Learning-based control for discrete-time constrained nonzero-sum games

Chaoxu Mu | Jiangwen Peng | Yufei Tang

Abstract
A generalized policy-iteration-based solution to a class of discrete-time multi-player nonzero-sum games concerning the control constraints was proposed. Based on initial admissible control policies, the iterative value function of each player converges to the optimum approximately, which is structured by the iterative control policies satisfying the Nash equilibrium. Afterwards, the stability analysis is shown to illustrate that the iterative control policies can stabilize the system and minimize the performance index function of each player. Meanwhile, neural networks are implemented to approximate the iterative control policies and value functions with the impact of control constraints. Finally, two numerical simulations of the discrete-time two-player non-zero-sum games for linear and non-linear systems are shown to illustrate the effectiveness of the proposed scheme.

1 | INTRODUCTION

Reinforcement learning, one of the popular research fields in machine learning, is devoted to constructing an interactive bridge which lets an agent be contacted with the environment to achieve a goal. The interactive bridge relies on the influence between an agent and the environment through a series of operations that an agent acts based on the reward supplied by the environment and the environment states are changed by the action [1–4]. Related to reinforcement learning and optimal control, Werbos advocated adaptive dynamic programming (ADP) for the first time [5]. Different from dynamic programming (DP), the traditional optimal control solution, it solves the optimal control problem forward-in-time rather than backwards, avoiding the difficulty brought by the dimensional explosion in the traditional method for high-dimensional and complex systems. The actor-critic structure is the basic one in ADP algorithm where the actor structure chooses control policy for the system based on the perceived environment and the critic structure improves control policy by system performance evaluation. So that an agent can interact with the environment and it will converge to the approximate optimal point in the continuous iteration for the optimal control problem [6–8].

It is shared for the iterative method in ADP algorithm which is divided into policy iteration (PI) and value iteration (VI) algorithm. VI algorithm consists of two parts which are value update and policy improvement. Meanwhile, PI algorithm also consists of two parts which are policy evaluation and policy improvement. As for VI algorithm, the convergence speed is low for the reason that the iterative control policies may not be admissible ones in the iteration process. By contrast, they are all admissible control policies in the policy iteration procedure with the problem of computational complexity [9,10]. In order to find the balance on the properties of PI and VI algorithm, generalized policy iteration algorithm (GPI) is put forward in [5] for the first time. Besides, PI algorithm and VI algorithm are all special forms in the GPI algorithm [11]. Even though the idea of GPI algorithm has been put forward for many years, the proof of stability is not given until the appearance of [11]. As for the actor-critic structure and iterative method, it becomes an efficient and simple approximate optimal method for high-dimensional and complex systems with the effort of many researchers, such as multi-agent optimal control [12,13], zero-sum game (NZ) [14,15], non-zero-sum game (NZS) [10,16].

NZS game is a complex and difficult optimal control problem for multiplayer which is what we called multiple controllers. Different from other optimal control problems, which should guarantee the stability of the system and the...
minimum of performance index function for each player, it also needs to meet Nash equilibrium formed by a sequence of control inputs. Most of the discussion is on how to solve the coupled Hamilton-Jacobi equation for the Nash equilibrium solution. It can be seen that many researchers paid attention to the continuous-time NZS games. An online ADP method based on policy iteration for the continuous-time multi-player NZS game was presented in [17]. Afterwards, a near-optimal control was put forward for the continuous-time non-linear system using just a critic network rather than an actor-critic dual network in [18]. More recent studies which present an off-policy learning and experience replay method used for continuous system are aroused in [19,20] respectively. It is obvious that there are few pieces of research on discrete-time multi-player NZS game except [10]. Meanwhile, it is widespread for control constraints in control systems with the damage of system stability and performance such as [21–23]. Therefore, there is a tendency to form a unified statement on discrete-time multi-player NZS game solved by the idea of ADP algorithm, which motivates our drawing on the idea of GPI algorithm considering control constraints.

The contributions of this paper are provided as follows. In this paper, the GPI algorithm is the first time to solve the discrete-time multi-player NZS games considering control constraints. Benefit from the idea of the GPI algorithm, compared with the traditional PI and VI algorithms, is unified solution to the discrete-time multi-player NZS games. In addition, the proofs of convergence and optimality for the control-constraint multi-player NZS games is provided in detail. Next, simulations of the linear system and non-linear system are provided respectively to show the effectiveness of the proposed scheme. In the end, based on statistic technique, it is the first time to prove that the GPI algorithm has slightly less energy loss and greatly less needed runtime compared with the traditional PI and VI algorithms.

The rest of article is organized as follows. In Section 2, the background of discrete-time control-constraint multi-player NZS games is presented in detail. In Section 3, the GPI algorithm for discrete-time control-constraint multi-player NZS game is developed. Besides, stability is demonstrated. Meanwhile, the actor-critic neural-network structure is shown. In Section 4, two numerical examples to illustrate the effectiveness of the proposed scheme. In Section 5, a brief conclusion is provided.

2 PRELIMINARIES

Consider a class of discrete-time multi-player NZS games with control constraints described by

\[
x_{k+1} = f(x_k) + \sum_{q=1}^{N} g_q(x_k)\bar{u}_q(k)
\]

where \(x_k = [x_{1k}; x_{2k}; \ldots; x_{nk}]^T \in \mathbb{R}^n\) is the state vector, \(\bar{u}_q(k)\) is the control vector, \(f(x_k) \in \mathbb{R}^n\) denotes the state matrix and \(g_q(x_k) \in \mathbb{R}^{m \times m}\) denote the input matrices. Let \(\bar{u}_q(k) \in \{\bar{u}_q | \bar{u}_q \in \mathbb{R}^m, | \bar{u}_q | \leq \nu_q, q = 1, 2, \ldots, N\}\) be the control inputs that satisfy control constraints, where \(\nu_q\) represents saturation bound of the qth actuator. In the following sections, \(x_{k+1}\) is also written as \(\bar{F}(x_k, \bar{u}_q(k))\).

The performance index function of the qth player is presented as

\[
J_q(x_0, \bar{u}_1, \bar{u}_2, \ldots, \bar{u}_N) = \sum_{k=0}^{\infty} U_q(x_k, \bar{u}_q(k), \bar{u}_q(k)),
\]

where

\[
U_q(x_k, \bar{u}_q(k), \bar{u}_q(k)) = Q_q(x_k) + \bar{u}_q^T(k)M_{qq}\bar{u}_q(k) + \sum_{j=1,j \neq q}^{N} \bar{u}_j^T(k)M_{qj}\bar{u}_j(k);
\]

\[
\bar{u}_q = \{\bar{u}_j | j = 1, 2, \ldots, N, j \neq q\}.
\]

Let \(U_q(x_k, \bar{u}_q(k), \bar{u}_q(k)) > 0\) for \(\forall x_k, \bar{u}_q(k), \bar{u}_q(k) \neq 0\) be the utility function and \(Q_q(x_k), M_{qq}, M_{qj}\) be the positive definite symmetric matrices.

In this paper, a set of control policies \(\{\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_N\}\), which is devoted to stabilizing the system (1) and minimizing the performance index function (2), is considered as admissible.

Definition 1 (Admissible Control) Let \(\Omega_\mu\) denote the set of admissible control policies. If the control policy \(\bar{u}_q\) for \(\forall q\) can be regarded as an admissible one with respect to Equation (2) on \(\Omega_\mu\), that is, \(\bar{u}_q \subseteq \Omega_\mu\), it should satisfy the following conditions.

1. \(\bar{u}_q\) should be continuous on \(\Omega_\mu\);
2. \(\bar{u}_q(x)|_{x=0} = 0\);
3. \(\bar{u}_q\) stabilizes the system (Equation 1) on \(\Omega_\mu\) and makes the performance index function (Equation 2) finite simultaneously for \(\forall x_0 \subseteq \Omega_\mu\).

Given the admissible control policies of the qth player \(\bar{u}_q\), the optimal performance index function can be expressed as

\[
J_q^*(x_0, \bar{u}_1, \bar{u}_2, \ldots, \bar{u}_N) = \inf_{\bar{u}_q} \{J_q(x_0, \bar{u}_1, \bar{u}_2, \ldots, \bar{u}_N)\}.
\]

Assume that the optimal control policy of the qth player \(\bar{u}_q^*(k)\) can be expressed as

\[
\bar{u}_q^*(k) = \arg \inf_{\bar{u}_q} \{J_q(x_0, \bar{u}_1, \bar{u}_2, \ldots, \bar{u}_N)\}.
\]

It should be pointed out that each player has equivalent competitiveness and arrives at a Nash equilibrium as the following definition.

Definition 2 (Nash Equilibrium) A set of control policies \(\{\bar{u}_1^*, \bar{u}_2^*, \ldots, \bar{u}_N^*\}\) satisfies that
\[ J_q(\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_i, \ldots, \bar{u}_N) \leq J_q(\bar{u}'_1, \bar{u}'_2, \ldots, \bar{u}'_i, \ldots, \bar{u}'_N), \]  
which is deemed a Nash equilibrium solution.

Let the value function of the \( q \)th player be

\[ V_q(x) = \sum_{k} U_q(x, \bar{u}_q(k), \bar{u}_q(k)). \]  

Hence, the multi-player NZS game can be expressed as

\[ V_q^*(x) = \min_{\bar{u}_q(k)} \sum_{k} U_q(x, \bar{u}_q(k), \bar{u}_q(k)). \]  

In order to satisfy Equation (9) and control constraints [22], Equation (6) can be expressed as

\[ \bar{u}_q^*(k) = -\nu_q \tanh \left( \frac{1}{2}(\nu_q M_q) - \frac{1}{2} \delta_q V_q(x) \right), \]  

where \( \nabla V_q(x_k) = \frac{\partial V_q^*(x_k)}{\partial x_k} \). Then, Equation (9) can be derived as

\[ V_q^*(x) = U_q(x, \bar{u}_q^*(x), \bar{u}_q^*(x)) + V_q(x_{k+1}), \]  

which is known as the coupled NZS game Bellman optimality equation [10].

3 | GENERALIZED POLICY ITERATION FOR CONTROL-CONSTRAINT NONZERO-SUM GAME

3.1 | Generalized policy iteration algorithm

There are two iterations in the GPI algorithm, which is different from the traditional policy iteration and value iteration algorithm. Meanwhile, two iteration indices \( i \) and \( j \) related to two iterations are presented, which increase from 1 and 0, respectively.

Let the iterative value function of the \( q \) th player \( V_q^{ij}(x_k) \) constructed by \( \bar{v}_q^j(x_k) \) satisfy Equation (11):

\[ V_q^{ij}(x_k) = U_q(x_k, \bar{v}_q^j(x_k), \bar{v}_q^j(x_k)) + V_q^{ij}(F(x_k, \bar{v}_q^j(x_k))), \]  

Define the initial control policy of the \( q \) th player \( \bar{v}_q^j(x_k) \) as an admissible one, which means \( \bar{v}_q^j(x_k) \in \Omega_v \). Hence, for all \( x_k \in \Omega_v \), \( V_q^j(x_k) \) is an initial value function constructed by \( \bar{v}_q^j(x_k) \) which satisfies Equation (11):

\[ V_q^j(x_k) = U_q(x_k, \bar{v}_q^j(x_k), \bar{v}_q^j(x_k)) + V_q^j(F(x_k, \bar{v}_q^j(x_k))). \]  

Then, two iterations in GPI algorithm can be expressed as follows:

1) Policy improvement

\[ \bar{v}_q^j(x_k) = \arg \min_{\bar{u}_q(k)} \left\{ U_q(x_k, \bar{u}_q(k), \bar{u}_q(k)) + V_q^{i-1}(x_{k+1}) \right\}, \]  

2) Policy evaluation

\[ V_q^{i+1}(x_k) = U_q(x_k, \bar{v}_q^j(x_k), \bar{v}_q^j(x_k)) + V_q^{i}(F(x_k, \bar{v}_q^j(x_k))), \]

where

\[ V_q^{i+1}(x_k) = \min_{\bar{u}_q(k)} \left\{ U_q(x_k, \bar{u}_q(k), \bar{u}_q(k)) + V_q^{i-1}(x_{k+1}) \right\} \]

\[ = U_q(x_k, \bar{v}_q^j(x_k), \bar{v}_q^j(x_k)) + V_q^{i}(F(x_k, \bar{v}_q^j(x_k))), \]

\[ V_q^i(x_k) = V_q^{N_i}(x_k), \]  

which is known as an arbitrary non-negative integer. In summary, the total process of the GPI algorithm can be presented in Figure 1.

It should be noted that they, PI and VI algorithms, are the special forms of GPI algorithm. When the iteration sequence \( \{N_1, N_2, \ldots, N_i\} \) is all settled as 0, then the algorithm can be deemed as VI algorithm. Similarly, if \( N_i \to \infty \), the algorithm can be regarded as PI algorithm [11].

3.2 | Stability analysis

In this subsection, there will be a proof that the iterative value function of each player \( V_q^{ij}(x_k) \) converges to the optimal performance index function \( J_q(x_k, \bar{u}_1, \bar{u}_2, \ldots, \bar{u}_N) \) as \( i \to \infty \) for \( x_k \in \Omega_v \).

**Theorem 1** Let the initial control policy of each player be an admissible one which satisfies (12). Let the iterative control policy \( \bar{v}_q^j(x_k) \) and the iterative value function \( V_q^{ij+1}(x_k) \) be obtained by Equations (12)–(17) for \( i = 1, 2, \ldots \) and all \( x_k \in \Omega_v \). Then, these properties can be established.

1. For \( i = 1, 2, \ldots, j_i = 0, 1, \ldots, N_i \) and all \( x_k \in \Omega_v \)
\begin{align*}
V_{q}^{j+1}(x_k) &= U_q(x_k, \bar{v}_{q}^{j}(x_k)) + V_{q}^{j}(F(x_k, \bar{v}_{q}^{j}(x_k))) \\
&= \min_{\bar{u}_{q}(k)} \left\{ U_q(x_k, \bar{u}_{q}(k), \bar{u}_{q}(k)) + V_{q}^{j}(x_{k+1}) \right\} \\
&\leq U_q(x_k, \bar{v}_{q}^{j}(x_k), \bar{v}_{q}^{j}(x_k)) + V_{q}^{j}(F(x_k, \bar{v}_{q}^{j}(x_k))) \\
&= V_{q}^{j}(x_k). (20)
\end{align*}

Then, for \( j_1 = 0 \), there is

\begin{align*}
V_{q}^{1}(x_k) &= U_q(x_k, \bar{v}_{q}^{0}(x_k), \bar{v}_{q}^{0}(x_k)) + V_{q}^{0}(F(x_k, \bar{v}_{q}^{0}(x_k))) \\
&= U_q(x_k, \bar{v}_{q}^{0}(x_k), \bar{v}_{q}^{0}(x_k)) + V_{q}^{0}(F(x_k, \bar{v}_{q}^{0}(x_k))) \\
&= V_{q}^{0}(x_k). (21)
\end{align*}

Assume that Equation (18) holds for \( j_1 = l_1 - 1, l_1 = 1, 2, \ldots, N_1 \). Then, for \( j_1 = l_1 \) and for all \( x_k \in \Omega_\alpha \),

\begin{align*}
V_{q}^{1,l_1+1}(x_k) &= U_q(x_k, \bar{v}_{q}^{1,l_1}(x_k), \bar{v}_{q}^{1,l_1}(x_k)) \\
&+ V_{q}^{1,l_1}(F(x_k, \bar{v}_{q}^{1,l_1}(x_k))) \\
&\leq U_q(x_k, \bar{v}_{q}^{1,l_1}(x_k), \bar{v}_{q}^{1,l_1}(x_k)) + V_{q}^{1,l_1}(F(x_k, \bar{v}_{q}^{1,l_1}(x_k))) \\
&= V_{q}^{1,l_1}(x_k). (22)
\end{align*}

Therefore, Equation (18) holds for \( i = 1 \) and \( j_i = 0, 1, \ldots, N_i \).

Assume that Equation (18) holds for \( i = r, r = 1, 2, \ldots, \) that is

\begin{align*}
V_{q}^{r,j_i+1}(x_k) &\leq V_{q}^{r,j_i}(x_k). (23)
\end{align*}

Then, for \( i = r + 1 \) and all \( x_k \in \Omega_\alpha \), according to Equation (14), the iterative control policy \( \bar{v}_{q}^{r+1}(x_k) \) can be expressed as

\begin{align*}
\bar{v}_{q}^{r+1}(x_k) &= \arg \min_{\bar{u}_{q}(k)} \left\{ U_q(x_k, \bar{u}_{q}(k), \bar{u}_{q}(k)) + V_{q}^{r}(x_{k+1}) \right\}. (24)
\end{align*}

According to Equations (16) and (17), for all \( x_k \in \Omega_\alpha \), there is

\begin{align*}
V_{q}^{r+1,0}(x_k) &= U_q(x_k, \bar{v}_{q}^{r+1}(x_k), \bar{v}_{q}^{r+1}(x_k)) \\
&+ V_{q}^{r}(F(x_k, \bar{v}_{q}^{r+1}(x_k))) \\
&= \min_{\bar{u}_{q}(k)} \left\{ U_q(x_k, \bar{u}_{q}(k), \bar{u}_{q}(k)) + V_{q}^{r}(x_{k+1}) \right\} \\
&\leq U_q(x_k, \bar{v}_{q}^{r+1}(x_k), \bar{v}_{q}^{r+1}(x_k)) + V_{q}^{r}(F(x_k, \bar{v}_{q}^{r+1}(x_k))) \\
&\leq V_{q}^{r,N}(x_k) = V_{q}^{r}(x_k). (25)
\end{align*}
Then, for \( j_{i+1} = 0 \), there is

\[
V_{q}^{r+1}(x_k) = U_q(x_k, \bar{v}_q^{r+1}(x_k), \bar{\nu}_q^{r+1}(x_k)) + V_{q}^{r+1,0}(F(x_k, \bar{v}_q^{r+1}(x_k))) \\
\leq U_q(x_k, \bar{v}_q^{r+1}(x_k), \bar{\nu}_q^{r+1}(x_k)) + V_{q}^{r+1,0}(F(x_k, \bar{v}_q^{r+1}(x_k))) \\
= V_{q}^{r+1,0}(x_k).
\]

(26)

Therefore, Equation (18) holds for \( i = r + 1 \) and \( j_{i+1} = 0, 1, \ldots, N_{r+1} \). Above all, Equation (18) holds for \( i = 1, 2, \ldots \) and \( j_i = 0, 1, \ldots, N_i \).

Part 2: Equation (19) will be proved with mathematical induction as follows.

Firstly, for \( i = 1 \), let \( 0 \leq j_1 \leq N_1 \) and \( 0 \leq j_{i+1} \leq N_{r+1} \). According to Equations (20)–(27), for all \( x_k \in \Omega_x \),

\[
V_{q}^{r+1}(x_k) = U_q(x_k, \bar{v}_q^{r+1}(x_k), \bar{\nu}_q^{r+1}(x_k)) + V_{q}^{r+1,0}(F(x_k, \bar{v}_q^{r+1}(x_k))) \\
\leq U_q(x_k, \bar{v}_q^{r+1}(x_k), \bar{\nu}_q^{r+1}(x_k)) + V_{q}^{r+1,0}(F(x_k, \bar{v}_q^{r+1}(x_k))) \\
= V_{q}^{r+1,0}(x_k).
\]

Therefore, Equation (19) can be deduced for \( i = 1, 2, \ldots \). The proof is completed.

**Lemma 1** Let the iterative value function of the qth player \( V_{q}^{i}(x_k) \) be defined as Equation (17). Then, for \( i = 1, 2, \ldots \), the iterative value function of the qth player \( V_{q}^{i}(x_k) \) is a monotonically non-increasing and convergent sequence.

**Lemma 2** If a monotonically non-increasing sequence \( \{a_n\} \), \( n = 1, 2, \ldots \), contains an arbitrary convergent subsequence, the sequence \( \{a_n\} \) is convergent to the same limit of its subsequence.

The proof of Lemmas 1 and 2 is omitted, which is easily obtained by [11]. Then, the convergence of the iterative value function \( V_{q}^{i}(x_k) \) to the optimal performance index function \( f_q(x_k, \bar{u}_1, \bar{u}_2, \ldots, \bar{u}_N) \) is provided as below by using above lemmas.

**Theorem 2** Let the iterative value function \( V_{q}^{i}(x_k) \) and the iterative control policy \( \bar{v}_q^{i}(x_k) \) be obtained using Equations (12)–(17) for \( i = 1, 2, \ldots, j_i = 0, 1, \ldots, N_i \) and all \( x_k \in \Omega_x \). The iterative value function \( V_{q}^{i}(x_k) \) converges to the optimal performance index function \( f_q(x_k, \bar{u}_1, \bar{u}_2, \ldots, \bar{u}_N) \) as \( i \to \infty \), that is

\[
\lim_{i \to \infty} V_{q}^{i}(x_k) = f_q(x_k, \bar{u}_1, \bar{u}_2, \ldots, \bar{u}_N).
\]

(29)

**Proof.** Define a sequence of the iterative value functions as

\[
\left\{ V_{q}^{i}(x_k) \right\} = \left\{ V_{q}^{1,0}(x_k), V_{q}^{1,1}(x_k), \ldots, V_{q}^{N_1}(x_k), V_{q}^{1}(x_k), V_{q}^{2}(x_k), \ldots, V_{q}^{N_2}(x_k), V_{q}^{2}(x_k), \ldots \right\}.
\]

(30)

Meanwhile, another sequence can be defined as

\[
\left\{ V_{q}^{i}(x_k) \right\} = \left\{ V_{q}^{j}(x_k), V_{q}^{j+1}(x_k), \ldots \right\}.
\]

(31)

Then, it is obviously that \( \{ V_{q}^{i}(x_k) \} \), a monotonically non-increasing sequence, is a subsequence of \( \{ V_{q}^{i}(x_k) \} \). According to Lemma 1, the limit of \( \{ V_{q}^{i}(x_k) \} \) exists. From Lemma 2, the sequence \( \{ V_{q}^{i}(x_k) \} \) and its subsequence \( \{ V_{q}^{j}(x_k) \} \) process the same limit, that is,

\[
\lim_{i \to \infty} V_{q}^{i}(x_k) = \lim_{i \to \infty} V_{q}^{j}(x_k).
\]

(32)

Therefore, the proposition stated in Theorem 2 turns into the following statement:

\[
\lim_{i \to \infty} V_{q}^{i}(x_k) = f_q(x_k, \bar{u}_1, \bar{u}_2, \ldots, \bar{u}_N).
\]

(33)

According to Lemma 1, the iterative value function of the qth player \( V_{q}^{\infty}(x_k) \) can be defined as the limit of the iterative value function \( V_{q}^{i}(x_k) \), that is,

\[
V_{q}^{\infty}(x_k) = \lim_{i \to \infty} V_{q}^{i}(x_k).
\]

(34)

According to Equations (17) and (18), there is
\[ V_q^i(x_k) \leq V_q^{i,0}(x_k) \]
\[ = U_q(x_k, \bar{e}_q(x_k), \bar{e}_q^2(x_k)) + V_q^{i-1}(F(x_k, \bar{e}_q^2(x_k))) \]
\[ = \min_{\bar{u}_q(k)} \left\{ U_q(x_k, \bar{u}_q(k), \bar{u}_q^2(k)) + V_q^{i-1}(x_{k+1}) \right\}. \]

(35)

Then, according to Equation (34), there is

\[ V_q^\infty(x_k) = \lim_{i \to \infty} V_q^i(x_k) \leq V_q^\infty(x_k) \]
\[ \leq \min_{\bar{u}_q(k)} \left\{ U_q(x_k, \bar{u}_q(k), \bar{u}_q^2(k)) + V_q^{\infty}(x_{k+1}) \right\}. \]

(36)

When \( i \to \infty \), for all \( x_k \in \Omega \), there is

\[ V_q^\infty(x_k) \leq \min_{\bar{u}_q(k)} \left\{ U_q(x_k, \bar{u}_q(k), \bar{u}_q^2(k)) + V_q^{\infty}(x_{k+1}) \right\}. \]

(37)

Let \( \varepsilon_p > 0 \) be an arbitrary positive number. Since \( V_q^i(x_k) \) is monotonically non-increasing for \( i > 0 \) and \( V_q^\infty(x_k) = \lim_{i \to \infty} V_q^i(x_k) \), there exists a positive integer \( p \) which satisfies

\[ V_q^p(x_k) - \varepsilon_p \geq V_q^\infty(x_k) \geq V_q^p(x_k). \]

(38)

Hence, according to Equation (15), there is

\[ V_q^\infty(x_k) \geq U_q(x_k, \bar{e}_q^p(x_k), \bar{e}_q^2(x_k)) \]
\[ + V_q^p(F(x_k, \bar{e}_q^p(x_k))) - \varepsilon_p \]
\[ \geq U_q(x_k, \bar{e}_q^p(x_k), \bar{e}_q^2(x_k)) \]
\[ + V_q^\infty(F(x_k, \bar{e}_q^p(x_k))) - \varepsilon_p \]
\[ \geq \min_{\bar{u}_q(k)} \left\{ U_q(x_k, \bar{u}_q(k), \bar{u}_q^2(k)) + V_q^\infty(x_{k+1}) \right\} - \varepsilon_p. \]

(39)

Since \( \varepsilon_p > 0 \) and \( \varepsilon_p \) is arbitrary, for all \( x_k \in \Omega \), there is

\[ V_q^\infty(x_k) \geq \min_{\bar{u}_q(k)} \left\{ U_q(x_k, \bar{u}_q(k), \bar{u}_q^2(k)) + V_q^{\infty}(x_{k+1}) \right\}. \]

(40)

According to Equations (37) and (40), for all \( x_k \in \Omega \), \( V_q^\infty(x_k) \) can be obtained, that is,

\[ V_q^\infty(x_k) = \min_{\bar{u}_q(k)} \left\{ U_q(x_k, \bar{u}_q(k), \bar{u}_q^2(k)) + V_q^{\infty}(x_{k+1}) \right\}. \]

(41)

Let \( \varepsilon_r > 0 \) be an arbitrary positive number. According to Equation (2), there exists a sequence of the iterative control policies \( \bar{\gamma} = \{ \bar{\gamma}_1, \bar{\gamma}_2, \ldots, \bar{\gamma}_N \} \) which satisfies

\[ J_q(x_0, \bar{\gamma}_1, \bar{\gamma}_2, \ldots, \bar{\gamma}_N) \leq J_q^\prime(x_0, \bar{u}_1, \bar{u}_2, \ldots, \bar{u}_N) + \varepsilon_r. \]

(42)

Assume that the iteration index \( r \) related to the sequence \( \bar{\gamma}_r \). Then, according to Equation (2) and Theorem 1, there is

\[ V_q^\infty(x_k) \leq V_q^\infty(x_k) \leq \min_{\bar{u}_q(k)} \left\{ U_q(x_k, \bar{u}_q(k), \bar{u}_q^2(k)) + V_q^{\infty}(x_{k+1}) \right\} \]
\[ \leq J_q(x_0, \bar{\gamma}_1, \bar{\gamma}_2, \ldots, \bar{\gamma}_N). \]

(43)

Combined with Equation (42), there is

\[ V_q^\infty(x_k) \leq \min_{\bar{u}_q(k)} \left\{ U_q(x_k, \bar{u}_q(k), \bar{u}_q^2(k)) + V_q^{\infty}(x_{k+1}) \right\} \]
\[ \leq J_q(x_0, \bar{u}_1, \bar{u}_2, \ldots, \bar{u}_N) + \varepsilon_r. \]

(44)

Since \( \varepsilon_r > 0 \) and \( \varepsilon_r \) is arbitrary, there is

\[ V_q^\infty(x_k) \leq J_q(x_0, \bar{u}_1, \bar{u}_2, \ldots, \bar{u}_N). \]

(45)

Next, according to Equation (5), for \( i = 1, 2, \ldots, V_q^\infty(x_k) \geq J_q(x_k, \bar{u}_1, \bar{u}_2, \ldots, \bar{u}_N). \) Let \( i \to \infty \), there is

\[ V_q^\infty(x_k) \geq J_q(x_k, \bar{u}_1, \bar{u}_2, \ldots, \bar{u}_N). \]

(46)

Therefore, according to Equations (45) and (46), there is

\[ V_q^\infty(x_k) = \lim_{i \to \infty} V_q^i(x_k) = J_q(x_k, \bar{u}_1, \bar{u}_2, \ldots, \bar{u}_N). \]

(47)

The proof is completed.

### 3.3 Neural networks approximation

In this subsection, a neural network-based approach is presented to approximate the iterative value function and iterative control policy of each player respectively, which are illustrated as

\[ \hat{F}(x_k, W_q) = W_q^T\phi_q(x_k) \]

(48)

where \( W_q \) denotes the weight matrix of the \( q \)th player, \( \phi_q(\cdot) \) denotes the activation function of the \( q \)th player.

Firstly, the critic network is constructed to approximate the iterative value functions of the \( q \)th player \( \hat{V}_q^{i,j,l}(x_k) \) which is denoted as

\[ \hat{V}_q^{i,j,l}(x_k) = W_q^{\phi(i,j,l)}\phi_q(x_k), \]

(49)

where \( l = 0, 1, 2, \ldots \) is denoted as the iteration index of the following gradient descent iteration method. Recall that \( W_q^{\phi(i,j,l)} \) is the critic network weights and \( \phi_q(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^l \) is the
activation function for the critic neural network. Define the approximation error of the critic network for the $q$th player as

$$
e_q^{(i,j,l)} = V_q^{(i,j,l)}(x_k) - V_q^{i,j}(x_k). \quad (50)$$

Then, define the objective function to be minimized in the critic network as

$$E_q^{(i,j,l)} = \frac{1}{2}\left( e_q^{(i,j,l)} \right)^2. \quad (51)$$

Hence, the gradient-based weight update law [24,25] of the critic network is derived as

$$\begin{align*}
\tilde{W}_q^{a(i,l)} &= \tilde{W}_q^{a(i,l)} - \alpha_q \frac{\partial E_q^{(i,j,l)}}{\partial W_q^{a(i,l)}} \\
&= \tilde{W}_q^{a(i,l)} - \alpha_q \frac{\partial E_q^{(i,j,l)}}{\partial W_q^{a(i,l)}} \tilde{V}_q^{i,j}(x_k) \\
&= \tilde{W}_q^{a(i,l)} - \alpha_q \frac{\partial E_q^{(i,j,l)}}{\partial W_q^{a(i,l)}} \tilde{V}_q^{i,j}(x_k)
\end{align*} \quad (52)$$

where $\alpha_q > 0$ is the learning rate of the critic network for the $q$th player. Next, the actor network for the $q$th player will be introduced as follows.

The actor network is constructed to approximate the iterative control policy of the $q$th player $\hat{\varphi}_q^{s} (x_k)$, which is denoted as

$$\hat{\varphi}_q^{s} (x_k) = W_q^{a(i)} \phi_q^{s}(x_k), \quad (53)$$

where $l = 0, 1, 2, \ldots$ is denoted as the iteration index of the following gradient descent iteration method. Recall that $W_q^{a(i)}$ is the actor network weights and $\phi_q^{s} \cdot : \mathbb{R}^n \to \mathbb{R}^s$ is the activation function for the actor neural network. $s$ is the number of neurons in the hidden layer. Define the approximation error of the actor network for the $q$th player as

$$e_q^{a(i)} = \hat{\varphi}_q^{s}(x_k) - \phi_q^{s}(x_k). \quad (54)$$

Then, define the objective function to be minimized in the actor network as

$$E_q^{(i,l)} = \frac{1}{2}\left( e_q^{(i,l)} \right)^2. \quad (55)$$

Hence, the gradient-based weight update law [24,25] of the actor network is derived as

$$\begin{align*}
\tilde{W}_q^{a(i,l+1)} &= \tilde{W}_q^{a(i,l)} - \beta^a_q \frac{\partial E_q^{(i,l)}}{\partial W_q^{a(i,l)}} \\
&= \tilde{W}_q^{a(i,l)} - \beta^a_q \frac{\partial E_q^{(i,l)}}{\partial W_q^{a(i,l)}} \phi_q^{s}(x_k) \\
&= \tilde{W}_q^{a(i,l)} - \beta^a_q \phi_q^{s}(x_k) \left( e_q^{a(i)} \right)^T
\end{align*} \quad (56)$$

where $\beta^a_q > 0$ is the learning rate of the actor network for the $q$th player.

## 4 | SIMULATION RESULTS

In this section, there are two simulation examples to show the performance of the proposed scheme, which solves discrete-time control-constraint multi-player NZS games.

### 4.1 Example 1

Consider the following control-constraint two-player linear NZS game based on [26,27] of the form

$$x_{k+1} = f(x_k) + g_1(x_k)u_1(k) + g_2(x_k)u_2(k), \quad (57)$$

where

$$f(x_k) = \begin{bmatrix}
-0.0200 & 0.0193 & 0.4444 & 0.0002 \\
0.0703 & 0.9340 & 0.0000 & -0.4011 \\
0.0000 & 0.0001 & 1.0000 & 0.0191 \\
-0.0781 & 0.0057 & 0.0000 & 0.9091 \\
0.04 & 0.0331 & 0.3331 & 0.0053 \\
0.0003 & 0.0341 & 0.5413 & 0.0051
\end{bmatrix},$$

$$g_1(x_k) = \begin{bmatrix}
0.1169 \\
0.0055 \\
0.5413
\end{bmatrix},$$

The performance index function is defined as Equation (2) where the weight matrices in the utility function for two players are expressed as

$$Q_1(x_k) = \begin{bmatrix}
0.1 & 0 & 0 & 0 \\
0 & 0.01 & 0 & 0 \\
0 & 0 & 0.1 & 0 \\
0 & 0 & 0 & 0.01
\end{bmatrix}, M_{11} = M_{12} = 1,$$

$$Q_2(x_k) = \begin{bmatrix}
0.1 & 0 & 0 & 0 \\
0 & 0.1 & 0 & 0 \\
0 & 0 & 0.1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}, M_{21} = 1, M_{22} = 10^{-5}.$$

Firstly, define the state space as $\bar{\Omega}_k = \{x_k| -0.5 \leq x_{1k} \leq 0.5, -0.5 \leq x_{2k} \leq 0.5\}$. Besides, states are chosen in $\bar{\Omega}_k$ at 0.1 intervals to apply the GPI algorithm for the acquirement of the optimal control policies. Four neural networks are utilized to approximate the iterative value functions and control policies, whose activation functions are

$$\phi_1^s(x_k) = \phi_2^s(x_k) \in \{x_{1k}, x_{2k}, x_{3k}, x_{4k}\}^T$$

in the actor networks and
Consider the following control-constraint two-player nonlinear NCS game based on [9,28] of the form
\[ x_{k+1} = f(x_k) + g_1(x_k)u_1(k) + g_2(x_k)u_2(k), \]  
(58)

where
\[ f(x_k) = \begin{bmatrix} 0.99x_{1k} + 0.01x_{2k} + 0.02x_{1k}^3 \\ -0.005x_{1k} + 0.995x_{2k} + 0.005x_{2k}(1+2x_{2k}^2)\sin^2(x_{1k}) \end{bmatrix}, \]
\[ g_1(x_k) = \begin{bmatrix} 0.01\sin(x_{1k}) \\ 0 \end{bmatrix}, \quad g_2(x_k) = \begin{bmatrix} 0 \\ 0.01\cos(x_{2k}) \end{bmatrix}. \]

The performance index function is defined in Equation (2) where the weight matrices in the utility function for two players are expressed as
\[ Q_1(x_k) = 0.01x_{1k}^2 + 0.01x_{2k}^2 + 0.02x_{1k}^4, \]
\[ Q_2(x_k) = 0.02x_{1k}^2 + 0.02x_{2k}^2 + 0.01x_{2k}^4, \]
\[ M_{11} = M_{12} = M_{21} = M_{22} = 0.01. \]

in the critic networks respectively. In addition, the initial control policies chosen for two players should obtain admissible control inputs. The iteration sequence \( \{N_1, N_2, ..., N_l\} \) for each player can be randomly chosen. Let the initial state be \( x_0 = [0.5,0.5,0.5,0.5]^T \), the iteration index \( i = 50 \) and actuator saturation \( |v_1| \leq 0.2, |v_2| \leq 0.2 \). Then, train the actor and critic networks of two players under the learning rate 0.65, 0.05, 0.8, 0.75, respectively. The trajectories of the system states and control inputs are shown in Figure 2. Trajectories of the performance index function for system and each player under initial control policies and obtained control policies are presented in Figure 3. The trajectories of the actor and critic network weights for two players are shown in Figure 4, where the different colored lines represent each term of the actor and critic network weights.

From the above results, it is obvious to see that the system states converge to equilibrium after around 550-time steps. Besides, it can be observed that the obtained control inputs of two players achieve convergence under control constraints, as is shown in Figure 2. Compared with the initial control policies, the system and each player performance index functions obtained under the obtained control policies are significantly reduced, as is shown in Figure 3. Meanwhile, the actor and critic network weights approach to equilibrium after about 30 iteration steps.

### Example 2

Consider the following control-constraint two-player nonlinear NCS game based on [9,28] of the form
\[ x_{k+1} = f(x_k) + c_1(x_k)c_1(k) + c_2(x_k)c_2(k), \]  
(58)

where
\[ f(x_k) = \begin{bmatrix} 0.99x_{1k} + 0.01x_{2k} + 0.02x_{1k}^3 \\ -0.005x_{1k} + 0.995x_{2k} + 0.005x_{2k}(1+2x_{2k}^2)\sin^2(x_{1k}) \end{bmatrix}, \]
\[ c_1(x_k) = \begin{bmatrix} 0.01\sin(x_{1k}) \\ 0 \end{bmatrix}, \quad c_2(x_k) = \begin{bmatrix} 0 \\ 0.01\cos(x_{2k}) \end{bmatrix}. \]

The performance index function is defined in Equation (2) where the weight matrices in the utility function for two players are expressed as
\[ Q_1(x_k) = 0.01x_{1k}^2 + 0.01x_{2k}^2 + 0.02x_{1k}^4, \]
\[ Q_2(x_k) = 0.02x_{1k}^2 + 0.02x_{2k}^2 + 0.01x_{2k}^4, \]
\[ M_{11} = M_{12} = M_{21} = M_{22} = 0.01. \]

in the actor networks and
in the critic networks, respectively. In addition, the initial control policies chosen for two players should obtain admissible control inputs. The iteration sequence \( \{N_1, N_2, ..., N_i\} \) for each player can be randomly chosen. Let the initial state be \( x_0 = [0.5, 0.5]^T \), the iteration index \( i = 150 \) and actuator saturation \( |\nu_1| \leq 0.3, |\nu_2| \leq 0.15 \). Then, train the actor and critic networks of two players under the learning rate 0.05, 0.05, 0.15, 0.2, respectively. The trajectories of the system states and the control inputs are shown in Figure 5. Trajectories of the performance index function for system and each player under initial control policies and obtained control policies are presented in Figure 6. The trajectories of the actor and critic network weights for two players are shown in Figure 7, where the different coloured lines represent each term of the actor and critic network weights. As shown in Figure 8, the proposed scheme by using GPI algorithm, PI algorithm or VI algorithm is tested for 100 times respectively in the same condition, which the parameter \( N_i \) is settled as 0 for PI algorithm and 50 for VI algorithm, respectively. Besides, the error of each test is presented as a shaded part related to the colour of the algorithm used. Meanwhile, the runtime results of 10 times tests are presented in Figure 9. Finally, some concrete numerical analysis of runtime results and system performance index function results are given in Tables 1 and 2.

From the above results, it is clear to see that the system states converge to equilibrium after around 700 time steps. Besides, it can be observed that obtained control inputs of two players achieve convergence under control constraints, as shown in Figure 5. Compared with the initial control policies, the system and each player performance index functions obtained under the obtained control policies are significantly reduced, as shown in Figure 6. Meanwhile, the actor and critic network weights approach to equilibrium after about 100 iteration steps. What’s more, it can be obviously seen from Figure 8 that, on the whole, GPI algorithm is similar with the traditional PI and VI algorithms in the results of the system performance index function. When comparing carefully, it is obvious that slightly less energy loss is required in GPI algorithm whether it is the mean value or the minimum value. Although the GPI algorithm is slightly better than the PI algorithm in the results of the system performance index function, it will be much better in runtime results. As shown in Figure 9, in 100 times of tests, runtime results constructed by the GPI algorithm are more intensive than the ones in the

**Figure 4** Trajectories of the actor network weights and the critic network weights. (a) First player, (b) second player, (c) first player, (d) second player.

**Figure 5** Trajectories of the system states and the control inputs. (a) System states, (b) control inputs.

**Figure 6** Trajectories of the performance index function under initial control policies and obtained control policies. (a) System, (b) first player, (c) second player.
traditional PI and VI algorithms. Besides, the runtime results constructed by the GPI algorithm are significantly less than the ones in the traditional PI and VI algorithms whether it is the mean, the minimum or the maximum value. From tables 1 and 2, the runtime is greatly reduced by more than 25% in GPI algorithm compared with the PI algorithm whose results of system performance index function is suboptimal. Additionally, the standard deviation of the runtime results constructed by the GPI algorithm is smaller than the one used the PI algorithm up to around 40%, and the VI algorithm up to around 60%, which reflects the stability and reliability of the GPI algorithm are improved up to 40% compared with the PI algorithm. Therefore, the GPI algorithm holds higher effectiveness compared with PI and VI algorithms for discrete-time control-constraint multi-player NZS games.

It should be noted that, for the sake of clarity, a mark point is made every 20 time intervals in Figures 2 and 5. Besides, a mark point is also made every five iteration intervals in Figures 4 and 7. In addition, the index ‘Mean’ means the average of the results. The index ‘Std’ means the standard deviation of the results. The index ‘Trimmean’ means the average of the results removing ten maximum and ten minimum in Tables 1 and 2.

**CONCLUSION**

In this article, the discrete-time control-constraint multi-player NZS games based on generalized policy iteration ADP approach is presented. Besides, the stability analysis of the proposed scheme is illustrated in detail. Meanwhile, a neural network-based implement is developed to approximate the iterative control policy and value function. Finally, the scheme we discussed is employed to show the performance in the linear and non-linear simulation examples, respectively.
REFERENCES

1. Lewis, F.L., Vrabie, D.: Reinforcement learning and adaptive dynamic programming for feedback control. IEEE Circ. Syst. Mag. 9(3), 32–50 (2009)

2. Modares, H., Lewis, F.L., Naghibi, S.M.B.: Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems. Automatica. 50(1), 193–202 (2014)

3. Bhattacharjee, S., et al.: Natural actor-critic algorithms. Automatica. 45(11), 2471–2482 (2009)

4. Bai, W., et al.: Adaptive reinforcement learning neural network control for uncertain nonlinear system with input saturation. IEEE Trans. Cyber. 50(8), 3433–3443 (2019)

5. Werbos, P.: Stable adaptive control using new critic designs. Proc. SPIE - Int. Soc. Opt. Eng. 3728 (1998)

6. Jiang, Y., Jiang, Z.: Global adaptive dynamic programming for continuous-time nonlinear systems. IEEE Trans. Auto. Control. 60(11), 2917–2929 (2015)

7. Yang, X., et al.: Adaptive dynamic programming for robust neural control of unknown continuous-time non-linear systems. IET Contr. Theor. App. 11(14), 2307–2316 (2017)

8. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press (1998)

9. Lewis, F.L., Vrabie, D., Vamvoudakis, K.G.: Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. IEEE Contr. Syst. Mag. 32(6), 76–105 (2012)

10. Zhang, H., et al.: Discrete-time nonzero-sum games for multiplayer using policy-iteration-based adaptive dynamic programming algorithms. IEEE Trans. Cyber. 47(10), 3331–3340 (2017)

11. Liu, D., Wei, Q., Yan, P.: Generalized policy iteration adaptive dynamic programming for discrete-time nonlinear systems. IEEE Trans. Syst. Man Cyber.: Syst. 45(12), 1577–1591 (2015)

12. Vamvoudakis, K.G., Lewis, F.L., Hudas, G.R.: Multi-agent differential graphical games: online adaptive learning solution for synchronization with optimality. Automatica. 48(8), 1598–1611 (2012)

13. Mu, C., Zhao, Q., Sun, C.: Optimal model-free output synchronization of heterogeneous multi-agent systems under switching topologies. IEEE Trans. Ind. Electr. 67(12), 10951–10964 (2019). https://doi.org/10.1109/TIE.2019.2958277

14. Mu, C., et al.: Cooperative differential game-based optimal control and its application to power systems. IEEE Trans. Ind. Inf. (16)(8), 5169–5179 (2019). https://doi.org/10.1109/TII.2019.2955966. in press

15. Wei, Q., Song, R., Yan, P.: 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), 444–458 (2016)

16. Vrabie, D., Lewis, F.: Integral reinforcement learning for online computation of feedback Nash strategies of nonzero-sum differential games. In: 49th IEEE Conference on Decision and Control (CDC), pp. 3066–3071 (2010)

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

18. Zhang, H., Cui, L., Luo, Y.: Near-optimal control for nonzero-sum differential games of continuous-time nonlinear systems using single-network ADP. IEEE Trans. Cyber. 43(1), 206–216 (2013)

19. Song, R., Lewis, F.L., Wei, Q.: Off-policy integral reinforcement learning method to solve nonlinear continuous-time multiplayer nonzero-sum games. IEEE Trans. Neural Netw. Learn. Syst. 28(3), 704–713 (2017)

20. Zhao, D., et al.: Experience replay for optimal control of nonzero-sum game systems with unknown dynamics. IEEE Trans. Cyber. 46(3), 854–865 (2016)

21. Abu-Khalaf, M., Lewis, F.L.: Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach. Automatica. 41(5), 779–791 (2005)

22. Mu, C., Wang, K., Sun, C.: Policy-iteration-based learning for nonlinear player game systems with constrained inputs. IEEE Trans. Syst. Man Cyber.: Syst. (2019). https://doi.org/10.1109/TSMC.2019.2962629. in press

23. Modares, H., Lewis, F.L., Naghibi-Sistani, M.: Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks. IEEE Trans. Neural Netw. Learn. Syst. 24(10), 1513–1525 (2013)

24. Al-Tamimi, A., Lewis, F.L., Abu-Khalaf, M.: Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof. IEEE Trans. Syst. Man Cyber. Part B. 38(4), 943–949 (2008)

25. Chen, J., et al.: Gradient-based particle filter algorithm for an ARX model with nonlinear communication output. IEEE Trans. Syst. Man Cyber.: Syst. 50(6), 2198–2207 (2018). https://doi.org/10.1109/TSMC.2018.2810277

26. Freiling, G., Jank, G., Abou-Kandil, H.: On global existence of solutions to coupled matrix riccati equations in closed-loop Nash games. IEEE Trans. Auto. Contr. 41(2), 264–269 (1996)

27. Schmitendorf, W.E.: Designing stabilizing controllers for uncertain systems using the Riccati equation approach. IEEE Trans. Auto. Contr. 33(4), 376–379 (1988)

28. Si, J., Wang, Y-T: Online learning control by association and reinforcement. IEEE Trans. Neural Netw. 12(2), 264–276 (2001)

How to cite this article: Mu C, Peng J, Tang Y. Learning-based control for discrete-time constrained nonzero-sum games. CAAI Trans. Intell. Technol. 2021;6:203–213. https://doi.org/10.1049/cit2.12015