ \int_{t_j}^{t_{j+1}} r_1(x, \hat{w}_{u1,k}^T \phi_{u1}, \hat{w}_{u2,k}^T \phi_{u2}) d\tau \end{aligned} \quad (15)$$

$$\begin{aligned}
e_{2,j}^{k+1} &= (\phi_{c2}(x(t_{j+1})) - \phi_{c2}(x(t_j)))^T \hat{w}_{c2,k+1} \\
&+ \int_{t_j}^{t_{j+1}} 2\phi_{u2}^T \hat{w}_{u2,k+1} R_{22} (u'_2 - \hat{w}_{u2,k}^T \phi_{u2}) d\tau \\
&- \int_{t_j}^{t_{j+1}} \phi_{d2}^T \hat{w}_{d2,k+1} (u'_1 - \hat{w}_{u1,k}^T \phi_{u1}) d\tau \\
&+ \int_{t_j}^{t_{j+1}} r_2(x, \hat{w}_{u1,k}^T \phi_{u1}, \hat{w}_{u2,k}^T \phi_{u2}) d\tau
\end{aligned} \tag{16}$$

By the Kronecker product \otimes , we have

$$\begin{aligned}
&\phi_{ui}^T \hat{w}_{ui,k+1} R_{ii} (u'_i - \hat{w}_{ui,k}^T \phi_{ui}) \\
&= ((u'_i - \hat{w}_{ui,k}^T \phi_{ui})^T R \otimes \phi_2^T) \mathbf{v}(\hat{w}_{ui,k+1}) \\
&\phi_{d1}^T \hat{w}_{d1,k+1} (u'_2 - \hat{w}_{u2,k}^T \phi_{u2}) \\
&= ((u'_2 - \hat{w}_{u2,k}^T \phi_{u2})^T \otimes \phi_{d1}^T) \mathbf{v}(\hat{w}_{d1,k+1}) \\
&\phi_{d2}^T \hat{w}_{d2,k+1} (u'_1 - \hat{w}_{u1,k}^T \phi_{u1}) \\
&= ((u'_1 - \hat{w}_{u1,k}^T \phi_{u1})^T \otimes \phi_{d2}^T) \mathbf{v}(\hat{w}_{d2,k+1})
\end{aligned}$$

where $\mathbf{v}(\cdot)$ is a vector operator, which transforms a matrix into a vector by stacking its columns. Then, the residual errors can be rewritten as

$$\begin{aligned}
e_{1,j}^{k+1} &= \rho_{1,j}^T (\bar{W}_{1,k}) \bar{W}_{1,k+1} + \pi_{1,j} (\bar{W}_{1,k}) \\
e_{2,j}^{k+1} &= \rho_{2,j}^T (\bar{W}_{2,k}) \bar{W}_{2,k+1} + \pi_{2,j} (\bar{W}_{2,k})
\end{aligned} \tag{17}$$

where $\bar{W}_{i,k+1} = [\hat{w}_{ci,k+1}^T, \mathbf{v}(\hat{w}_{ui,k+1})^T, \mathbf{v}(\hat{w}_{di,k+1})^T]^T \in \mathbb{R}^{\bar{K}_i}$ is named the estimated weighting function vector with $\bar{K}_i = K_{ci} + m_1 K_{ui} + m_2 K_{di}$, $\bar{W}_k = [\hat{w}_{c1,k}^T, \hat{w}_{c2,k}^T, \mathbf{v}(\hat{w}_{u1,k})^T, \mathbf{v}(\hat{w}_{u2,k})^T, \mathbf{v}(\hat{w}_{d1,k})^T, \mathbf{v}(\hat{w}_{d2,k})^T]^T$, the player $i = 1, 2$, the iterative index $k \in \{0, 1, \dots\}$, the time sequence index $j \in \{0, \dots, q\}$, and $\rho_{i,j}(\bar{W}_{i,k})$, $\pi_j(\bar{W}_{i,k})$ are defined as

$$\begin{aligned}
\rho_{1,j}(\bar{W}_k) &= \begin{bmatrix} \phi_{c1}(x(t_{j+1})) - \phi_{c1}(x(t_j)) \\ \int_{t_j}^{t_{j+1}} 2R_{11}(u'_1 - \hat{w}_{u1,k}^T \phi_{u1}) \otimes \phi_{u1} d\tau \\ - \int_{t_j}^{t_{j+1}} (u'_2 - \hat{w}_{u2,k}^T \phi_{u2}) \otimes \phi_{d1} d\tau \end{bmatrix} \\
\rho_{2,j}(\bar{W}_k) &= \begin{bmatrix} \phi_{c2}(x(t_{j+1})) - \phi_{c2}(x(t_j)) \\ \int_{t_j}^{t_{j+1}} 2R_{22}(u'_2 - \hat{w}_{u2,k}^T \phi_{u2}) \otimes \phi_{u2} d\tau \\ - \int_{t_j}^{t_{j+1}} (u'_1 - \hat{w}_{u1,k}^T \phi_{u1}) \otimes \phi_{d2} d\tau \end{bmatrix} \\
\pi_{1,j}(\bar{W}_k) &= \int_{t_j}^{t_{j+1}} \{x^T Q_1 x + \phi_{u1}^T(x) \hat{w}_{u1,k} R_{11} \hat{w}_{u1,k}^T \phi_{u1}(x) \\
&\quad + \phi_{u2}^T(x) \hat{w}_{u2,k} R_{12} \hat{w}_{u2,k}^T \phi_{u2}(x)\} d\tau \\
\pi_{2,j}(\bar{W}_k) &= \int_{t_j}^{t_{j+1}} \{x^T Q_2 x + \phi_{u1}^T(x) \hat{w}_{u1,k} R_{21} \hat{w}_{u1,k}^T \phi_{u1}(x) \\
&\quad + \phi_{u2}^T(x) \hat{w}_{u2,k} R_{22} \hat{w}_{u2,k}^T \phi_{u2}(x)\} d\tau
\end{aligned}$$

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

Assumption 1 [14]: Let the signal $\rho_{i,j}(\bar{W}_k)$ be persistently existed, that is there exist $q_0 > 0$ and $\delta > 0$ such that for all $q \leq q_0$, we have

$$\frac{1}{q} \sum_{k=0}^{q-1} \rho_{i,j}(\bar{W}_{i,k}) \rho_{i,j}^T(\bar{W}_k) \geq \delta I_{\bar{K}_i}$$

where $I_{\bar{K}_i}$ is the identity matrix of appropriate dimensions. Based on the least-squares (LS) principle, it is desired to determine the estimated weighting function vector $\bar{W}_{i,k+1}$ by minimizing $\min_{\bar{W}_{i,k+1}} \sum_{j=0}^q (e_{i,j}^{k+1})^2$. According to (17), the solution to this LS problem yields

$$\bar{W}_{i,k+1} = [P_i^T(\bar{W}_k) P_i(\bar{W}_k)]^{-1} P_i^T(\bar{W}_k) \Pi_i(\bar{W}_k) \tag{18}$$

where

$$P_i(\bar{W}_k) = [\rho_{i,0}(\bar{W}_k), \dots, \rho_{i,q}(\bar{W}_k)]^T \tag{19}$$

$$\Pi_i(\bar{W}_k) = [\pi_{i,0}(\bar{W}_k), \dots, \pi_{i,q}(\bar{W}_k)]^T \tag{20}$$

Similar with [17], this iterative ADP is actually off-policy learning method. Note that $\rho_{i,j}(\bar{W}_k)$ and $\pi_{i,j}(\bar{W}_k)$ can be computed with a suitable initial policies weights $w_{ui,0}$ and $w_{di,0}$ and collected system data. Then the algorithm is iterated using the expression (18). Accordingly, the unknown function $\hat{V}_i^{k+1}(x)$ and function vectors $\hat{u}_i^{k+1}(x)$ and $\hat{D}_i^{k+1}(x)$ can be approximately computed by (14) with the convergent $\bar{W}_{i,k+1}$. That is, the equations (11) and (12) are solved iteratively.

3.3 Algorithm Implementation

This subsection is how to implement the data-based ADP algorithm to solve the unknown nonlinear NZS with online data. Motivated by [21], we need to transform the main equation (18) into the following Kronecker product representation

$$\begin{aligned}
P_1(\bar{W}_k) &= [\delta_{1,1}, 2\delta_{1,2}(R_{11} \otimes I_{K_{u1}}) - 2\delta_{1,3}(\hat{w}_{u1,k} R_{11} \otimes I_{K_{u1}}) \\
&\quad - \delta_{1,4} + \delta_{1,5}(\hat{w}_{u2,k} \otimes I_{K_{d1}})] \\
P_2(\bar{W}_k) &= [\delta_{2,1}, 2\delta_{2,2}(R_{22} \otimes I_{K_{u2}}) - 2\delta_{2,3}(\hat{w}_{u2,k} R_{22} \otimes I_{K_{u2}}) \\
&\quad - \delta_{2,4} + \delta_{2,5}(\hat{w}_{u1,k} \otimes I_{K_{d2}})] \\
\Pi_1(\bar{W}_k) &= [\delta_{1,6} + \delta_{1,3} \mathbf{v}(\hat{w}_{u1,k} R_{11} \hat{w}_{u1,k}^T) \\
&\quad + \delta_{2,3} \mathbf{v}(\hat{w}_{u2,k} R_{12} \hat{w}_{u2,k}^T)] \\
\Pi_2(\bar{W}_k) &= [\delta_{2,6} + \delta_{1,3} \mathbf{v}(\hat{w}_{u1,k} R_{21} \hat{w}_{u1,k}^T) \\
&\quad + \delta_{2,3} \mathbf{v}(\hat{w}_{u2,k} R_{22} \hat{w}_{u2,k}^T)]
\end{aligned}$$

where

$$\begin{aligned}
\delta_{i,1} &= [\phi_{ci}(x(t_1)) - \phi_{ci}(x(t_0)), \dots, \phi_{ci}(x(t_q)) - \phi_{ci}(x(t_{q-1}))]^T \\
\delta_{i,2} &= \left[\int_{t_0}^{t_1} u'_i \otimes \phi_{ui} d\tau, \dots, \int_{t_q}^{t_{q-1}} u'_i \otimes \phi_{ui} d\tau \right]^T \\
\delta_{i,3} &= \left[\int_{t_0}^{t_1} \phi_{ui} \otimes \phi_{ui} d\tau, \dots, \int_{t_q}^{t_{q-1}} \phi_{ui} \otimes \phi_{ui} d\tau \right]^T \\
\delta_{1,4} &= \left[\int_{t_0}^{t_1} u'_2 \otimes \phi_{d1} d\tau, \dots, \int_{t_q}^{t_{q-1}} u'_2 \otimes \phi_{d1} d\tau \right]^T \\
\delta_{2,4} &= \left[\int_{t_0}^{t_1} u'_1 \otimes \phi_{d2} d\tau, \dots, \int_{t_q}^{t_{q-1}} u'_1 \otimes \phi_{d2} d\tau \right]^T
\end{aligned}$$

$$\begin{aligned}\delta_{1,5} &= \left[\int_{t_0}^{t_1} \phi_{u2} \otimes \phi_{d1} d\tau, \dots, \int_{t_q}^{t_{q-1}} \phi_{u2} \otimes \phi_{d1} d\tau \right]^T \\ \delta_{2,5} &= \left[\int_{t_0}^{t_1} \phi_{u1} \otimes \phi_{d2} d\tau, \dots, \int_{t_q}^{t_{q-1}} \phi_{u1} \otimes \phi_{d2} d\tau \right]^T \\ \delta_{i,6} &= \left[\int_{t_0}^{t_1} x^T Q_i x d\tau, \dots, \int_{t_q}^{t_{q-1}} x^T Q_i x d\tau \right]^T\end{aligned}$$

with $i = 1, 2$. The above matrices depend on system input data u'_i , which can be used repeatedly to update P_i and Π_i at each iteration with new NN weights, which help to reduce the online interaction with the system.

4 SIMULATION

Consider the following two-player affine nonlinear nonzero-sum game system as follows [7], [8]:

$$\dot{x} = f(x) + g_1(x)u_1 + g_2(x)u_2 \quad (21)$$

where

$$\begin{aligned}f(x) &= \begin{bmatrix} x_2 \\ -x_2 - 0.5x_1 + 0.25x_2(\cos(2x_1) + 2)^2 \\ + 0.25x_2(\sin(2x_1) + 2)^2 \end{bmatrix} \\ g_1(x) &= \begin{bmatrix} 0 \\ \cos(2x_1) + 2 \end{bmatrix}, \quad g_2(x) = \begin{bmatrix} 0 \\ \sin(4x_1^2) + 2 \end{bmatrix}\end{aligned}$$

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

Select $Q_1(x) = x^T x$, $Q_2(x) = \frac{1}{2}x^T x$, $R_{11} = R_{12} = 2I$, and $R_{21} = R_{22} = I$, where I is an identity matrix. From [8], the optimal value functions are $V_1^*(x) = 0.25x_1^2 + x_2^2$ and $V_2^*(x) = 0.25x_1^2 + 0.5x_2^2$. The activation functions of the critic NNs of two players are selected as

$$\phi_{c1}(x) = \phi_{c2}(x) = [x_1^2 \ x_1 x_2 \ x_2^2]^T$$

and the activation functions of the \hat{u}_i and \hat{D}_i are chosen as

$$\phi_{ui}(x) = \phi_{di}(x) = [x_1 \ x_2 \ x_1^2 \ x_1 x_2 \ x_2^2 \ x_1^4 \ x_1^3 x_2 \ x_1^2 x_2^2 \ x_1 x_2^3 \ x_2^4]^T$$

The initial state vector is chosen as $x_0 = [1, -1]^T$. Set the initial probing control inputs $u'_1 = 0.7e^{-0.006t} \sin(t)^2 \cos(t) + \sin(2t)^2 \cos(0.1t) + \sin(-1.2t)^2 \cos(0.5t) + \sin(t)^5 + 0.5(x_1 + x_2)(\cos(2x_1) + 2)$ and $u'_2 = 0.7e^{-0.006t} \sin(t)^2 \cos(t) + \sin(2t)^2 \cos(0.1t) + \sin(1.12t)^2 + \cos(2.4t) \sin(2.4t)^3 + (x_1 + x_2)(\sin(4x_1) + 2)$, and the convergence threshold $\varepsilon = 10^{-6}$. The integral time interval is chosen as 0.1s. We choose the length index $q = 200$, which means the online data collection phase is terminated after 20s. The initial weights of the critic NNs are initialized by $w_{c1} = [2.2 \ 0.2 \ 0.8]^T$ and $w_{c2} = [0.15 \ 0.25 \ 2.4]^T$, and the initial weights of the actor and auxiliary NNs are both initialized to be zero. The convergence curves of w_{ci} are shown in Figs. 1-2.

After 10 iterations, the critic NNs weights $w_{ci,k+1}$ converge to $\hat{w}_{c1} = [0.2546 \ -0.098 \ 0.9893]^T$ and $\hat{w}_{c2} = [0.2736 \ -0.063 \ 0.4923]^T$, which are nearly the ideal values above. The similar approached results are shown based

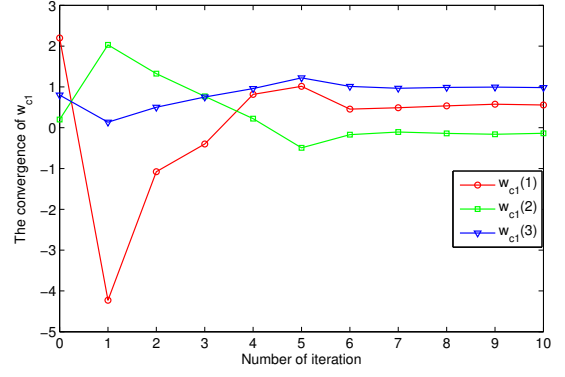


Figure 1: The convergence curves of w_{c1}

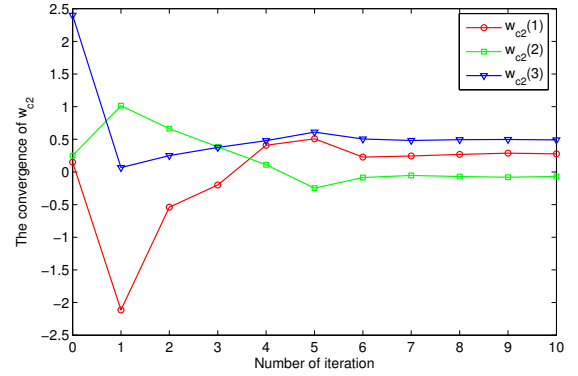


Figure 2: The convergence curves of w_{c2}

on the online learning scheme in [7,8]. Compared with [7], the knowledge of system dynamics is relaxed in the proposed off-policy learning algorithm. Different with [8], the system identifier is also not required. The trajectories of system state, the control inputs u_1 and u_2 are shown in Fig. 3 and Fig. 4, respectively. We can see the system state is stable under the obtained optimal controllers. These simulation results verify the effectiveness of the developed control scheme for the NZS game with unknown dynamics.

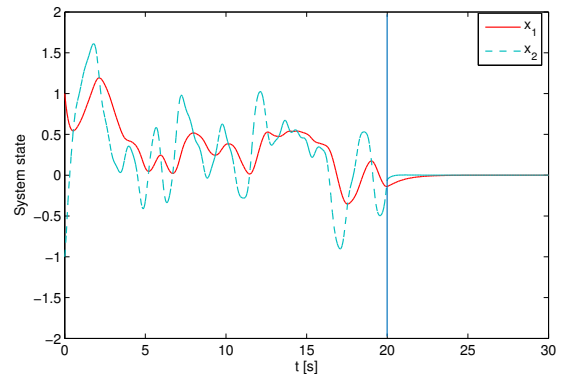


Figure 3: Trajectories of system state

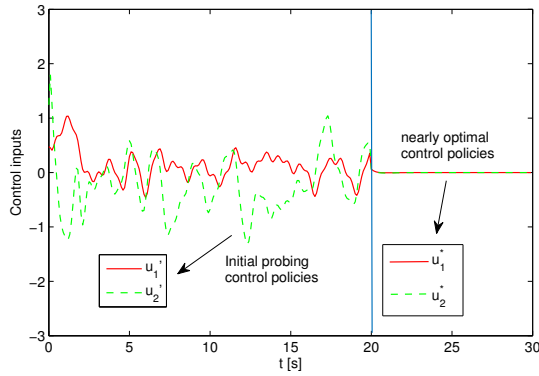


Figure 4: Trajectories of control inputs

5 CONCLUSION

The two-player unknown NZS game is solved by a data-driven ADP algorithm based on the collected online data. The neural networks are constructed using the off-policy learning scheme to approach the optimal solution of the model-free iterative equation based on real system data. The application on a nonlinear 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 based on the on-policy scheme.

REFERENCES

- [1] P. Morris, Introduction to game theory. New York, NY, USA: Springer, 2012
- [2] H. Zhang, Q. Wei, D. Liu, An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games, *Automatica*, vol.47, no.1, 207-214, 2011.
- [3] D. Liu, H. Li, D. Wang, Online synchronous approximate optimal learning algorithm for multi-player non-zero-sum games with unknown dynamics, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol.44, no.8, 1015-1027, 2014.
- [4] Q. Zhang, D. Zhao, Y. Zhu, Event-triggered H_∞ control for continuous-time nonlinear system via concurrent learning, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, DOI: 10.1109/TSMC.2016.2531680, 2016.
- [5] 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 Transactions on Neural Networks and Learning Systems*, vol.27, no.2, 444-458, 2016.
- [6] Q. Zhang, D. Zhao, D. Wang, Event-based robust control for uncertain nonlinear systems using adaptive dynamic programming, *IEEE Transactions on Neural Networks and Learning Systems*, DOI: 10.1109/TNNLS.2016.2614002, 2016.
- [7] K. G. Vamvoudakis, F. L. Lewis, Multi-player non-zero-sum games: online adaptive learning solution of coupled Hamilton-Jacobi equations, *Automatica*, vol.47, no.8, 1556-1569, 2011.
- [8] D. Zhao, Q. Zhang, D. Wang, Y. Zhu, Experience replay for optimal control of nonzero-sum game systems with unknown dynamics, *IEEE Transactions on Cybernetics*, vol.46, no.3, 854-865, 2016.
- [9] Q. Wei, D. Liu, F. L. Lewis, Optimal distributed synchronization control for continuous-time heterogeneous multi-agent differential graphical games, *Information Sciences*, vol.317, 96-113, 2015.
- [10] K. G. Vamvoudakis, F. L. Lewis, G. R. Hudas, Multi-agent differential graphical games: Online adaptive learning solution for synchronization with optimality, *Automatica*, vol.48, 598-1611, 2012.
- [11] D. Vrabie, F. L. Lewis, Adaptive dynamic programming for online solution of a zero-sum differential game, *Journal of Control Theory and Applications*, vol.9, no.3, 353-360, 2011.
- [12] D. Vrabie, F. L. Lewis, Integral reinforcement learning for online computation of feedback nash strategies of nonzero-sum differential games, in *Proceeding of IEEE Conference ON Decision and Control (CDC)*, 3066-3071, 2010.
- [13] H. Wu, B. Luo, Neural network based online simultaneous policy update algorithm for solving the HJI equation in nonlinear control, *IEEE Transactions on Neural Networks and Learning Systems*, vol.23, no.12, 1884-1895, 2012.
- [14] Y. Jiang, Z. Jiang, Robust adaptive dynamic programming and feedback stabilization of nonlinear systems, *IEEE Transactions on Neural Networks and Learning Systems*, vol.25, no.5, 882-893, 2014.
- [15] J. Y. Lee, J. B. Park, Y. H. Choi, Integral reinforcement learning for continuous-time input-affine nonlinear systems with simultaneous invariant explorations, *IEEE Transactions on Neural Networks and Learning Systems*, vol.26, no.5, 916-932, 2015.
- [16] H. Li, D. Liu, D. Wang, Integral reinforcement learning for linear continuous-time zero-sum games with completely unknown dynamics, *IEEE Transactions on Automation Science and Engineering*, vol.11, no.3, 706C714, 2014.
- [17] B. Luo, H. Wu, T. Huang, Off-policy reinforcement learning for control design, *IEEE Transactions on Cybernetics*, vol.45, no.1, 65-76, 2015.
- [18] Y. Zhu, D. Zhao, X. Li, Iterative adaptive dynamic programming for solving unknown nonlinear zero-sum game based on online data, *IEEE Transactions on Neural Networks and Learning Systems*, DOI: 10.1109/TNNLS.2016.2561300, 2016.
- [19] M. Abu-Khalaf, F. L. Lewis, Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach, *Automatica*, vol.41, no.5, 779-791, 2005.
- [20] D. Liu, H. Li, D. Wang, Online synchronous approximate optimal learning algorithm for multi-player non-zero-sum games with unknown dynamics, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol.44, no.8, 1015-1027, 2014.
- [21] Y. Zhu, D. Zhao, X. Li, Iterative adaptive dynamic programming for solving unknown nonlinear zero-sum game based on online data, *IEEE Transactions on Neural Networks and Learning Systems*, DOI: 10.1109/TNNLS.2016.2561300, 2016.