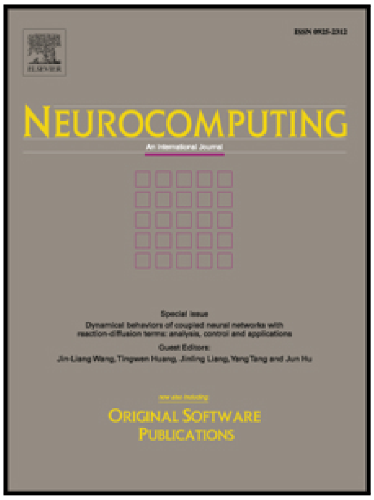
Communicated by Dr Hongyi Li

Accepted Manuscript



Neural-Network-Based Synchronous Iteration Learning Method for Multi-Player Zero-Sum Games

Ruizhuo Song, Qinglai Wei, Biao Song

PII: S0925-2312(17)30342-9
DOI: 10.1016/j.neucom.2017.02.051
Reference: NEUCOM 18125

To appear in: *Neurocomputing*

Received date: 29 August 2016
Revised date: 12 December 2016
Accepted date: 19 February 2017

Please cite this article as: Ruizhuo Song, Qinglai Wei, Biao Song, Neural-Network-Based Synchronous Iteration Learning Method for Multi-Player Zero-Sum Games, *Neurocomputing* (2017), doi: 10.1016/j.neucom.2017.02.051

Neural-Network-Based Synchronous Iteration Learning Method for Multi-Player Zero-Sum Games

Ruizhuo Song[1], Qinglai Wei[2], Biao Song[1]

[1] *School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, 100083, China*
[2] *The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China*

Abstract

In this paper, a synchronous solution method for multi-player zero-sum games without system dynamics is established based on neural network. The policy iteration (PI) algorithm is presented to solve the Hamilton-Jacobi-Bellman (HJB) equation. It is proven that the obtained iterative cost function is convergent to the optimal game value. For avoiding system dynamics, off-policy learning method is given to obtain the iterative cost function, controls and disturbances based on PI. Critic neural network (CNN), action neural networks (ANNs) and disturbance neural networks (DNNs) are used to approximate the cost function, controls and disturbances. The weights of neural networks compose the synchronous weight matrix, and the uniformly ultimately bounded (UUB) of the synchronous weight matrix is proven. Two examples are given to show that the effectiveness of the proposed synchronous solution method for multi-player ZS games.

Key words: Adaptive dynamic programming, Approximate dynamic programming, Adaptive critic designs, Multi-player, Iteration learning, Neural network
PACS:

Email address: `ruizhuosong@ustb.edu.cn` (Ruizhuo Song[1], Qinglai Wei[2], Biao Song[1]).

1 Introduction

The importance of strategic behavior in the human and social world is increasingly recognized in theory and practice. As a result, game theory has emerged as a fundamental instrument in pure and applied research [1]. Modern day society relies on the operation of complex systems, including aircraft, automobiles, electric power systems, economic entities, business organizations, banking and finance systems, computer networks, manufacturing systems, and industrial processes. Networked dynamical agents have cooperative team-based goals as well as individual selfish goals, and their interplay can be complex and yield unexpected results in terms of emergent teams. Cooperation and conflict of multiple decision-makers for such systems can be studied within the field of cooperative and noncooperative game theory [2]. It knows that many real-world systems are often controlled by more than one controller or decision maker with each using an individual strategy. These controllers often operate in a group with a general quadratic performance index function as a game. Therefore, some scholars research the multi-player games. In [3], off-policy integral reinforcement learning method was developed to solve nonlinear continuous-time multi-player non-zero-sum (NZS) games. In [4], a multi-player zero-sum (ZS) differential games for a class of continuous-time uncertain nonlinear systems were solved using upper and lower iterations. ZS game theory relies on solving the Hamilton-Jacobi-Isaacs (HJI) equations, a generalized version of the Hamilton-Jacobi-Bellman(HJB) equations appearing in optimal control problems. In the nonlinear case the HJI equations are difficult or impossible to solve, and may not have global analytic solutions even in simple cases. Therefore, many approximate methods are proposed to obtain the solution of HJI equations [5–8].

Adaptive dynamic programming (ADP) algorithm is an effective approximate method in optimal control field [9–13]. ADP algorithms include value iteration (VI) and policy iteration (PI) [14–17]. VI is a Lyapunov recursion, which is easy to implement and does not require Lyapunov equation solutions [18–21]. In [22], discrete-time VI was proposed to solve HJB equation approximately with convergence analysis. In [23], a novel non-model-based, data-driven adaptive optimal controller was presented by continuous-time VI. In [24], a class of continuous-time nonlinear two-player ZS differential games was considered, VI ADP method was proposed for the situations that the saddle point exists or does not exist. On the other hand, PI refers to a class of algorithms built as a two-step iteration: policy evaluation and policy improvement [25], starting from evaluating the performance index function of a given initial admissible (stabilizing) controller [26–28]. In [29], PI algorithm and convergence analysis were given for nonlinear systems with saturating actuators. In [30], optimal model-free output synchronization of heterogeneous systems using off-policy reinforcement learning was developed. In [31], a data-driven ADP method was proposed for a class of continuous-time unknown nonlinear systems ZS optimal control problems. In[32], an online solution method for two-player ZS games was presented by synchronous PI.

Although the progress on ADP algorithm is significant in the optimal control field, within the radius of our knowledge, it is still an open problem about how to solve multi-player ZS games for completely unknown continuous-time nonlinear systems. In this paper, this open problem will be explicitly figured out. The main contributions of this paper are summarized as follows.
1): A synchronous solution method based on PI algorithm and neural networks is established.
2): It is proven that the iterative cost function converges to the optimal game value with system dynamics for traditional PI algorithm.
3): Synchronous solution method is given to solve the off-policy HJB equation with convergence analysis, according to critic neural network (CNN), action neural networks (ANNs) and disturbance neural networks (DNNs).
4): The uniformly ultimately bounded (UUB) of the synchronous weight matrix is proven.

The rest of this paper is organized as follows. In Section 2, we present the motivations and preliminaries of the discussed problem. In Section 3, the synchronous solution of multi-player ZS games is developed and the convergence proof is given. In Section 4, two examples are given to demonstrate the effectiveness of the proposed scheme. In Section 5, the conclusion is drawn.

2 Motivations and Preliminaries

In this paper, we consider the continuous-time nonlinear system described by

$$\dot{x} = f(x) + g(x)\sum_{i=1}^{p} u_i + h(x)\sum_{j=1}^{q} d_j \tag{1}$$

where $x \in \Omega \in R^n$ is the system state, $u_i \in R^{m_1}$ and $d_j \in R^{m_2}$ are the control input and the disturbance input, respectively. $f(x) \in R^n$, $g(x)$ and $h(x)$ are unknown functions. $f(0) = 0$ and $x = 0$ is an equilibrium point of the system. Assume that $f(x)$, $g(x)$ and $h(x)$ are locally Lipschitz functions on the compact set Ω that contains the origin. The dynamical system is stabilizable on Ω. The performance index function is a generalized quadratic form given by

$$J(x(0), U_p, D_q) = \int_0^\infty \left\{x^T Q x + \sum_{i=1}^{p} u_i^T R_i u_i - \sum_{j=1}^{q} d_j^T S_j\, d_j\right\} dt \tag{2}$$

where Q, R_i and S_j are positive definite matrixes, $U_p = \{u_1, \cdots, u_p\}$ and $D_q = \{d_1, \cdots, d_q\}$. Then, we define the multi-player ZS differential game subject to (1) as

$$V^*(x(0)) = \inf_{u_1}\inf_{u_2}\cdots\inf_{u_p}\sup_{d_1}\sup_{d_2}\cdots\sup_{d_q} J(x(0), U_p, D_q) \tag{3}$$

The multi-player ZS differential game selects the minimizing player set U_p and the maximizing player set D_q such that the saddle point U_p^* and D_q^* satisfies the following inequalities:

$$J(x, U_p^*, D_q) \leq J(x, U_p^*, D_q^*) \leq J(x, U_p, D_q^*) \tag{4}$$

where $U_p^* = \{u_1^*, \cdots, u_p^*\}$ and $D_q^* = \{d_1^*, \cdots, d_q^*\}$.

In this paper, we assume that the multi-player optimal control problem has a unique solution if and only if the Nash condition holds [33]

$$V^*(x) = \inf_{U_p}\sup_{D_q} J(x, U_p, D_q) = \sup_{D_q}\inf_{U_p} J(x, U_p, D_q) \tag{5}$$

If we give the feedback policy $(U_p(x), D_q(x))$, then the value or cost of the policy is

$$V(x(t)) = \int_t^\infty \left\{x^T Q x + \sum\nolimits_{i=1}^p u_i^T R_i u_i - \sum\nolimits_{j=1}^q d_j^T S_j\, d_j\right\} dt \tag{6}$$

By using Leibniz's formula and differentiating, (6) has a differential equivalent. Then we can obtain the nonlinear ZS game Bellman equation, which is given in terms of the Hamiltonian function

$$\begin{aligned} &H(x, \nabla V, U_p, D_q) \\ &= x^T Q x + \sum\nolimits_{i=1}^p u_i^T R_i u_i - \sum\nolimits_{j=1}^q d_j^T S_j d_j \\ &\quad + \nabla V^T (f + g \sum\nolimits_{i=1}^p u_i + h \sum\nolimits_{j=1}^q d_j) \\ &= 0 \end{aligned} \tag{7}$$

where $\nabla V = \dfrac{\partial V}{\partial x}$. The stationary conditions are

$$\frac{\partial H}{\partial u_i} = 0, i = 1, 2, \cdots, p \tag{8}$$

and

$$\frac{\partial H}{\partial d_j} = 0, j = 1, 2, \cdots, q \tag{9}$$

According to (7), we have the optimal controls and the disturbances are

$$u_i^* = -\frac{1}{2} R_i^{-1} g^T \nabla V^*, i = 1, 2, \cdots, p \tag{10}$$

and

$$d_j^* = \frac{1}{2} S_j^{-1} h^T \nabla V^*, j = 1, 2, \cdots, q \tag{11}$$

From Bellman equation (7), we can derive V^* from the solution of the HJI equation

$$\begin{aligned} 0 = x^T Q x + \nabla V^T f - \frac{1}{4} \sum_{i=1}^{p} \nabla V^T g R_i^{-1} g^T \nabla V \\ + \frac{1}{4} \sum_{j=1}^{q} \nabla V^T h S_j^{-1} h^T \nabla V \end{aligned} \tag{12}$$

Note that if (12) is solved, then the optimal controls are obtained. In general case, the PI algorithm can be applied to get V^*. The algorithm implementation process is given in Algorithm 1.

Algorithm 1 PI for nonlinear multi-player ZS differential games

1: Start with stabilizing initial policies $u_1^{[0]}$, $u_2^{[0]}$, $\cdots$, $u_p^{[0]}$, and $d_1^{[0]}$, $d_2^{[0]}$, $\cdots$, $d_q^{[0]}$.

2: Let $k = 1, 2, 3, \cdots$, solve $V^{[k]}$ from

$$\begin{aligned} 0 = x^T Q x + \sum_{i=1}^{p} u_i^{[k]T} R_i u_i^{[k]} - \sum_{j=1}^{q} d_j^{[k]T} S_j d_j^{[k]} \\ + \nabla V^{[k]T} (f + g \sum_{i=1}^{p} u_i^{[k]} + h \sum_{j=1}^{q} d_j^{[k]}) \end{aligned} \tag{13}$$

3: Update control and disturbance using

$$u_i^{[k+1]} = -\frac{1}{2} R_i^{-1} g^T \nabla V^{[k]} \tag{14}$$

and

$$d_j^{[k+1]} = \frac{1}{2} S_j^{-1} h^T \nabla V^{[k]} \tag{15}$$

4: Let $k = k + 1$, return to Step 2 and continue.

The convergence of Algorithm 1 will be analyzed in the next theorem.

Theorem 1 *Define $V^{[k]}$ as in (13). Let control policy $u_i^{[k]}$ and disturbance policy $d_j^{[k]}$ be in (14) and (15), respectively. Then the iterative values $V^{[k]}$ converge to the optimal game values V^*, as $k \to \infty$.*

Định lý 1 chứng minh tính ổn định của hệ thống
Dùng Lyapunov

Proof: According to (13), we have

$$\dot{V}^{[k+1]} = -x^T Q x - \sum_{i=1}^{p} u_i^{[k+1]T} R_i u_i^{[k+1]} + \sum_{j=1}^{q} d_j^{[k+1]T} S_j d_j^{[k+1]} \tag{16}$$

Then

$$\begin{aligned}\dot{V}^{[k]} &= -x^T Q x - \sum_{i=1}^{p} u_i^{[k]T} R_i u_i^{[k]} + \sum_{j=1}^{q} d_j^{[k]T} S_j d_j^{[k]}\\ &\quad - \sum_{i=1}^{p} u_i^{[k+1]T} R_i u_i^{[k+1]} + \sum_{j=1}^{q} d_j^{[k+1]T} S_j d_j^{[k+1]}\\ &\quad + \sum_{i=1}^{p} u_i^{[k+1]T} R_i u_i^{[k+1]} - \sum_{j=1}^{q} d_j^{[k+1]T} S_j d_j^{[k+1]}\\ &= \dot{V}^{[k+1]} + \sum_{i=1}^{p} u_i^{[k+1]T} R_i u_i^{[k+1]} - \sum_{j=1}^{q} d_j^{[k+1]T} S_j d_j^{[k+1]}\\ &\quad - \sum_{i=1}^{p} u_i^{[k]T} R_i u_i^{[k]} + \sum_{j=1}^{q} d_j^{[k]T} S_j d_j^{[k]}\end{aligned} \tag{17}$$

By transformation, we have

$$\begin{aligned}\dot{V}^{[k]} &= \dot{V}^{[k+1]} - \sum_{i=1}^{p} (u_i^{[k+1]} - u_i^{[k]})^T R_i (u_i^{[k+1]} - u_i^{[k]})\\ &\quad +2\sum_{i=1}^{p} u_i^{[k+1]T} R_i (u_i^{[k+1]} - u_i^{[k]})\\ &\quad +\sum_{j=1}^{q} (d_j^{[k+1]} - d_j^{[k]})^T S_j (d_j^{[k+1]} - d_j^{[k]})\\ &\quad -2\sum_{i=1}^{p} d_j^{[k+1]T} S_j (d_j^{[k+1]} - d_j^{[k]})\end{aligned} \tag{18}$$

Let $\Delta u_i^{[k]} = u_i^{[k+1]} - u_i^{[k]}$ and $\Delta d_j^{[k]} = d_j^{[k+1]} - d_j^{[k]}$, then

$$\begin{aligned}\dot{V}^{[k]} &= \dot{V}^{[k+1]} - \sum_{i=1}^{p} \Delta u_i^{[k]T} R_i \Delta u_i^{[k]} + 2\sum_{i=1}^{p} u_i^{[k+1]T} R_i \Delta u_i^{[k]}\\ &\quad + \sum_{j=1}^{q} \Delta d_j^{[k]T} S_j \Delta d_j^{[k]} - 2\sum_{i=1}^{p} d_j^{[k+1]T} S_j \Delta d_j^{[k]}\end{aligned} \tag{19}$$

From (14) and (15), we have

$$\nabla V^{[k]T} g = -2u_i^{[k+1]T} R_i \tag{20}$$

and

$$\nabla V^{[k]T} h = 2d_j^{[k+1]T} S_j \tag{21}$$

Then (19) is expressed as

$$\begin{aligned}\dot{V}^{[k]} &= \dot{V}^{[k+1]} - \sum_{i=1}^{p} \Delta u_i^{[k]T} R_i \Delta u_i^{[k]} - \sum_{i=1}^{p} \Delta V^{[k]T} g \Delta u_i^{[k]}\\ &\quad + \sum_{j=1}^{q} \Delta d_j^{[k]T} S_j \Delta d_j^{[k]} - \sum_{i=1}^{p} \Delta V^{[k]T} h \Delta d_j^{[k]}\end{aligned} \tag{22}$$

Thus a sufficient conditions for $\dot{V}^{[k]} \leq \dot{V}^{[k+1]}$ are

$$\Delta u_i^{[k]T} R_i \Delta u_i^{[k]} - \Delta V^{[k]T} g \Delta u_i^{[k]} > 0 \tag{23}$$

and

$$\Delta d_j^{[k]T} S_j \Delta d_j^{[k]} - \Delta V^{[k]T} h \Delta d_j^{[k]} < 0 \tag{24}$$

Hence, if $\delta^H(S_j)||\Delta d_j^{[k]}|| \leq ||\Delta V^{[k]T} h||$ and $\Delta V^{[k]T} g \Delta u_i^{[k]} > 0$, or $\delta^H(S_j)||\Delta d_j^{[k]}|| \leq ||\Delta V^{[k]T} h||$ and $\delta_L(R_i)||\Delta u_i^{[k]}|| > ||\Delta V^{[k]T} g||$, where δ_L is the operator which takes the minimum singular value, and δ^H is the operator which takes the maximum singular value. Then $\dot{V}^{[k]} \leq \dot{V}^{[k+1]}$. The proof completes.

From Algorithm 1, we can see that the PI algorithm depends on system dynamics, which is unknown in this paper. Therefore, in the next section, off-policy PI algorithm will be presented which can solve the control and disturbance policies synchronously.

3 Synchronous Solution of Multi-player ZS Games

In this section, off-policy algorithm will be proposed based on Algorithm 1. The neural networks implementation process is also given. Based on that, the stability of the synchronous solution method is proven.

3.1 Derivation of off-policy algorithm

Let $u_i^{[k]}$ and $d_j^{[k]}$ be obtained by (14) and (15), then the original system (1) is rewritten as

$$\begin{aligned} \dot{x} = f + g \sum\nolimits_{i=1}^{p} u_i^{[k]} + h \sum\nolimits_{j=1}^{q} d_j^{[k]} \\ + g \sum\nolimits_{i=1}^{p} (u_i - u_i^{[k]}) + h \sum\nolimits_{j=1}^{q} (d_j - d_j^{[k]}) \end{aligned} \tag{25}$$

Substitute (25) into (6), we have

$$\begin{aligned} & V^{[k]}(x(t+T)) - V^{[k]}(x(t)) \\ & = \int_t^{t+T} \nabla V^{[k]T} \dot{x} d\tau \\ & = \int_t^{t+T} \nabla V^{[k]T} \left(f + g \sum\nolimits_{i=1}^{p} u_i^{[k]} + h \sum\nolimits_{j=1}^{q} d_j^{[k]} \right) d\tau \\ & \quad + \int_t^{t+T} \nabla V^{[k]T} \left(g \sum\nolimits_{i=1}^{p} (u_i - u_i^{[k]}) + h \sum\nolimits_{j=1}^{q} (d_j - d_j^{[k]}) \right) d\tau \end{aligned} \tag{26}$$

According to (13), (26) is

$$
\begin{aligned}
&V^{[k]}(x(t+T))-V^{[k]}(x(t))\\
&=-\int\limits_{t}^{t+T}\left(x^TQx+\sum\nolimits_{i=1}^{p}u_i^{[k]T}R_iu_i^{[k]}-\sum\nolimits_{j=1}^{q}d_j^{[k]T}\,S_jd_j^{[k]}\right)d\tau\\
&\quad+\int\limits_{t}^{t+T}\nabla V^{[k]T}\left(g\sum\nolimits_{i=1}^{p}(u_i-u_i^{[k]})+h\sum\nolimits_{j=1}^{q}(d_j-d_j^{[k]})\right)d\tau
\end{aligned}
\tag{27}
$$

Then (27) is the off-policy Bellman equation for multi-player ZS games, which is expressed as

$$
\begin{aligned}
&V^{[k]}(x(t+T))-V^{[k]}(x(t))\\
&=-\int\limits_{t}^{t+T}\left(x^TQx+\sum\nolimits_{i=1}^{p}u_i^{[k]T}R_iu_i^{[k]}-\sum\nolimits_{j=1}^{q}d_j^{[k]T}\,S_jd_j^{[k]}\right)d\tau\\
&\quad+\int\limits_{t}^{t+T}-2\Big(u_i^{[k+1]T}R_i\sum\nolimits_{i=1}^{p}(u_i-u_i^{[k]})\\
&\quad-d_j^{[k+1]T}S_j\sum\nolimits_{j=1}^{q}(d_j-d_j^{[k]})\Big)\,d\tau
\end{aligned}
\tag{28}
$$

It can be seen that (28) shows two points. First, the system dynamics is not necessary for obtaining $V^{[k]}$. Second, $u_i^{[k]}$, $d_j^{[k]}$ and $V^{[k]}$ can be obtained synchronously. In the next part, the implementation method for solving (28) will be presented.

3.2 Implementation method for off-policy algorithm

In this part, the method for solving off-policy Bellman equation (28) is given. Critic, action and disturbance networks are applied to approximate $V^{[k]}$, $u_i^{[k]}$ and $d_j^{[k]}$. The implementation block diagram is shown in Fig. 1. Here CNN, ANNs and DNNs are used to approximate the cost, control policies and disturbances.

In the neural network, if the number of hidden layer neurons is L, the weight matrix between the input layer and hidden layer is Y, the weight matrix between the hidden layer and output layer is W and the input vector of the neural network is X, then the output of three-layer neural network is represented by:

$$F_N(X,Y,W)=W^T\hat{\sigma}(YX), \tag{29}$$

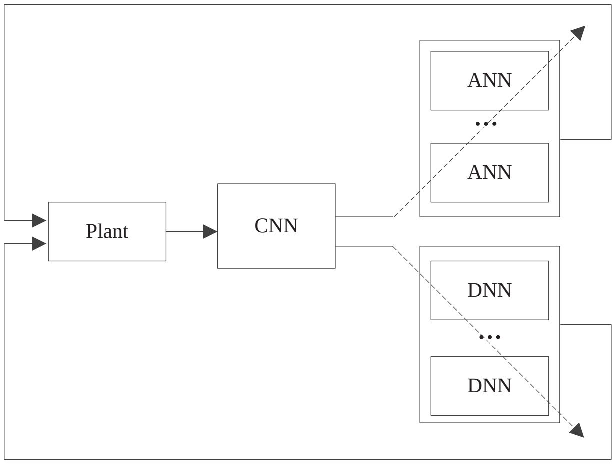


Fig. 1. Implementation block diagram

where $\hat{\sigma}(YX)$ is the activation function. For convenience of analysis, only the output weight W is updating during the training, while the hidden weight is kept unchanged. Hence, in the following part, the neural network function (29) can be simplified by the expression

$$F_N(X,W) = W^T\sigma(X). \tag{30}$$

The neural network expression of CNN is given as

$$V^{[k]}(x) = A^{[k]T}\phi_V(x) + \delta_V(x) \tag{31}$$

where $A^{[k]}$ is the ideal weight of critic network, $\phi_V(x)$ is the active function, and $\delta_V(x)$ is residual error. Let the estimation of $A^{[k]}$ is $\hat{A}^{[k]}$. Then the estimation of $V^{[k]}(x)$ is

$$\hat{V}^{[k]}(x) = \hat{A}^{[k]T}\phi_V(x) \tag{32}$$

and

$$\nabla\hat{V}^{[k]}(x) = \nabla\phi_V^T(x)\hat{A}^{[k]} \tag{33}$$

The neural network expression of ANN is

$$u_i^{[k]} = B_i^{[k]T}\phi_u(x) + \delta_u(x) \tag{34}$$

where $B_i^{[k]}$ is the ideal weight of action network, $\phi_u(x)$ is the active function, and $\delta_u(x)$ is residual error. Let $\hat{B}_i^{[k]}$ be the estimation of $B_i^{[k]}$, then the estimation of $u_i^{[k]}$ is

$$\hat{u}_i^{[k]} = \hat{B}_i^{[k]T}\phi_u(x) \tag{35}$$

The neural network expression of DNN is

$$d_j^{[k]} = C_j^{[k]T}\phi_d(x) + \delta_d(x) \tag{36}$$

where $C_j^{[k]}$ is the ideal weight of action network, $\phi_d(x)$ is the active function, and $\delta_d(x)$ is residual error. Let $\hat{C}_j^{[k]}$ be the estimation of $C_j^{[k]}$, then the estimation of $d_j^{[k]}$ is

$$\hat{d}_j^{[k]} = \hat{C}_j^{[k]T}\phi_d(x) \tag{37}$$

According to (28), we define the equation error as

$$\begin{aligned} e^{[k]} &= \hat{V}^{[k]}(x(t)) - \hat{V}^{[k]}(x(t+T)) \\ &\quad - \int_t^{t+T} \left(x^T Q x + \sum_{i=1}^{p} \hat{u}_i^{[k]T} R_i \hat{u}_i^{[k]} - \sum_{j=1}^{q} \hat{d}_j^{[k]T} S_j \hat{d}_j^{[k]}\right) d\tau \\ &\quad + \int_t^{t+T} -2\Big(\hat{u}_i^{[k+1]T} R_i \sum_{i=1}^{p} (u_i - \hat{u}_i^{[k]}) \\ &\quad -\hat{d}_j^{[k+1]T} S_j \sum_{j=1}^{q} (d_j - \hat{d}_j^{[k]})\Big)\, d\tau \end{aligned} \tag{38}$$

Therefore, substitute (32), (35) and (37) into (38), we have

$$\begin{aligned} e^{[k]} &= \hat{V}^{[k]}(x(t)) - \hat{V}^{[k]}(x(t+T)) \\ &\quad - \int_t^{t+T} \left(x^T Q x + \sum_{i=1}^{p} \hat{u}_i^{[k]T} R_i \hat{u}_i^{[k]} - \sum_{j=1}^{q} \hat{d}_j^{[k]T} S_j \hat{d}_j^{[k]}\right) d\tau \\ &\quad + \int_t^{t+T} -2\Big(\phi_u^T \hat{B}_i^{[k+1]} R_i \sum_{i=1}^{p} (u_i - \hat{u}_i^{[k]}) \\ &\quad - \phi_d^T \hat{C}_j^{[k+1]} S_j \sum_{j=1}^{q} (d_j - \hat{d}_j^{[k]})\Big)\, d\tau \end{aligned} \tag{39}$$

Since

$$\begin{aligned} &\phi_u^T \hat{B}_i^{[k+1]T} R_i \sum_{i=1}^{p} (u_i - \hat{u}_i^{[k]}) \\ &= \left(\left((\sum_{i=1}^{p} (u_i - \hat{u}_i^{[k]}))^T R_i\right) \otimes \phi_u^T\right) vec(\hat{B}_i^{[k+1]}) \end{aligned} \tag{40}$$

where $\otimes$ denotes kronecker product, and

$$\begin{aligned} &\phi_d^T \hat{C}_j^{[k+1]T} S_j \sum_{j=1}^{q} (d_j - \hat{d}_j^{[k]}) \\ &= \left(\left((\sum_{j=1}^{q} (d_j - \hat{d}_j^{[k]}))^T S_j\right) \otimes \phi_d^T\right) vec(\hat{C}_j^{[k+1]}) \end{aligned} \tag{41}$$

Substitute (40) and (41) into (39)

$$
\begin{aligned}
e^{[k]} = & \left((\phi_V(x(t)) - \phi_V(x(t+T)))^T \otimes I \right) \hat{A}^{[k]} \\
& - \int_t^{t+T} \left(x^T Q x + \sum\nolimits_{i=1}^{p} \hat{u}_i^{[k]T} R_i \hat{u}_i^{[k]} - \sum\nolimits_{j=1}^{q} \hat{d}_j^{[k]T} S_j \hat{d}_j^{[k]} \right) d\tau \\
& + \int_t^{t+T} -2 \left(\left(\left((\sum\nolimits_{i=1}^{p} (u_i - \hat{u}_i^{[k]}))^T R_i \right) \otimes \phi_u^T \right) vec(\hat{B}_i^{[k+1]}) \right. \\
& \left. - \left(\left((\sum\nolimits_{j=1}^{q} (d_j - \hat{d}_j^{[k]}))^T S_j \right) \otimes \phi_d^T \right) vec(\hat{C}_j^{[k+1]}) \right) d\tau
\end{aligned} \tag{42}
$$

Define

$$
\Pi_V = (\phi_V(x(t)) - \phi_V(x(t+T)))^T \otimes I \tag{43}
$$

$$
\Pi = \int_t^{t+T} \left(x^T Q x + \sum\nolimits_{i=1}^{p} \hat{u}_i^{[k]T} R_i \hat{u}_i^{[k]} - \sum\nolimits_{j=1}^{q} \hat{d}_j^{[k]T} S_j \hat{d}_j^{[k]} \right) d\tau \tag{44}
$$

$$
\Pi_u = \int_t^{t+T} -2 \left(\left(\left((\sum\nolimits_{i=1}^{p} (u_i - \hat{u}_i^{[k]}))^T R_i \right) \otimes \phi_u^T \right) d\tau \tag{45}
$$

$$
\Pi_d = -\int_t^{t+T} \left((\sum\nolimits_{j=1}^{q} (d_j - \hat{d}_j^{[k]}))^T S_j \right) \otimes \phi_d^T d\tau \tag{46}
$$

Then we have

$$
\begin{aligned}
e^{[k]} &= \Pi_V \hat{A}^{[k]} - \Pi + \Pi_u vec(\hat{B}_i^{[k+1]}) + \Pi_d vec(\hat{C}_j^{[k+1]}) \\
&= [\Pi_V \;\; \Pi_u \;\; \Pi_d] \begin{bmatrix} \hat{A}^{[k]} \\ vec(\hat{B}_i^{[k+1]}) \\ vec(\hat{C}_j^{[k+1]}) \end{bmatrix} - \Pi
\end{aligned} \tag{47}
$$

Define activation function matrix

$$
\Pi_\Pi = [\Pi_V \;\; \Pi_u \;\; \Pi_d] \tag{48}
$$

and the synchronous weight matrix

$$\hat{W}_{i,j}^{[k]} = \begin{bmatrix} \hat{A}^{[k]} \\ vec(\hat{B}_i^{[k+1]}) \\ vec(\hat{C}_j^{[k+1]}) \end{bmatrix} \tag{49}$$

Then (47) is

$$e^{[k]} = \Pi_\Pi \hat{W}_{i,j}^{[k]} - \Pi \tag{50}$$

Define $E^{[k]} = 1/2 e^{[k]T} e^{[k]}$, then according to gradient descent algorithm, the update method of the weight $\hat{W}_{i,j}^{[k]}$ is

$$\dot{\hat{W}}_{i,j}^{[k]} = -\eta_{i,j}^{[k]} \Pi_\Pi^T \left(\Pi_\Pi \hat{W}_{i,j}^{[k]} - \Pi \right) \tag{51}$$

where $\eta_{i,j}^{[k]}$ is a positive number.

According to gradient descent algorithm, the optimal weight $\hat{W}_{i,j}^{[k]}$ makes $E^{[k]}$ minimum, which can be obtained adaptively by (51). Therefore, the weights of critic, action and disturbance networks are solved simultaneously. In this proposed method, only one equation is necessary instead of (13)-(15) in Algorithm 1 to obtain the optimal solution for the multi-player ZS games.

3.3 Stability analysis

Theorem 2 *Let the update method for critic, action and disturbance networks be as in (51). Define the weight estimation error as $\tilde{W}_{i,j}^{[k]} = W_{i,j}^{[k]} - \hat{W}_{i,j}^{[k]}$, Then $\tilde{W}_{i,j}^{[k]}$ is UUB.*

Proof: Let Lyapunov function candidate be:

$$\Lambda_{i,j}^{[k]} = \frac{\alpha}{2\eta_{i,j}^{[k]}} \tilde{W}_{i,j}^{[k]T} \tilde{W}_{i,j}^{[k]}, \forall i, j, k \tag{52}$$

where $\alpha > 0$.

According to (51), we have

$$\begin{aligned} \dot{\tilde{W}}_{i,j}^{[k]} &= \eta_{i,j}^{[k]} \Pi_\Pi^T \left(\Pi_\Pi (W_{i,j}^{[k]} - \tilde{W}_{i,j}^{[k]}) - \Pi \right) \\ &= -\eta_{i,j}^{[k]} \Pi_\Pi^T \Pi_\Pi \tilde{W}_{i,j}^{[k]} + \eta_{i,j}^{[k]} \Pi_\Pi^T \Pi_\Pi W_{i,j}^{[k]} - \eta_{i,j}^{[k]} \Pi_\Pi^T \Pi \end{aligned} \tag{53}$$

Therefore, the gradient of (52) is

$$\begin{aligned}
\dot{\Lambda}_{i,j}^{[k]} &= \frac{\alpha}{\eta_{i,j}^{[k]}} \tilde{W}_{i,j}^{[k]T} \dot{\tilde{W}}_{i,j}^{[k]} \\
&= \alpha \tilde{W}_{i,j}^{[k]T} \left(-\Pi_\Pi^T \Pi_\Pi \tilde{W}_{i,j}^{[k]} + \Pi_\Pi^T \Pi_\Pi W_{i,j}^{[k]} - \Pi_\Pi^T \Pi \right) \\
&= -\alpha \tilde{W}_{i,j}^{[k]T} \Pi_\Pi^T \Pi_\Pi \tilde{W}_{i,j}^{[k]} + \alpha \tilde{W}_{i,j}^{[k]T} \Pi_\Pi^T \Pi_\Pi W_{i,j}^{[k]} - \alpha \tilde{W}_{i,j}^{[k]T} \Pi_\Pi^T \Pi \\
&\leq -\alpha ||\tilde{W}_{i,j}^{[k]}||^2 ||\Pi_\Pi||^2 + \alpha \tilde{W}_{i,j}^{[k]T} \Pi_\Pi^T \Pi_\Pi W_{i,j}^{[k]} - \alpha \tilde{W}_{i,j}^{[k]T} \Pi_\Pi^T \Pi \\
&\leq -\alpha ||\tilde{W}_{i,j}^{[k]}||^2 ||\Pi_\Pi||^2 + \frac{1}{2} ||\tilde{W}_{i,j}^{[k]}||^2 ||\Pi_\Pi||^2 + \frac{\alpha^2}{2} ||W_{i,j}^{[k]}||^2 ||\Pi_\Pi||^2 \\
&\quad + \frac{1}{2} ||\tilde{W}_{i,j}^{[k]}||^2 ||\Pi_\Pi||^2 + \frac{\alpha^2}{2} ||\Pi||^2
\end{aligned} \tag{54}$$

By transformation, (54) is

$$\dot{\Lambda}_{i,j}^{[k]} \leq (-\alpha + 1) ||\tilde{W}_{i,j}^{[k]}||^2 ||\Pi_\Pi||^2 + \frac{\alpha^2}{2} ||W_{i,j}^{[k]}||^2 ||\Pi_\Pi||^2 + \frac{\alpha^2}{2} ||\Pi||^2 \tag{55}$$

Define

$$\Sigma_{i,j}^{[k]} = \frac{\alpha^2}{2} ||W_{i,j}^{[k]}||^2 ||\Pi_\Pi||^2 + \frac{\alpha^2}{2} ||\Pi||^2 \tag{56}$$

Then (55) is

$$\dot{\Lambda}_{i,j}^{[k]} \leq (-\alpha + 1) ||\tilde{W}_{i,j}^{[k]}||^2 ||\Pi_\Pi||^2 + \Sigma_{i,j}^{[k]} \tag{57}$$

Thus, if

$$\alpha > 1 \tag{58}$$

and

$$||\tilde{W}_{i,j}^{[k]}||^2 > \frac{\Sigma_{i,j}^{[k]}}{(\alpha - 1) ||\Pi_\Pi||^2} \tag{59}$$

then $\tilde{W}_{i,j}^{[k]}$ is UUB. The proof completes.

According to Theorem 2, if the convergence condition is satisfied, then $\hat{V}^{[k]} \to V^{[k]}$, $\hat{u}_i^{[k]} \to u_i^{[k]}$ and $\hat{d}_j^{[k]} \to d_j^{[k]}$.

Remark 1 *This paper establishes a synchronous solution method based on PI algorithm to solve the multi-player zero-sum games. First, the method is different with other optimal control methods, such as the ones in [34], [35], [36] and [37]. In [34], an online reinforcement learning algorithm is proposed for a class of affine multiple input and multiple output (MIMO) nonlinear discrete-time systems with unknown functions and disturbances. In the paper, only two parameters are needed to be adjusted, and thus the number of the adaptation laws* *is smaller than the previous results. The updating parameters do not depend on the number of the subsystems for MIMO systems and the tuning rules are replaced by adjusting the norms on optimal weight vectors in both action and critic networks. In [35], an adaptive fuzzy optimal control design is addressed for a class of unknown nonlinear discrete-time systems. Fuzzy logic systems are employed to approximate the unknown functions in the systems. By applying the backsteppping design technique, a reinforcement learning algorithm is used to develop an optimal control signal. The adaptation auxiliary signal for unknown dead-zone parameters is established to compensate for the effect of nonsymmetric dead-zone on the control performance. In [36], an optimal control scheme-based adaptive neural network design for a class of unknown nonlinear discrete-time systems is proposed. The systems are transformed into an output predictor form. For the output predictor, the ideal control signal and the strategic utility function can be approximated by using an action network and a critic network, respectively. In [37], the optimal tracking control problem for the Henon Mapping chaotic system is solved using the direct heuristic dynamic programming setting with filtered tracking error. The fuzzy logic system is used to approximate the long-term utility function.*

Second, the method is different with the existing adaptive control method, such as the ones in [38], [39] and [40]. In [38], an approximation-based adaptive tracking control approach is proposed for a class of MIMO nonlinear systems. By introducing Nussbaum function, the issue of unknown control directions is handled. In the backstepping design process, the dynamic surface control technique is employed to avoid differentiating certain nonlinear functions repeatedly. Neural networks approximate the desired control signals directly. In [39], the adaptive neural network controller design is proposed for nonlinear MIMO discrete-time systems. In [40], an adaptive neural network tracking control method for uncertain nonlinear discrete-time systems with nonaffine dead-zone input is presented.

Therefore, this paper is the first time to discuss the solution method for the multi-player zero-sum games with unknown system dynamics.

4 Simulation Study

In this section, two examples will be provided to demonstrate the effectiveness of the optimal control scheme proposed in this paper.

4.1 Example 1

Consider the following linear system [41] with modifications

$$\dot{x} = x + u + d \tag{60}$$

In this paper, the initial state is $x(0) = 1$. We select hyperbolic tangent functions as the activation functions of critic, action and disturbance networks. The structures of critic, action and disturbance networks are $1-8-1$. The initial weight W is selected arbitrarily from $(-1, 1)$, the dimension of W is 24×1. For the cost function, Q, R and S in the utility function are identity matrices of appropriate dimensions. After 500 time steps, the simulation results are obtained. In Fig. 2, the cost function is shown, which converges to zero as time increasing. The control and disturbance trajectories are given in Figs. 3 and 4. Under the action of the obtained control and disturbance inputs, the state trajectory is displayed in Fig. 5. It is clear that the presented method in this paper is very effective and feasible.

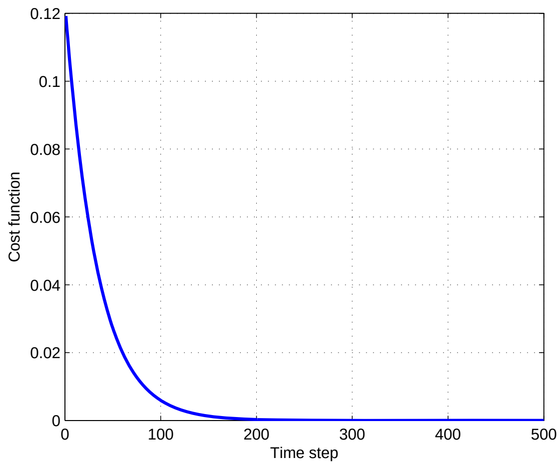


Fig. 2. Cost function

4.2 Example 2

Consider the following affine in control input nonlinear system [42]

$$\dot{x} = f(x) + g(x)\sum_{i=1}^{p} u_i + h(x)\sum_{j=1}^{q} d_j \tag{61}$$

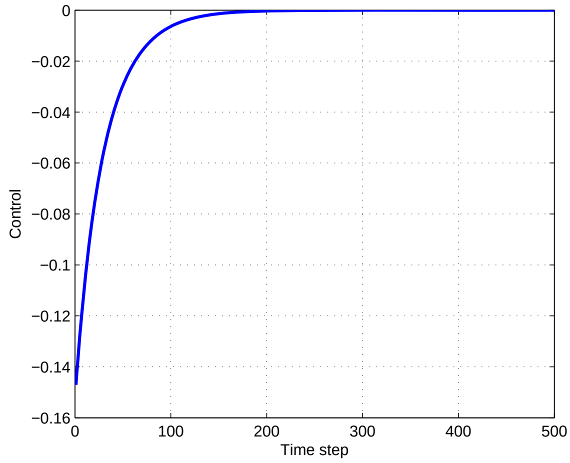


Fig. 3. Control

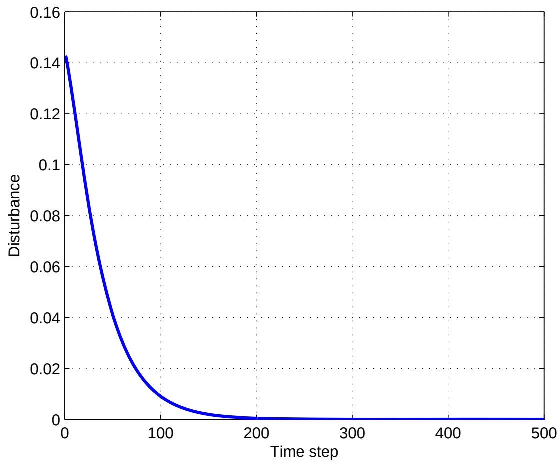


Fig. 4. Disturbance

where $f(x) = \begin{bmatrix} x_2 \\ -x_2 - \frac{1}{2}x_1 + \frac{1}{4}x_2(\cos(2x_1)+2)^2 + \frac{1}{4}x_2(\sin(4x_1^2)+2)^2 \end{bmatrix}$, $g(x) = \begin{bmatrix} 0 \\ \cos(2x_1)+2 \end{bmatrix}$, $h(x) = \begin{bmatrix} 0 \\ \sin(4x_1^2)+2 \end{bmatrix}$, $p = q = 1$.

In this simulation, the initial state is $x(0) = [1, -1]^T$. Hyperbolic tangent functions are used to be as the activation functions of critic, action and disturbance networks. The structures of the networks are $2-8-1$. The initial weight W is selected arbitrarily from $(-1, 1)$, the dimension of W is 24×1. For the cost function of (61), Q, R and S in the utility function are identity matrices of appropriate dimensions. The simulation results are obtained by 2500

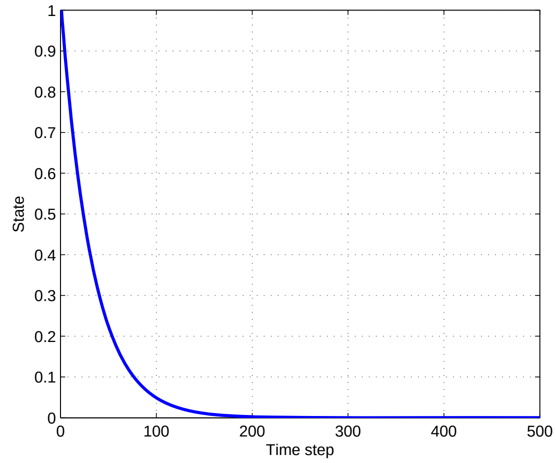


Fig. 5. State

time steps. The cost function is shown in Fig. 6, it is zero-sum. The control and disturbance trajectories are given in Figs. 7 and 8. The state trajectories are displayed in Fig. 9. We can see that the closed-loop system state, control and disturbance inputs converge to zero, as time step increasing. So the proposed synchronous method for multi-player zero-sum games in this paper is very effective.

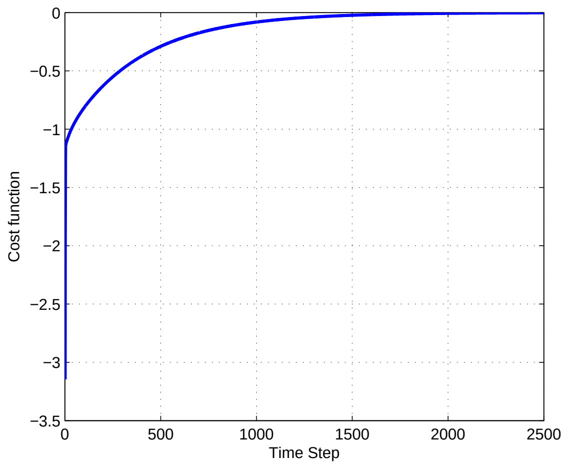


Fig. 6. Cost function

5 Conclusions

This paper proposed a synchronous solution method for multi-player zero-sum games without system dynamics based on neural network. PI algorithm is presented to solve the HJB equation with system dynamics. It is proven

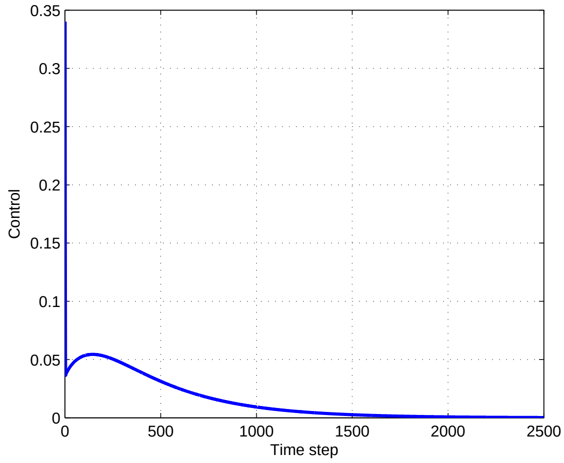


Fig. 7. Control

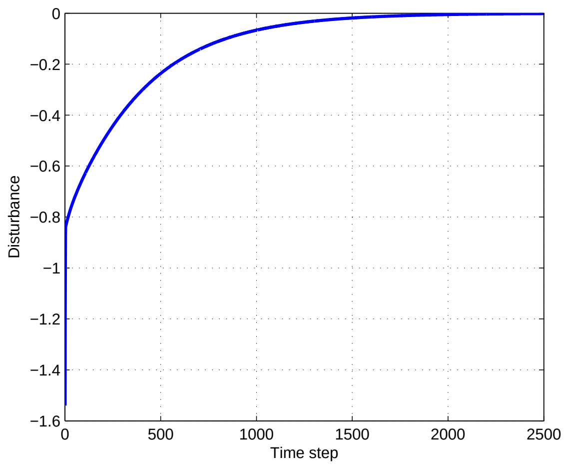


Fig. 8. Disturbance

that the obtained iterative cost function by PI is convergent to optimal game value. Based on PI, off-policy learning method is given to obtain the iterative cost function, controls and disturbances. The weights of CNN, ANNs and DNNs compose synchronous weight matrix, which is proven to be UUB by Lyapunov technique. Simulation study indicates the effectiveness of the proposed synchronous solution method for multi-player ZS games. A future research problem is to use the proposed approach to a class of systems with interconnection term.

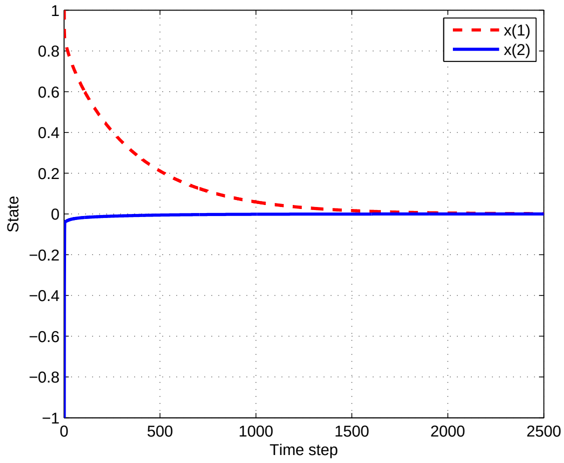


Fig. 9. State

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 61304079, 61673054, in part by Fundamental Research Funds for the Central Universities under Grant FRF-TP-15-056A3, and in part by the Open Research Project from SKLMCCS under Grant 20150104.

References

[1] D. W. K. Yeung and L. A. Petrosyan, Cooperative Stochastic Differential Games, Springer, 2006.

[2] F. Lewis, D. Vrabie, and V. L. Syrmos, Optimal Control, Third Edition, Wiley, 2012.

[3] R. Song, F. Lewis, and Q. Wei, Off-Policy Integral Reinforcement Learning Method to Solve Nonlinear Continuous-Time Multi-Player Non-Zero-Sum Games, IEEE Transactions on Neural Networks and Learning Systems, DOI: 10.1109/ TNNLS. 2016. 2582849.

[4] D. Liu and Q. Wei, Multiperson zero-sum differential games for a class of uncertain nonlinear systems, International Journal of Adaptive Control and Signal Processing, 28(3-5) (2014) 205–231.

[5] C. Mu, C. Sun, A. Song, and H. Yu, Iterative GDHP-based approximate optimal tracking control for a class of discrete-time nonlinear systems, Neurocomputing, http://dx.doi.org/10.1016/j.neucom.2016.06.059, 2016

[6] X. Fang, D. Zheng, H. He, and Z. Ni, Data-driven heuristic dynamic programming with virtual reality, Neurocomputing, 166(20) (2015) 244–255.

[7] T. Feng, H. Zhang, Y. Luo, and J. Zhang, Stability analysis of heuristic dynamic programming algorithm for nonlinear systems, Neurocomputing, 149(Part C, 3) (2015) 1461–1468.

[8] T. Feng, H. Zhang, Y. Luo, and H. Liang, Globally optimal distributed cooperative control for general linear multi-agent systems, Neurocomputing, 203(26) (2016) 12–21.

[9] H. Zhang, C. Qing, and Y. Luo, Neural-network-based constrained optimal control scheme for discrete-time switched nonlinear system using dual heuristic programming, IEEE Transactions on Automation Science and Engineering, 11(3) (2014) 839–849.

[10] W. Gao, Y. Jiang, Z. Jiang, and T. Chai, Output-feedback adaptive optimal control of interconnected systems based on robust adaptive dynamic programming, Automatica, 72 (2017) 37–45.

[11] Q. Wei, H. Zhang, and J. Dai, Model-free multiobjective approximate dynamic programming for discrete-time nonlinear systems with general performance index functions, Neurocomputing, 72(7) (2009) 1839–1848.

[12] X. Yang, D. Liu, Q. Wei, and D. Wang, Guaranteed cost neural tracking control for a class of uncertain nonlinear systems using adaptive dynamic programming, Neurocomputing, 198(19) (2016) 80–90.

[13] T. Wang, H. Zhang, and Y. Luo, Infinite-time stochastic linear quadratic optimal control for unknown discrete-time systems using adaptive dynamic programming approach, Neurocomputing, 171(1) (2016) 379–386.

[14] Y. Tang, H. He, Z. Ni, X. Zong, D. Zhao, and X. Xu, Fuzzy-based goal representation adaptive dynamic programming, IEEE Transactions on Fuzzy Systems, DOI: 10.1109/TFUZZ.2015.2505327, 2016.

[15] Q. Wei, F. Wang, D. Liu, and X. Yang, Finite-approximation-error based discrete-time iterative adaptive dynamic programming, IEEE Transactions on Cybernetics, 44(12) (2014) 2820–2833.

[16] D. Liu and Q. Wei, Finite-approximation-error-based optimal control approach for discrete-time nonlinear systems, IEEE Transactions on Cybernetics, 43(2) (2013) 779–789.

[17] T. L. Nguyen, Adaptive dynamic programming-based design of integrated neural network structure for cooperative control of multiple MIMO nonlinear systems, Neurocomputing, http://dx.doi.org/10.1016/j.neucom.2016.05.044, 2016.

[18] R. Song, H. Zhang, Y. Luo, and Q. Wei, Optimal Control Laws for Time-Delay Systems with Saturating Actuators Based on Heuristic Dynamic Programming, Neurocomputing, 73(16-18) (2010) 3020–3027.

[19] R. Song, W. Xiao, and H. Zhang, Multi-objective optimal control for a class of unknown nonlinear systems based on finite-approximation-error ADP algorithm, Neurocomputing, 119(7) (2013) 212–221.

[20] R. Song, Q. Wei, and Q. Sun, Nearly finite-horizon optimal control for a class of nonaffine time-delay nonlinear systems based on adaptive dynamic programming, Neurocomputing, 156(25) (2015) 166–175.

[21] C. Qin, H. Zhang, Y. Wang, and Y. Luo, Neural network-based online H control for discrete-time affine nonlinear system using adaptive dynamic programming, Neurocomputing, 198(19) (2016) 91–99.

[22] A. Al-Tamimi, F. Lewis, and M. Abu-Khalaf, Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 38(4) (2008) 943–949.

[23] T. Bian and Z. Jiang, Value iteration and adaptive dynamic programming for data-driven adaptive optimal control design, Automatica, 71 (2016) 348–360.

[24] 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.

[25] J. Wang, X. Xu, D. Liu, Z. Sun, and Q. Chen, Self-learning cruise control using kernel-based least squares policy iteration, IEEE Transactions on Control Systems Technology, 22(3) (2014) 1078–1087.

[26] D. Vrabie, O. Pastravanu, F. Lewis, and M. Abu-Khalaf, Adaptive optimal control for continuous-time linear systems based on policy iteration, Automatica, 45(2) (2009) 477–484.

[27] R. Song, W. Xiao, H. Zhang, and C. Sun, Adaptive dynamic programming for a class of complex-valued nonlinear systems, IEEE Transactions on Neural Networks and Learning Systems, 25(9) (2014) 1733–1739.

[28] R. Song, F. Lewis, Q. Wei, H. Zhang, Z. Jiang, and D. Levine, Multiple actor-critic structures for continuous-time optimal control using input-output data, IEEE Transactions on Neural Networks and Learning Systems, 26(4) (2015) 851–865.

[29] M. Abu-Khalaf and F. Lewis, Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach, Automatica, 41 (2005) 779–791.

[30] H. Modares, S. Nageshrao, G. Lopes, R. Babuška, and F. Lewis, Optimal model-free output synchronization of heterogeneous systems using off-policy reinforcement learning, Automatica, 71 (2016) 334–341.

[31] Q. Wei, R. Song, and 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, 27(2) (2016) 444–458.

[32] K. Vamvoudakis and F. Lewis, Online solution of nonlinear twoplayer zero-sum games using synchronous policy iteration, International Journal of Robust Nonlinear Control, 22(13) (2012) 1460–1483.

[33] F. Lewis, D. Vrabie, and V. L. Syrmos, Optimal Control. NewYork, NY, USA: Wiley, 2012.

[34] Y. J. Liu, L. Tang, S. C. Tong, C. L. P. Chen, and D. J. Li, Reinforcement learning design-based adaptive tracking control with less learning parameters for nonlinear discrete-time MIMO systems, IEEE Transactions on Neural Networks and Learning Systems, 26(1) (2015) 165–176.

[35] Y. J. Liu, Y. Gao, S. C. Tong and Y. M. Li, Fuzzy approximation-based adaptive backstepping optimal control for a class of nonlinear discrete-time systems with dead-zone, IEEE Transactions on Fuzzy Systems, 24(1) (2016) 16–28.

[36] Y. J. Liu and S. C. Tong, Optimal control-based adaptive NN design for a class of nonlinear discrete-time block-triangular systems, IEEE Transactions on Cybernetics, 46(11) (2016) 2670–2680.

[37] Y. Gao and Y. J. Liu, Adaptive fuzzy optimal control using direct heuristic dynamic programming for chaotic discrete-time system, Journal of Vibration and Control, 22(2) (2014) 595–603.

[38] Q. Zhou, P. Shi, Y. Tian and M. Wang, Approximation-Based Adaptive Tracking Control for MIMO Nonlinear Systems With Input Saturation. IEEE Transactions on Cybernetics, 45(10) (2015) 2119–2128.

[39] Y. J. Liu, L. Tang, S. C. Tong and C. L. P. Chen, Adaptive NN controller design for a class of nonlinear MIMO discrete-time systems, IEEE Transactions on Neural Networks and Learning Systems, 26(5) (2015) 1007–1018.

[40] Y. J. Liu and S. C. Tong, Adaptive neural network tracking control of uncertain nonlinear discrete-time systems with nonaffine dead-zone input, IEEE Transactions on Cybernetics, 45(3) (2015) 497–505.

[41] T. Basar and G. Olsder, Dynamic Noncooperative Game Theory. New York: Academic, 1982.

[42] K. G. Vamvoudakis and F. Lewis, Multi-player non-zero-sum games: Online adaptive learning solution of coupled Hamilton-Jacobi equations, Automatica, 47(8) (2011) 1556–1569.

Ruizhuo Song received the Ph.D. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2012. She was a postdoctoral fellow with University of Science and Technology Beijing, Beijing, China. She is currently an Associate Professor with the School of Automation and Electrical Engineering, University of Science and Technology Beijing. She was a Visiting Scholar with the Department of Electrical Engineering at University of Texas at Arlington, Arlington, TX, USA, from 2013 to 2014. Her current research interests include optimal control, neural-networks-based control, nonlinear control, wireless sensor networks, and adaptive dynamic programming and their industrial application. She has published over 40 journal and conference papers, and coauthored 2 monographs.

Qinglai Wei received the B.S. degree in Automation, and the Ph.D. degree in control theory and control engineering, from the Northeastern University, Shenyang, China, in 2002 and 2009, respectively. From 2009–2011, he was a postdoctoral fellow with The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. He is currently a Professor of the institute. He has authored three books, and published over 60 international journal papers. His research interests include adaptive dynamic programming, neural-networks-based control, optimal control, nonlinear systems and their industrial applications.

Dr. Wei is an Associate Editor of IEEE Transaction on Systems Man, and Cybernetics: Systems since 2016, Information Sciences since 2016, Neurocomputing since 2016, Optimal Control Applications and Methods since 2016, Acta Automatica Sinica since 2015, and has been holding the same position for IEEE Transactions on Neural Networks and Learning Systems during 2014–2015. He is the Secretary of IEEE Computational Intelligence Society (CIS) Beijing Chapter since 2015. He was Registration Chair of the 12th World Congress on Intelligent Control and Automation (WCICA2016), 2014 IEEE World Congress on Computational Intelligence (WCCI2014), the 2013 International Conference on Brain Inspired Cognitive Systems (BICS 2013), and the Eighth International Symposium on Neural Networks (ISNN 2011). He was the Publication Chair of 5th International Conference on Information Science and Technology (ICIST2015) and the Ninth International Symposium on Neural Networks (ISNN 2012). He was the Finance Chair of the 4th International Conference on Intelligent Control and Information Processing (ICICIP 2013) and the Publicity Chair of the 2012 International Conference on Brain Inspired Cognitive Systems (BICS 2012). He was guest editors for several international journals. He was a recipient of Shuang-Chuang Talents in Jiangsu Province, China, in 2014. He was a recipient of the Outstanding Paper Award of Acta Automatica Sinica in 2011 and Zhang Siying Outstanding Paper Award of Chinese Control and Decision Conference (CCDC) in 2015. He was a recipient of Young Researcher Award of Asia Pacific Neural Network Society (APNNS) in 2016.

Biao Song received the B.S. degree in electronic information engineering from Yanbian University, Yanbian, China, in 2011. He is currently a Ph.D. student with the School of Automation and Electrical Engineering, University of Science and Technology Beijing. His research interests include wireless sensor networks, adaptive dynamic programming.