Data-based Optimal Control for Discrete-time Zero-sum Games of 2-D Systems Using Adaptive Critic Designs

WEI Qing-Lai[1] ZHANG Hua-Guang[2] CUI Li-Li[2]

Abstract In this paper, an iterative adaptive critic design (ACD) algorithm is proposed to solve a class of discrete-time two-person zero-sum games for Roesser type 2-D system. The idea is to use adaptive critic technique to obtain the optimal control pair iteratively to make the performance index function reach the saddle point of the zero-sum games. The proposed iterative ACD algorithm can be implemented based on the input and state data without the system model. Stability analysis of the 2-D system is presented and the convergence property of the performance index function is also proved. Neural networks are used to approximate the performance index function and compute the optimal control policies, respectively, for facilitating the implementation of the iterative ACD algorithm. The optimal control scheme of the air drying process is given to illustrate the performance of the proposed method.

Key words Adaptive critic designs (ACD), optimal control, zero-sum game, 2-D system, neural networks

A large class of complicated practical systems are 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 performance index function as a game[1]. Zero-sum game theory has been widely applied to decision making and control engineering problems[2−5]. In these situations, many control schemes are presented in order to reach some form of optimality[6−7]. In [8], zero-sum game was proposed to solve multiuser optimal flow control. In [9], the zero-sum game problem was discussed for noncooperative decision makers. Based on the zero-sum theory, the designs of controller in the worst case and the design of H_∞ controller were proposed in $[10-12]$. However, aforementioned results on zero-sum game are only for the one-dimensional systems. In the real world, many complicated control systems are described by 2-dimensional (2-D) structures[13−14]. The key feature of a 2-D system is that the information is propagated along two independent directions. Many physical processes, such as thermal processes, image processing, signal filtering, etc., have a clear 2-D structure. The 2-D system theory is frequently used as an analysis tool to solve some problems, e.g., iterative learning control[15] and repetitive process control[16]. So many control schemes are presented for 2-D system in order to obtain the optimal performance[17−18], while there are few results on the zero-sum games for 2-D systems. The great difficulty of the zero-sum games for 2-D systems is that the optimal recurrent equation, so called Hamilton-Jacobi-Isaacs (HJI) equation, is invalid in 2-D structure, which means that the optimal control pair cannot be obtained by the classical dynamic programming theory. Another difficulty lies in the fact that for many 2-D systems the model of the system cannot be obtained inherently. So it is important and necessary to give a new method to solve the zero-sum games for 2-D system without a system model. This motivates our research.

The adaptive critic designs (ACDs) are very useful tools in solving the optimal control problems and have received considerable attention for the past three decades[19−22]. ACDs were firstly proposed in $[23-25]$ as a way to solve optimal control problems forward-in-time. ACDs combine reinforcement learning technique and dynamic programming theory with neural networks. In [13], the ACDs were classified into four main schemes: heuristic dynamic programming (HDP), dual heuristic dynamic programming (DHP), action dependent heuristic dynamic programming (ADHDP), also known as Q-learning[23], and action dependent dual heuristic dynamic programming (ADDHP). In [26], another two ACD schemes known as globalized-DHP (GDHP) and ADGDHP were developed. Though in recent years, ACDs have been further studied by many researchers such as $[27-35]$, wherein most results focus on the optimal control problem with a single controller. Only in [36], based on HJI equation, zero-sum game was discussed for 1-D system. To the best of our knowledge, there are no results discussing how to solve the zero-sum game problem for 2-D systems.

In brief, it is the first time for the zero-sum game to solve for a 2-D system by ACD technique. The main contributions of this paper include:

1) Propose a new optimality principle for Roesser type 2-D system and obtain the optimal control formulation in theory.

2) Propose an iterative algorithm based on ACD technique (iterative ACD algorithm for brief) to obtain the optimal control pair iteratively with rigorous stability and convergence analysis.

3) Develop the iterative ACD algorithm into data-driven situation. What is needed to know is only the input and state data, and the model of the system is not required.

This paper is organized as follows. Section 1 presents the preliminaries and assumptions. In Section 2, the optimal control for zero-sum games for 2-D systems is proposed and the properties of the optimal control are also discussed. In Section 3, data-based iterative ACD algorithm is proposed with the convergence analysis. In Section 4, the neural network implementation for the control scheme is discussed. In Section 5, an example is given to demonstrate the effectiveness of the proposed control scheme. The conclusion is drawn in Section 6.

1 Preliminaries and assumptions

Basically, we consider the following discrete-time linear Roesser type 2-D system

$$\boldsymbol{x}^+(k,l) = A\boldsymbol{x}(k,l) + B\boldsymbol{u}(k,l) + C\boldsymbol{w}(k,l) \tag{1}$$

Received December 16, 2008; in revised form March 9, 2009
Supported by National High Technology Research and Development Program of China (863 Program) (2006AA04Z183), National Natural Science Foundation of China (60621001, 60534010, 60572070, 60774048, 60728307), Program for Changjiang Scholars and Innovative Research Groups of China (60728307, 4031002)
1. The Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, P. R. China 2. The School of Information Science and Engineering, Northeastern University, Shenyang 110004, P. R. China
DOI: 10.3724/SP.J.1004.2009.00682

$$\boldsymbol{x}^{\mathrm{h}}(0,l)=\boldsymbol{f}(l),\quad \boldsymbol{x}^{\mathrm{v}}(k,0)=\boldsymbol{g}(k) \tag{2}$$

with

$$\boldsymbol{x}(k,l)=\begin{bmatrix}\boldsymbol{x}^{\mathrm{h}}(k,l)\\ \boldsymbol{x}^{\mathrm{v}}(k,l)\end{bmatrix},\quad \boldsymbol{x}^{+}(k,l)=\begin{bmatrix}\boldsymbol{x}^{\mathrm{h}}(k+1,l)\\ \boldsymbol{x}^{\mathrm{v}}(k,l+1)\end{bmatrix} \tag{3}$$

where $\boldsymbol{x}^{\mathrm{h}}(k,l)$ is the horizon state in $\mathbf{R}^{n_1}$, $\boldsymbol{x}^{\mathrm{v}}(k,l)$ is the vertical state in $\mathbf{R}^{n_2}$, $\boldsymbol{u}(k,l)$ and $\boldsymbol{w}(k,l)$ are the control inputs in $\mathbf{R}^{m_1}$ and $\mathbf{R}^{m_2}$. Let the system matrices $A\in\mathbf{R}^{(n_1+n_2)\times(n_1+n_2)}$, $B\in\mathbf{R}^{(n_1+n_2)\times n_1}$, and $C\in\mathbf{R}^{(n_1+n_2)\times m_2}$. Assume all the system matrices are nonsingular and the system matrices can be expressed by

$$A=\begin{bmatrix}A_1\ A_2\\ A_3\ A_4\end{bmatrix},\ B=\begin{bmatrix}B_1\\ B_2\end{bmatrix},\ C=\begin{bmatrix}C_1\\ C_2\end{bmatrix} \tag{4}$$

The function $\boldsymbol{f}(l)$ and $\boldsymbol{g}(k)$ are corresponding boundary conditions along two independent indirections.

We define the following denotements

$$\begin{aligned}&(k,l)\le(m,n)\quad \text{if and only if } k\le m \text{ and } l\le n\\ &(k,l)=(m,n)\quad \text{if and only if } k=m \text{ and } l=n\\ &(k,l)<(m,n)\quad \text{if and only if } (k,l)\le(m,n) \text{ and}\\ &\qquad\qquad (k,l)\ne(m,n)\end{aligned} \tag{5}$$

Then, the infinite-time performance index function for 2-D systems can be given by

$$\begin{aligned}V(\boldsymbol{x}(0,0),\boldsymbol{u},\boldsymbol{w})=&\sum_{(0,0)\le(k,l)<(\infty,\infty)}\sum\Big(\boldsymbol{x}^{\mathrm{T}}(k,l)Q\boldsymbol{x}(k,l)+\\ &\boldsymbol{u}^{\mathrm{T}}(k,l)R\boldsymbol{u}(k,l)+\boldsymbol{w}^{\mathrm{T}}(k,l)S\boldsymbol{w}(k,l)\Big)\end{aligned} \tag{6}$$

where $Q\ge 0$, $R>0$, and $S<0$ are with suitable dimensions and $L(\boldsymbol{x}(k,l),\boldsymbol{u}(k,l))=\boldsymbol{x}^{\mathrm{T}}(k,l)Q\boldsymbol{x}(k,l)+\boldsymbol{u}^{\mathrm{T}}(k,l)R\boldsymbol{u}(k,l)+\boldsymbol{w}^{\mathrm{T}}(k,l)S\boldsymbol{w}(k,l)$ is the utility function. For the above zero-sum game, the two control variables $\boldsymbol{u}$ and $\boldsymbol{w}$ are chosen, respectively, by player I and player II where player I tries to minimize the performance index function $V(\boldsymbol{x})$, while player II attempts to maximize it. The following assumptions are proposed that are in effect in the remaining sections.

Assumption 1. The 2-D system (1) is controllable under the control variables $\boldsymbol{u}$ and $\boldsymbol{w}$.

Assumption 2. For the boundary conditions for the 2-D system (1), the terms $\sum_{k=0}^{\infty}\boldsymbol{x}^{\mathrm{vT}}(k,0)\,\boldsymbol{x}^{\mathrm{v}}(k,0)$, $\sum_{l=0}^{\infty}\boldsymbol{x}^{\mathrm{hT}}(0,l)\,\boldsymbol{x}^{\mathrm{h}}(0,l)$, and $\sum_{(0,0)\le(k,l)<(\infty,\infty)}\sum\boldsymbol{x}^{\mathrm{vT}}(k,0)\times\boldsymbol{x}^{\mathrm{h}}(0,l)$ are all bounded.

Assumption 3. There exists an unique saddle point of the zero-sum game for the 2-D system (1).

There are some important characters that must be pointed out. Firstly, for the 1-D control system, the boundary condition is just an initial point of state, while the boundary conditions of 2-D system are two given state curves along two different directions. Secondly, for the zero-sum games of 2-D system under the infinite time horizon, the boundary state trajectories are uncontrollable and so the terms $\sum_{l=0}^{\infty}\boldsymbol{x}^{\mathrm{T}}(k,0)Q\boldsymbol{x}(k,0)$, $\sum_{l=0}^{\infty}\boldsymbol{x}^{\mathrm{T}}(0,l)Q\boldsymbol{x}(0,l)$, and $\sum_{(0,0)\le(k,l)<(\infty,\infty)}\sum\boldsymbol{x}^{\mathrm{T}}(k,0)Q\boldsymbol{x}(0,l)$ may be infinite, which means the performance index function (6) is infinite. Therefore, Assumption 2 is necessary. Thirdly, the boundary conditions $\boldsymbol{f}(l)$ and $\boldsymbol{g}(k)$ in (2) should be boundary, but not necessary smooth or continuous functions. For example, let

$$\boldsymbol{f}(l)=\begin{cases}\boldsymbol{c}, & l\le T\\ \boldsymbol{0}, & l>T\end{cases} \tag{7}$$

where c is any real constant number and T is a given real number. So, Assumption 2 is not very strong.

According to Assumption 3, the optimal performance index function can be expressed as

$$\begin{aligned}V^{*}(\boldsymbol{x}(k,l))=&\min_{\boldsymbol{u}}\max_{\boldsymbol{w}}\sum_{(k,l)\le(i,j)<(\infty,\infty)}\sum\Big(\boldsymbol{x}^{\mathrm{T}}(i,j)Q\boldsymbol{x}(i,j)+\\ &\boldsymbol{u}^{\mathrm{T}}(i,j)R\boldsymbol{u}(i,j)+\boldsymbol{w}^{\mathrm{T}}(i,j)S\boldsymbol{w}(i,j)\Big)=\\ &\max_{\boldsymbol{w}}\min_{\boldsymbol{u}}\sum_{(k,l)\le(i,j)<(\infty,\infty)}\sum\Big(\boldsymbol{x}^{\mathrm{T}}(i,j)Q\boldsymbol{x}(i,j)+\\ &\boldsymbol{u}^{\mathrm{T}}(i,j)R\boldsymbol{u}(i,j)+\boldsymbol{w}^{\mathrm{T}}(i,j)S\boldsymbol{w}(i,j)\Big)\end{aligned} \tag{8}$$

2 The optimal control for the zero-sum games for 2-D systems

For zero-sum games for 1-D systems, the optimal performance index function can be written by a recurrent formulation according to the dynamic programming principle[36]. However, for zero-sum games for 2-D systems, the dynamic programming principle may not be true. The main difficulty lies in the state of the 2-D system in the next stage coupling with the states of two different directions in the current stage and then the dynamic programming equation of the zero-sum games for 2-D systems does not exist. So in this paper, we propose an optimality principle for 2-D system and obtain the expressions of optimal control pair for the zero-sum game.

2.1 The optimality principle for zero-sum games for 2-D systems

In this subsection, we will propose the optimality principle for the zero-sum games for 2-D systems, and discuss the properties of the optimal control pair derived by the principle.

Theorem 1. Given the performance index function defined as (6), if $\boldsymbol{u}(k,l)$ minimizes and $\boldsymbol{w}(k,l)$ maximizes the performance index function (6), respectively, subject to the system equation (1), and then there are $(n+m)$-dimensional vector sequences $\boldsymbol{\lambda}(k,l)$ and $\boldsymbol{\lambda}^{+}(k,l)$ defined as

$$\boldsymbol{\lambda}^{+}(k,l)=\begin{bmatrix}\boldsymbol{\lambda}^{\mathrm{h}}(k+1,l)\\ \boldsymbol{\lambda}^{\mathrm{v}}(k,l+1)\end{bmatrix},\quad \boldsymbol{\lambda}(k,l)=\begin{bmatrix}\boldsymbol{\lambda}^{\mathrm{h}}(k,l)\\ \boldsymbol{\lambda}^{\mathrm{v}}(k,l)\end{bmatrix} \tag{9}$$

where $\boldsymbol{\lambda}^{\mathrm{h}}(k,l)\in\mathbf{R}^{n_1}$ and $\boldsymbol{\lambda}^{\mathrm{v}}(k,l)\in\mathbf{R}^{n_2}$, such that for all $(0,0)\le(k,l)<(\infty,\infty)$

1) State equation:

$$\boldsymbol{x}^{+}(k,l)=\frac{\partial H(k,l)}{\partial\boldsymbol{\lambda}^{+}(k,l)} \tag{10}$$

2) Costate equation:

$$\boldsymbol{\lambda}(k,l)=\frac{\partial H(k,l)}{\partial\boldsymbol{x}(k,l)} \tag{11}$$

3) Stationarity equation:

$$\mathbf{0} = \frac{\partial H(k,l)}{\partial \boldsymbol{u}(k,l)}, \quad \mathbf{0} = \frac{\partial H(k,l)}{\partial \boldsymbol{w}(k,l)} \tag{12}$$

where $H(k,l)$ is a Hamilton function defined as

$$\begin{aligned} H(k,l) =& \boldsymbol{x}^{\mathrm{T}}(k,l)Q\boldsymbol{x}(k,l) + \boldsymbol{u}^{\mathrm{T}}(k,l)R\boldsymbol{u}(k,l) + \\ & \boldsymbol{w}^{\mathrm{T}}(k,l)S\boldsymbol{w}(k,l) + \boldsymbol{\lambda}^{+\mathrm{T}}(k,l)(A\boldsymbol{x}(k,l) + \\ & B\boldsymbol{u}(k,l) + C\boldsymbol{w}(k,l)) \end{aligned} \tag{13}$$

Proof. By examining the gradients of each of the state equations, i.e., vector of partial derivatives with respect to all the variables $\boldsymbol{x}(k,l)$, $\boldsymbol{u}(k,l)$, and $\boldsymbol{w}(k,l)$ appearing in (6), they are easily seen to be linearly independent. The optimum of the performance index function (6) is, therefore, a regular point of system (1) (see [37], pp.187). The existence of linear independent $\boldsymbol{\lambda}(k,l)$, $(0,0) \le (k,l) < (\infty,\infty)$ is immediately from the Lagrange multiplier theory (see [37], pp.187–198).

Let

$$\begin{aligned} V'(\boldsymbol{x}(k,l),\boldsymbol{u},\boldsymbol{w}) = & \sum_{(k,l)\le(i,j)<(\infty,\infty)} \sum \Big\{ \boldsymbol{x}^{\mathrm{T}}(i,j)Q\boldsymbol{x}(i,j) + \\ & \boldsymbol{u}^{\mathrm{T}}(i,j)R\boldsymbol{u}(i,j) + \boldsymbol{w}^{\mathrm{T}}(i,j)S\boldsymbol{w}(i,j) + \\ & \boldsymbol{\lambda}^{+\mathrm{T}}(i,j)(A\boldsymbol{x}(i,j) + B\boldsymbol{u}(i,j) + C\boldsymbol{w}(i,j) - \\ & \boldsymbol{x}^{+}(i,j)) \Big\} \end{aligned} \tag{14}$$

We introduce the Hamilton function of (13) and rewrite (14) as

$$\begin{aligned} & V'(\boldsymbol{x}(k,l),\boldsymbol{u},\boldsymbol{w}) = \\ & \sum_{(k,l)\le(i,j)<(\infty,\infty)} \sum \Big\{ H(i,j) - \boldsymbol{\lambda}^{+\mathrm{T}}(i,j)\boldsymbol{x}^{+}(i,j) \Big\} \end{aligned} \tag{15}$$

The last term in the previous double summation can be be expanded as

$$\begin{aligned} & \sum_{(k,l)\le(i,j)<(\infty,\infty)} \sum \boldsymbol{\lambda}^{+\mathrm{T}}(i,j)\boldsymbol{x}^{+}(i,j) = \\ & \sum_{(k,l)\le(i,j)<(\infty,\infty)} \sum \left[\boldsymbol{\lambda}^{\mathrm{T}}(i,j)\boldsymbol{x}(i,j) - \sum_{j=l}^{\infty} \boldsymbol{\lambda}^{\mathrm{hT}}(0,j)\boldsymbol{x}^{\mathrm{h}}(0,j) - \right. \\ & \left. \sum_{i=k}^{\infty} \boldsymbol{\lambda}^{\mathrm{vT}}(i,0)\boldsymbol{x}^{\mathrm{h}}(i,0) \right] \end{aligned} \tag{16}$$

According to the Lagrange multiplier theory, the increment V' due to increments in all $\boldsymbol{x}(k,l)$, $\boldsymbol{u}(k,l)$, $\boldsymbol{w}(k,l)$, and $\boldsymbol{\lambda}(k,l)$ must be zero at a constrained minimum. Hence,

$$\begin{aligned} \mathrm{d}V'(\boldsymbol{x}(k,l),\boldsymbol{u},\boldsymbol{w}) = & \sum_{(k,l)\le(i,j)<(\infty,\infty)} \sum \left\{ \left[\frac{\partial H(i,j)}{\partial \boldsymbol{u}(i,j)} \right] \mathrm{d}\boldsymbol{u}(i,j) + \right. \\ & \left[\frac{\partial H(i,j)}{\partial \boldsymbol{w}(i,j)} \right] \mathrm{d}\boldsymbol{w}(i,j) + \\ & \left[\frac{\partial H(i,j)}{\partial \boldsymbol{\lambda}^{+}(i,j)} - \boldsymbol{x}^{+}(i,j) \right] \mathrm{d}\boldsymbol{\lambda}^{+}(i,j) + \\ & \left. \left[\frac{\partial H(i,j)}{\partial \boldsymbol{x}(i,j)} - \boldsymbol{\lambda}(i,j) \right] \mathrm{d}\boldsymbol{x}(i,j) \right\} \end{aligned} \tag{17}$$

Equation (17) yields (10) $\sim$ (12), with the following remarks.

1) Increments $\mathrm{d}\boldsymbol{u}(i,j)$, $\mathrm{d}\boldsymbol{w}(i,j)$, $\mathrm{d}\boldsymbol{x}(i,j)$, and $\mathrm{d}\boldsymbol{\lambda}(i,j)$, with the exception of $\mathrm{d}\boldsymbol{x}^{\mathrm{h}}(0,j)$ and $\mathrm{d}\boldsymbol{x}^{\mathrm{v}}(i,0)$, are independent arbitrary vectors.

2) $\mathrm{d}\boldsymbol{x}^{\mathrm{h}}(0,j) = 0$ and $\mathrm{d}\boldsymbol{x}^{\mathrm{v}}(i,0) = 0$, since $\boldsymbol{x}^{\mathrm{h}}(i,j)$ and $\boldsymbol{x}^{\mathrm{v}}(i,j)$ are fixed boundary conditions. □

According to (12), the optimal control $\boldsymbol{u}^{*}(k,l)$ and $\boldsymbol{w}^{*}(k,l)$ can be expressed as

$$\boldsymbol{u}^{*}(k,l) = -\frac{1}{2}R^{-1}B^{\mathrm{T}}\boldsymbol{\lambda}^{+}(k,l) \tag{18}$$

and

$$\boldsymbol{w}^{*}(k,l) = -\frac{1}{2}S^{-1}C^{\mathrm{T}}\boldsymbol{\lambda}^{+}(k,l) \tag{19}$$

Theorem 2. For system (1) with respect to the performance index function (6), if the controls $\boldsymbol{u}(k,l)$ and $\boldsymbol{w}(k,l)$ are expressed as (18) and (19), respectively, then the optimal Hamilton function (13) satisfies a certain Riccati function.

Proof. For the zero-sum games of 2-D linear system, the optimal state feedback control should also be linear depending on the system state. As the system function is time-invariant, there exists matrix P that satisfies

$$\boldsymbol{\lambda}(k,l) = 2P\boldsymbol{x}(k,l) \tag{20}$$

Then, (18) and (19) can be rewritten as

$$\boldsymbol{u}^{*}(k,l) = -B^{\mathrm{T}}P(A\boldsymbol{x}(k,l) + B\boldsymbol{u}(k,l) + C\boldsymbol{w}(k,l)) \tag{21}$$

and

$$\boldsymbol{w}^{*}(k,l) = -C^{\mathrm{T}}P(A\boldsymbol{x}(k,l) + B\boldsymbol{u}(k,l) + C\boldsymbol{w}(k,l)) \tag{22}$$

So, the optimal state feedback controls $\boldsymbol{u}(k,l)$ and $\boldsymbol{w}(k,l)$ can be expressed as

$$\begin{aligned} \boldsymbol{u}^{*}(k,l) = & -(R + B^{\mathrm{T}}PB - B^{\mathrm{T}}PC(S + C^{\mathrm{T}}PC)^{-1}C^{\mathrm{T}}PB)^{-1} \times \\ & (B^{\mathrm{T}}PA - B^{\mathrm{T}}PC(S + C^{\mathrm{T}}PC)^{-1}C^{\mathrm{T}}PA)\boldsymbol{x}(k,l) \end{aligned} \tag{23}$$

and

$$\begin{aligned} \boldsymbol{w}^{*}(k,l) = & -(S + C^{\mathrm{T}}PC - C^{\mathrm{T}}PB(R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PC)^{-1} \times \\ & (C^{\mathrm{T}}PA - C^{\mathrm{T}}PB(R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA)\boldsymbol{x}(k,l) \end{aligned} \tag{24}$$

According to (11), we have

$$\begin{aligned} 2P\boldsymbol{x}(k,l) = & 2Q\boldsymbol{x}(k,l) + 2A^{\mathrm{T}}P\boldsymbol{x}^{+}(k,l) = \\ & 2Q\boldsymbol{x}(k,l) + 2A^{\mathrm{T}}P(A\boldsymbol{x}(k,l) + B\boldsymbol{u}^{*}(k,l) + \\ & C\boldsymbol{w}^{*}(k,l)) \end{aligned} \tag{25}$$

Substituting (23) and (24) into (25), we have the following Riccati function

$$\begin{aligned} P = & Q + A^{\mathrm{T}}PA - A^{\mathrm{T}}PB(R + B^{\mathrm{T}}PB - B^{\mathrm{T}}PC(S + \\ & C^{\mathrm{T}}PC)^{-1}C^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA + A^{\mathrm{T}}PB(R + B^{\mathrm{T}}PB - \\ & B^{\mathrm{T}}PC(S + C^{\mathrm{T}}PC)^{-1}C^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PC(S + \\ & C^{\mathrm{T}}PC)^{-1}C^{\mathrm{T}}PA - A^{\mathrm{T}}PC(S + C^{\mathrm{T}}PC - C^{\mathrm{T}}PB \times \\ & (R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PC)^{-1}C^{\mathrm{T}}PA + A^{\mathrm{T}}PC(S + \\ & C^{\mathrm{T}}PC - C^{\mathrm{T}}PB(R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PC)^{-1}C^{\mathrm{T}}PB \times \\ & (R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA \end{aligned} \tag{26}$$

□

As the zero-sum game has a saddle point and is solvable, in order to obtain the unique feedback saddle point in the class of strictly feedback stabilizing control policy, the following inequalities should be satisfied[8]

$$P > 0 \tag{27}$$

$$S + C^{\mathrm{T}} P C < 0 \tag{28}$$

and

$$R + B^{\mathrm{T}} P B > 0 \tag{29}$$

Theorem 3. For system (1) with respect to the performance index function (6), if the optimal controls $\boldsymbol{u}^*(k,l)$ and $\boldsymbol{w}^*(k,l)$ are expressed as (18) and (19), respectively, then the optimal performance index function $V^*(\boldsymbol{x}(k,l))$ is a quadratic function depending on state $\boldsymbol{x}(k,l)$.

Proof. Substituting (18) and (19) into the Hamilton function (13), we have

$$\begin{aligned} H(k,l) =& \boldsymbol{x}^{\mathrm{T}}(k,l) Q \boldsymbol{x}(k,l) + \frac{1}{4} \boldsymbol{\lambda}^{+\mathrm{T}}(k,l) B R^{-1} B^{\mathrm{T}} \boldsymbol{\lambda}^{+}(k,l) + \\ & \frac{1}{4} \boldsymbol{\lambda}^{+\mathrm{T}}(k,l) C S^{-1} C^{\mathrm{T}} \boldsymbol{\lambda}^{+}(k,l) + \boldsymbol{\lambda}^{+}(k,l) (A \boldsymbol{x}(k,l) - \\ & \frac{1}{2} B R^{-1} B^{\mathrm{T}} \boldsymbol{\lambda}^{+}(k,l) - \frac{1}{2} C S^{-1} C^{\mathrm{T}} \boldsymbol{\lambda}^{+}(k,l)) \end{aligned} \tag{30}$$

Then, according to (20), (23), and (24), we have

$$\begin{aligned} H(k,l) =& \boldsymbol{x}^{\mathrm{T}}(k,l) \big(Q + A^{\mathrm{T}} P A - A^{\mathrm{T}} P B (R + B^{\mathrm{T}} P B - \\ & B^{\mathrm{T}} P C (S + C^{\mathrm{T}} P C)^{-1} C^{\mathrm{T}} P B)^{-1} B^{\mathrm{T}} P A + \\ & A^{\mathrm{T}} P B (R + B^{\mathrm{T}} P B - B^{\mathrm{T}} P C (S + C^{\mathrm{T}} P C)^{-1} \times \\ & C^{\mathrm{T}} P B)^{-1} B^{\mathrm{T}} P C (S + C^{\mathrm{T}} P C)^{-1} C^{\mathrm{T}} P A - \\ & A^{\mathrm{T}} P C (S + C^{\mathrm{T}} P C - C^{\mathrm{T}} P B (R + B^{\mathrm{T}} P B)^{-1} \times \\ & B^{\mathrm{T}} P C)^{-1} C^{\mathrm{T}} P A + A^{\mathrm{T}} P C (S + C^{\mathrm{T}} P C - \\ & C^{\mathrm{T}} P B (R + B^{\mathrm{T}} P B)^{-1} B^{\mathrm{T}} P C)^{-1} C^{\mathrm{T}} P B \times \\ & (R + B^{\mathrm{T}} P B)^{-1} B^{\mathrm{T}} P A) \big) \boldsymbol{x}(k,l) \end{aligned} \tag{31}$$

According to (26) and the optimality principle, we immediately have

$$H(k,l) = \boldsymbol{x}^{\mathrm{T}}(k,l) P \boldsymbol{x}(k,l) = V^*(\boldsymbol{x}(k,l)) \tag{32}$$

So, $V^*(\boldsymbol{x}(k,l))$ is a quadratic function depending on state $\boldsymbol{x}(k,l)$. □

According to Theorem 3, we have the following corollary.

Corollary 1. For system (1) with respect to the performance index function (6), if the controls $\boldsymbol{u}(k,l)$ and $\boldsymbol{w}(k,l)$ are expressed as (18) and (19), respectively, then the system is stable.

Proof. According to the definition of the performance index function in (8), let $(k,l) \to (\infty,\infty)$, and we have

$$\begin{aligned} V^*(\boldsymbol{x}(\infty,\infty)) =& H(\infty,\infty) = \\ & \boldsymbol{x}^{\mathrm{T}}(\infty,\infty) Q \boldsymbol{x}(\infty,\infty) + \boldsymbol{u}^{*\mathrm{T}}(\infty,\infty) \times \\ & R \boldsymbol{u}^*(\infty,\infty) + \boldsymbol{w}^{*\mathrm{T}}(\infty,\infty) S \boldsymbol{w}^*(\infty,\infty) \end{aligned} \tag{33}$$

On the other side, according to (13), let $(k,l) \to (\infty,\infty)$. Then, we have

$$\begin{aligned} H(\infty,\infty) =& \boldsymbol{x}^{\mathrm{T}}(\infty,\infty) Q \boldsymbol{x}(\infty,\infty) + \boldsymbol{u}^{*\mathrm{T}}(\infty,\infty) \times \\ & R \boldsymbol{u}^*(\infty,\infty) + \boldsymbol{w}^{*\mathrm{T}}(\infty,\infty) S \boldsymbol{w}^*(\infty,\infty) + \\ & \boldsymbol{\lambda}^{+\mathrm{T}}(\infty,\infty) (A \boldsymbol{x}(\infty,\infty) + B \boldsymbol{u}(\infty,\infty) + \\ & C \boldsymbol{w}(\infty,\infty)) \end{aligned} \tag{34}$$

According to (20), we have

$$\begin{aligned} H(\infty,\infty) =& \boldsymbol{x}^{\mathrm{T}}(\infty,\infty) Q \boldsymbol{x}(\infty,\infty) + \boldsymbol{u}^{*\mathrm{T}}(\infty,\infty) \times \\ & R \boldsymbol{u}^*(\infty,\infty) + \boldsymbol{w}^{*\mathrm{T}}(\infty,\infty) S \boldsymbol{w}^*(\infty,\infty) + \\ & (A \boldsymbol{x}(\infty,\infty) + B \boldsymbol{u}(\infty,\infty) + C \boldsymbol{w}(\infty,\infty))^{\mathrm{T}} \times \\ & 2P (A \boldsymbol{x}(\infty,\infty) + B \boldsymbol{u}(\infty,\infty) + C \boldsymbol{w}(\infty,\infty)) \end{aligned} \tag{35}$$

Then, we have

$$\begin{aligned} & (A \boldsymbol{x}(\infty,\infty) + B \boldsymbol{u}(\infty,\infty) + C \boldsymbol{w}(\infty,\infty))^{\mathrm{T}} \times \\ & 2P (A \boldsymbol{x}(\infty,\infty) + B \boldsymbol{u}(\infty,\infty) + C \boldsymbol{w}(\infty,\infty)) = 0 \end{aligned} \tag{36}$$

As the optimal controls $\boldsymbol{u}^*(k,l)$ and $\boldsymbol{w}^*(k,l)$ are the state feedback controls expressed in (23) and (24), respectively, and $P > 0$, we can obtain

$$\lim_{(k,l)\to\infty} \boldsymbol{x}(k,l) = 0 \tag{37}$$

□

2.2 Data-based optimal control using adaptive critic designs

In [36], the zero-sum games for 1-D system were discussed based on dynamic programming principle. In this paper, based on the optimality principle, we will expand the method into 2-D systems.

As the optimal controls $\boldsymbol{u}^*$ and $\boldsymbol{w}^*$ are both linear feedback depending on the state, let

$$\begin{aligned} \boldsymbol{u}^*(k,l) =& K^* \boldsymbol{x}(k,l) \\ \boldsymbol{w}^*(k,l) =& L^* \boldsymbol{x}(k,l) \end{aligned} \tag{38}$$

and let

$$\begin{aligned} & H(\boldsymbol{x}(k,l), \boldsymbol{u}(k,l), \boldsymbol{w}(k,l)) = \\ & \begin{bmatrix} \boldsymbol{x}^{\mathrm{T}}(k,l) & \boldsymbol{u}^{\mathrm{T}}(k,l) & \boldsymbol{w}^{\mathrm{T}}(k,l) \end{bmatrix} H \begin{bmatrix} \boldsymbol{x}(k,l) \\ \boldsymbol{u}(k,l) \\ \boldsymbol{w}(k,l) \end{bmatrix} \end{aligned} \tag{39}$$

Then, according to (13), we have

$$\begin{aligned} \begin{bmatrix} H_{xx} & H_{xu} & H_{xw} \\ H_{ux} & H_{uu} & H_{uw} \\ H_{wx} & H_{wu} & H_{ww} \end{bmatrix} =& \begin{bmatrix} Q & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & S \end{bmatrix} + \begin{bmatrix} A & B & C \\ K^*A & K^*B & K^*C \\ L^*A & L^*B & L^*C \end{bmatrix}^{\mathrm{T}} H \times \\ & \begin{bmatrix} A & B & C \\ K^*A & K^*B & K^*C \\ L^*A & L^*B & L^*C \end{bmatrix} = \\ & \begin{bmatrix} A^{\mathrm{T}}PA + Q & A^{\mathrm{T}}PB & A^{\mathrm{T}}PC \\ B^{\mathrm{T}}PA & B^{\mathrm{T}}PB + R & B^{\mathrm{T}}PC \\ C^{\mathrm{T}}PA & C^{\mathrm{T}}PB & C^{\mathrm{T}}PC + S \end{bmatrix} \end{aligned} \tag{40}$$

where $P = \begin{bmatrix} I & K^{*\mathrm{T}} & L^{*\mathrm{T}} \end{bmatrix} H \begin{bmatrix} I \\ K^* \\ L^* \end{bmatrix}$.

According to (12) and (40), we have

$$\begin{aligned}\mathbf{0} &= \frac{\partial H(k,l)}{\partial \boldsymbol{u}(k,l)}\\ \mathbf{0} &= 2H_{ux}\boldsymbol{x}(k,l) + 2H_{uw}\boldsymbol{w}(k,l) + 2H_{uu}\boldsymbol{u}(k,l)\end{aligned} \tag{41}$$

and

$$\begin{aligned}\mathbf{0} &= \frac{\partial H(k,l)}{\partial \boldsymbol{w}(k,l)}\\ \mathbf{0} &= 2H_{wx}\boldsymbol{x}(k,l) + 2H_{wu}\boldsymbol{u}(k,l) + 2H_{ww}\boldsymbol{w}(k,l)\end{aligned} \tag{42}$$

Substituting (42) into (41), we can get

$$\begin{aligned}\boldsymbol{u}^*(k,l) = &(H_{uu} - H_{uw}H_{ww}^{-1}H_{wu})^{-1}\times\\ &(H_{uw}H_{ww}^{-1}H_{wx} - H_{ux})\boldsymbol{x}(k,l)\end{aligned} \tag{43}$$

Taking (43) into (42), we can get

$$\begin{aligned}\boldsymbol{w}^*(k,l) = &(H_{ww} - H_{wu}H_{uu}^{-1}H_{uw})^{-1}\times\\ &(H_{wu}H_{uu}^{-1}H_{ux} - H_{wx})\boldsymbol{x}(k,l)\end{aligned} \tag{44}$$

So, we have

$$K^* = (H_{uu} - H_{uw}H_{ww}^{-1}H_{wu})^{-1}(H_{uw}H_{ww}^{-1}H_{wx} - H_{ux}) \tag{45}$$

and

$$L^* = (H_{ww} - H_{wu}H_{uu}^{-1}H_{uw})^{-1}(H_{wu}H_{uu}^{-1}H_{ux} - H_{wx}) \tag{46}$$

3 Data-based iterative ACD algorithm

Although we have obtained the optimal control pair expressed in (45) and (46) with the information of the matrix H, we can see that the matrix H in (40) is also unsolvable directly. Therefore, an iterative ACD algorithm is proposed in this suction and the convergence property is also discussed.

3.1 The derivation of data-based iterative ACD algorithm

We propose the iterative ACD algorithm for zero-sum games of 2-D systems in this subsection.

We start with an initial Hamilton function $H_0(k,l) = 0$, which is not necessarily optimal, and then we obtain the function $H_i(k,l)$ for solving the following equation with the iteration performance $i \geq 0$:

$$\begin{aligned}&\left[\boldsymbol{x}^{\mathrm{T}}(k,l)\,\boldsymbol{u}^{\mathrm{T}}(k,l)\,\boldsymbol{w}^{\mathrm{T}}(k,l)\right] H_{i+1}\begin{bmatrix}\boldsymbol{x}(k,l)\\ \boldsymbol{u}(k,l)\\ \boldsymbol{w}(k,l)\end{bmatrix} =\\ &\min_{\boldsymbol{u}}\max_{\boldsymbol{w}}\left\{\left[\boldsymbol{x}^{\mathrm{T}}(k,l)\,\boldsymbol{u}^{\mathrm{T}}(k,l)\,\boldsymbol{w}^{\mathrm{T}}(k,l)\right]\begin{bmatrix}Q&0&0\\0&R&0\\0&0&S\end{bmatrix}\times\right.\\ &\left.\begin{bmatrix}\boldsymbol{x}(k,l)\\ \boldsymbol{u}(k,l)\\ \boldsymbol{w}(k,l)\end{bmatrix} + \left[\boldsymbol{x}^{\mathrm{T}}(k,l)\,\boldsymbol{u}^{\mathrm{T}}(k,l)\,\boldsymbol{w}^{\mathrm{T}}(k,l)\right]\begin{bmatrix}A&B&C\\KA&KB&KC\\LA&LB&LC\end{bmatrix}^{\mathrm{T}}\right.\\ &\left.H_i\times\begin{bmatrix}A&B&C\\KA&KB&KC\\LA&LB&LC\end{bmatrix}\begin{bmatrix}\boldsymbol{x}(k,l)\\ \boldsymbol{u}(k,l)\\ \boldsymbol{w}(k,l)\end{bmatrix}\right\} =\\ &\left[\boldsymbol{x}^{\mathrm{T}}(k,l)\,\boldsymbol{u}_i^{\mathrm{T}}(k,l)\,\boldsymbol{w}_i^{\mathrm{T}}(k,l)\right]\begin{bmatrix}Q&0&0\\0&R&0\\0&0&S\end{bmatrix}\begin{bmatrix}\boldsymbol{x}(k,l)\\ \boldsymbol{u}_i(k,l)\\ \boldsymbol{w}_i(k,l)\end{bmatrix} +\\ &\left[\boldsymbol{x}^{\mathrm{T}}(k,l)\,\boldsymbol{u}_i^{\mathrm{T}}(k,l)\,\boldsymbol{w}_i^{\mathrm{T}}(k,l)\right]\begin{bmatrix}A&B&C\\K_iA&K_iB&K_iC\\L_iA&L_iB&L_iC\end{bmatrix}^{\mathrm{T}} H_i\times\\ &\begin{bmatrix}A&B&C\\K_iA&K_iB&K_iC\\L_iA&L_iB&L_iC\end{bmatrix}\begin{bmatrix}\boldsymbol{x}(k,l)\\ \boldsymbol{u}_i(k,l)\\ \boldsymbol{w}_i(k,l)\end{bmatrix}\end{aligned} \tag{47}$$

Then, according to (43) and (44), the iterative control laws can be expressed as

$$\begin{aligned}K_i = &(H_{uu}^i - H_{uw}^i(H_{ww}^i)^{-1}H_{wu}^i)^{-1}\times\\ &(H_{uw}^i(H_{ww}^i)^{-1}H_{wx}^i - H_{ux}^i)\end{aligned} \tag{48}$$

and

$$\begin{aligned}L_i = &(H_{ww}^i - H_{wu}^i(H_{uu}^i)^{-1}H_{uw}^i)^{-1}\times\\ &(H_{wu}^i(H_{uu}^i)^{-1}H_{ux}^i - H_{wx}^i)\end{aligned} \tag{49}$$

Then, iterative controls are

$$\boldsymbol{u}_i(k,l) = K_i\boldsymbol{x}(k,l) \tag{50}$$

and

$$\boldsymbol{w}_i(k,l) = L_i\boldsymbol{x}(k,l) \tag{51}$$

As we can see that the iterative control laws K_i and L_i can be updated by the H matrix without the system information. While the iteration of P_i changes into the iteration of H_i, the property of the iteration H_i should be discussed. So in the followings, the convergence and optimal properties are proposed. Also, we will show that the iteration of P_i is the same as the iteration of H_i.

3.2 The properties of data-based iterative ACD algorithm

In this subsection, the convergence analysis is conducted for the data-based iterative ACD algorithm to guarantee the iterative control pair converges to the optimum. First, we give the following lemma that is necessary for the proof.

Lemma 1[36]**.** The matrices H_{i+1}, K_i, and L_{i+1} can be expressed as

$$H_{i+1} = \begin{bmatrix}A^{\mathrm{T}}P_iA + Q & A^{\mathrm{T}}P_iB & A^{\mathrm{T}}P_iC\\ B^{\mathrm{T}}P_iA & B^{\mathrm{T}}P_iB + R & B^{\mathrm{T}}P_iC\\ C^{\mathrm{T}}P_iA & C^{\mathrm{T}}P_iB & C^{\mathrm{T}}P_iC + S\end{bmatrix} \tag{52}$$

$$\begin{aligned}K_{i+1} = &-(R + B^{\mathrm{T}}P_iB - B^{\mathrm{T}}P_iC(S + C^{\mathrm{T}}P_iC)^{-1}C^{\mathrm{T}}P_iB)^{-1}\times\\ &(B^{\mathrm{T}}P_iA - B^{\mathrm{T}}P_iC(S + C^{\mathrm{T}}P_iC)^{-1}C^{\mathrm{T}}P_iA)\end{aligned} \tag{53}$$

and

$$\begin{aligned}L_{i+1} = &-(S + C^{\mathrm{T}}P_iC - C^{\mathrm{T}}P_iB(R + B^{\mathrm{T}}P_iB)^{-1}B^{\mathrm{T}}P_iC)^{-1}\times\\ &(C^{\mathrm{T}}P_iA - C^{\mathrm{T}}P_iB(R + B^{\mathrm{T}}P_iB)^{-1}B^{\mathrm{T}}P_iA)\end{aligned} \tag{54}$$

where P_i is given as

$$P_i = \left[I\ K_i^{\mathrm{T}}\ L_i^{\mathrm{T}}\right] H_i\begin{bmatrix}I\\ K_i\\ L_i\end{bmatrix} \tag{55}$$

Proof. According to (47), we have

$$H_{i+1}=\begin{bmatrix}Q&0&0\\0&R&0\\0&0&S\end{bmatrix}+\begin{bmatrix}A&B&C\\K_iA&K_iB&K_iC\\L_iA&L_iB&L_iC\end{bmatrix}^{\mathrm{T}}H_i\begin{bmatrix}A&B&C\\K_iA&K_iB&K_iC\\L_iA&L_iB&L_iC\end{bmatrix}=\begin{bmatrix}Q&0&0\\0&R&0\\0&0&S\end{bmatrix}+\begin{bmatrix}A^{\mathrm{T}}\\B^{\mathrm{T}}\\C^{\mathrm{T}}\end{bmatrix}\begin{bmatrix}I&K_i^{\mathrm{T}}&L_i^{\mathrm{T}}\end{bmatrix}H_i\begin{bmatrix}I\\K_i^{\mathrm{T}}\\L_i^{\mathrm{T}}\end{bmatrix}\begin{bmatrix}A&B&C\end{bmatrix} \tag{56}$$

Substituting (55) into (56) yields

$$H_{i+1}=\begin{bmatrix}A^{\mathrm{T}}P_iA+Q&A^{\mathrm{T}}P_iB&A^{\mathrm{T}}P_iC\\B^{\mathrm{T}}P_iA&B^{\mathrm{T}}P_iB+R&B^{\mathrm{T}}P_iC\\C^{\mathrm{T}}P_iA&C^{\mathrm{T}}P_iB&C^{\mathrm{T}}P_iC+S\end{bmatrix} \tag{57}$$

According to (23) and (45), we have

$$\begin{aligned}K_i=&-(R+B^{\mathrm{T}}P_iB-B^{\mathrm{T}}P_iC(S+C^{\mathrm{T}}P_iC)^{-1}C^{\mathrm{T}}P_iB)^{-1}\times\\&(B^{\mathrm{T}}P_iA-B^{\mathrm{T}}P_iC(S+C^{\mathrm{T}}P_iC)^{-1}C^{\mathrm{T}}P_iA)=\\&(H^i_{uu}-H^i_{uw}(H^i_{ww})^{-1}H^i_{wu})^{-1}(H^i_{uw}(H^i_{ww})^{-1}H^i_{wx}-H^i_{ux})\end{aligned} \tag{58}$$

According to (24) and (46), we have

$$\begin{aligned}L_{\mathrm{i}}=&-(S+C^{\mathrm{T}}P_iC-C^{\mathrm{T}}P_iB(R+B^{\mathrm{T}}P_iB)^{-1}B^{\mathrm{T}}P_iC)^{-1}\times\\&(C^{\mathrm{T}}P_iA-C^{\mathrm{T}}P_iB(R+B^{\mathrm{T}}P_iB)^{-1}B^{\mathrm{T}}P_iA)=\\&(H^i_{ww}-H^i_{wu}(H^i_{uu})^{-1}H^i_{uw})^{-1}(H^i_{wu}(H^i_{uu})^{-1}H^i_{ux}-H^i_{wx})\end{aligned} \tag{59}$$

□

Lemma 2. Iterating on H_i is similar to iterating on P_i as

$$\begin{aligned}P_{i+1}=&Q+A^{\mathrm{T}}P_iA-A^{\mathrm{T}}P_iB(R+B^{\mathrm{T}}P_iB-B^{\mathrm{T}}P_iC(S+\\&C^{\mathrm{T}}P_iC)^{-1}C^{\mathrm{T}}P_iB)^{-1}B^{\mathrm{T}}P_iA+A^{\mathrm{T}}P_iB(R+\\&B^{\mathrm{T}}P_iB-B^{\mathrm{T}}P_iC(S+C^{\mathrm{T}}P_iC)^{-1}C^{\mathrm{T}}P_iB)^{-1}B^{\mathrm{T}}\times\\&P_iC(S+C^{\mathrm{T}}P_iC)^{-1}C^{\mathrm{T}}P_iA-A^{\mathrm{T}}P_iC(S+C^{\mathrm{T}}P_iC-\\&C^{\mathrm{T}}P_iB(R+B^{\mathrm{T}}P_iB)^{-1}B^{\mathrm{T}}P_iC)^{-1}C^{\mathrm{T}}P_iA+\\&A^{\mathrm{T}}P_iC(S+C^{\mathrm{T}}P_iC-C^{\mathrm{T}}P_iB(R+B^{\mathrm{T}}P_iB)^{-1}B^{\mathrm{T}}\times\\&P_iC)^{-1}C^{\mathrm{T}}P_iB(R+B^{\mathrm{T}}P_iB)^{-1}B^{\mathrm{T}}P_iA\end{aligned} \tag{60}$$

where P_i is defined in (55).

Proof. From (55), we have

$$P_{i+1}=\begin{bmatrix}I&K_{i+1}^{\mathrm{T}}&L_{i+1}^{\mathrm{T}}\end{bmatrix}H_{i+1}\begin{bmatrix}I\\K_{i+1}\\L_{i+1}\end{bmatrix} \tag{61}$$

Taking (52), we can obtain

$$P_{i+1}=\begin{bmatrix}I&K_{i+1}^{\mathrm{T}}&L_{i+1}^{\mathrm{T}}\end{bmatrix}\times\begin{bmatrix}A^{\mathrm{T}}P_iA+Q&A^{\mathrm{T}}P_iB&A^{\mathrm{T}}P_iC\\B^{\mathrm{T}}P_iA&B^{\mathrm{T}}P_iB+R&B^{\mathrm{T}}P_iC\\C^{\mathrm{T}}P_iA&C^{\mathrm{T}}P_iB&C^{\mathrm{T}}P_iC+S\end{bmatrix}\begin{bmatrix}I\\K_{i+1}\\L_{i+1}\end{bmatrix} \tag{62}$$

Substituting (53) and (54) into (62), we have

$$\begin{aligned}P_{i+1}=&Q+A^{\mathrm{T}}P_iA-A^{\mathrm{T}}P_iB(R+B^{\mathrm{T}}P_iB-B^{\mathrm{T}}P_iC(S+\\&C^{\mathrm{T}}P_iC)^{-1}C^{\mathrm{T}}P_iB)^{-1}B^{\mathrm{T}}P_iA+A^{\mathrm{T}}P_iB(R+\\&B^{\mathrm{T}}P_iB-B^{\mathrm{T}}P_iC(S+C^{\mathrm{T}}P_iC)^{-1}C^{\mathrm{T}}P_iB)^{-1}B^{\mathrm{T}}\times\\&P_iC(S+C^{T}P_iC)^{-1}C^{\mathrm{T}}P_iA-A^{\mathrm{T}}P_iC(S+\\&C^{\mathrm{T}}P_iC-C^{\mathrm{T}}P_iB(R+B^{\mathrm{T}}P_iB)^{-1}B^{\mathrm{T}}P_iC)^{-1}C^{\mathrm{T}}\times\\&P_iA+A^{\mathrm{T}}P_iC(S+C^{\mathrm{T}}P_iC-C^{\mathrm{T}}P_iB(R+B^{\mathrm{T}}P_i\times\\&B)^{-1}B^{\mathrm{T}}P_iC)^{-1}C^{\mathrm{T}}P_iB(R+B^{\mathrm{T}}P_iB)^{-1}B^{\mathrm{T}}P_iA\end{aligned} \tag{63}$$

□

From Lemma 2, we have

$$\boldsymbol{x}^{\mathrm{T}}(k,l)P_i\boldsymbol{x}(k,l)=\begin{bmatrix}\boldsymbol{x}^{\mathrm{T}}(k,l)&\boldsymbol{u}_i^{\mathrm{T}}(k,l)&\boldsymbol{w}_i^{\mathrm{T}}(k,l)\end{bmatrix}H_i\begin{bmatrix}\boldsymbol{x}(k,l)\\\boldsymbol{u}_i(k,l)\\\boldsymbol{w}_i(k,l)\end{bmatrix} \tag{64}$$

Then, (47) can be expressed as

$$\begin{aligned}H_{i+1}(k,l)=&\begin{bmatrix}\boldsymbol{x}^{\mathrm{T}}(k,l)&\boldsymbol{u}_i^{\mathrm{T}}(k,l)&\boldsymbol{w}_i^{\mathrm{T}}(k,l)\end{bmatrix}\times\\&\begin{bmatrix}Q&0&0\\0&R&0\\0&0&S\end{bmatrix}\begin{bmatrix}\boldsymbol{x}(k,l)\\\boldsymbol{u}_i(k,l)\\\boldsymbol{w}_i(k,l)\end{bmatrix}+H_i^+(k,l)=\\&\begin{bmatrix}\boldsymbol{x}^{\mathrm{T}}(k,l)&\boldsymbol{u}_i^{\mathrm{T}}(k,l)&\boldsymbol{w}_i^{\mathrm{T}}(k,l)\end{bmatrix}\times\\&\begin{bmatrix}Q&0&0\\0&R&0\\0&0&S\end{bmatrix}\begin{bmatrix}\boldsymbol{x}(k,l)\\\boldsymbol{u}_i(k,l)\\\boldsymbol{w}_i(k,l)\end{bmatrix}+\boldsymbol{x}^{+\mathrm{T}}(k,l)P_i\boldsymbol{x}^{+}(k,l)\end{aligned} \tag{65}$$

where

$$\begin{aligned}H_i^+(k,l)=&\begin{bmatrix}\boldsymbol{x}^{\mathrm{T}}(k,l)&\boldsymbol{u}_i^{\mathrm{T}}(k,l)&\boldsymbol{w}_i^{\mathrm{T}}(k,l)\end{bmatrix}\begin{bmatrix}A&B&C\\K_iA&K_iB&K_iC\\L_iA&L_iB&L_iC\end{bmatrix}^{\mathrm{T}}\times\\&H_i\times\begin{bmatrix}A&B&C\\K_iA&K_iB&K_iC\\L_iA&L_iB&L_iC\end{bmatrix}\begin{bmatrix}\boldsymbol{x}(k,l)\\\boldsymbol{u}_i(k,l)\\\boldsymbol{w}_i(k,l)\end{bmatrix}\end{aligned} \tag{66}$$

and

$$H_{i+1}(k,l)=\begin{bmatrix}\boldsymbol{x}^{\mathrm{T}}(k,l)&\boldsymbol{u}_i^{\mathrm{T}}(k,l)&\boldsymbol{w}_i^{\mathrm{T}}(k,l)\end{bmatrix}H_{i+1}\begin{bmatrix}\boldsymbol{x}(k,l)\\\boldsymbol{u}_i(k,l)\\\boldsymbol{w}_i(k,l)\end{bmatrix} \tag{67}$$

Theorem 4. For the 2-D system (1) with respect to the performance index function (6), if the saddle point exists under the state feedback controls $\boldsymbol{u}(k,l)$ and $\boldsymbol{w}(k,l)$, respectively, then the iteration on (47) will converge to the optimal performance index function.

Proof. In [6], it was shown that the iterating algebraic Riccati equation (63) is convergent, for $i\to\infty$ with $P_0=0$. S_0, let $\lim_{i\to\infty}P_i=P^*$. Then, for $i\to\infty$, we have

$$\begin{aligned}&\boldsymbol{u}(k,l)=\\&-(R+B^{\mathrm{T}}P^*B-B^{\mathrm{T}}P^*C(S+C^{\mathrm{T}}P^*C)^{-1}C^{\mathrm{T}}P^*B)^{-1}\times\\&(B^{\mathrm{T}}P^*A-B^{\mathrm{T}}P^*C(S+C^{\mathrm{T}}P^*C)^{-1}C^{\mathrm{T}}P^*A)\boldsymbol{x}(k,l)\end{aligned} \tag{68}$$

and

$$\begin{aligned}&\boldsymbol{w}(k,l)=\\&-(S+C^{\mathrm{T}}P^*C-C^{\mathrm{T}}P^*B(R+B^{\mathrm{T}}P^*B)^{-1}B^{\mathrm{T}}P^*C)^{-1}\times\\&(C^{\mathrm{T}}P^*A-C^{\mathrm{T}}P^*B(R+B^{\mathrm{T}}P^*B)^{-1}B^{\mathrm{T}}P^*A)\boldsymbol{x}(k,l)\end{aligned}\tag{69}$$

where P^* satisfies the algebraic Riccati equation (26).

According to Theorem 1, the controls $\boldsymbol{u}(k,l)$ and $\boldsymbol{w}(k,l)$ in (68) and (69) are both optimal. So, we have

$$V^*(\boldsymbol{x}(k,l))=\boldsymbol{x}^{\mathrm{T}}(k,l)P^*\boldsymbol{x}(k,l)\tag{70}$$

On the other side, according to Lemma 1, we have

$$\begin{aligned}\lim_{i\to\infty}H_{i+1}=&\lim_{i\to\infty}\begin{bmatrix}A^{\mathrm{T}}P_iA+Q & A^{\mathrm{T}}P_iB & A^{\mathrm{T}}P_iC\\ B^{\mathrm{T}}P_iA & B^{\mathrm{T}}P_iB+R & B^{\mathrm{T}}P_iC\\ C^{\mathrm{T}}P_iA & C^{\mathrm{T}}P_iB & C^{\mathrm{T}}P_iC+S\end{bmatrix}=\\&\begin{bmatrix}A^{\mathrm{T}}P^*A+Q & A^{\mathrm{T}}P^*B & A^{\mathrm{T}}P^*C\\ B^{\mathrm{T}}P^*A & B^{\mathrm{T}}P^*B+R & B^{\mathrm{T}}P^*C\\ C^{\mathrm{T}}P^*A & C^{\mathrm{T}}P^*B & C^{\mathrm{T}}P^*C+S\end{bmatrix}\end{aligned}\tag{71}$$

So, we can obtain

$$H_i\to\begin{bmatrix}A^{\mathrm{T}}P^*A+Q & A^{\mathrm{T}}P^*B & A^{\mathrm{T}}P^*C\\ B^{\mathrm{T}}P^*A & B^{\mathrm{T}}P^*B+R & B^{\mathrm{T}}P^*C\\ C^{\mathrm{T}}P^*A & C^{\mathrm{T}}P^*B & C^{\mathrm{T}}P^*C+S\end{bmatrix}\tag{72}$$

as $i\to\infty$. □

In the iterative ACD algorithm, the Hamilton function $H_i^+(k,l)$ is generally difficult to obtain. Therefore, a parameter structure is necessary to approximate the actual $H_i^+(k,l)$. In this paper, a neural network called critic network is adopted to approximate $H_i^+(k,l)$. Similarly, we adopt two neural networks (called action networks) to approximate the controls $\boldsymbol{u}(k,l)$ and $\boldsymbol{w}(k,l)$, respectively. Let the output of action networks be expressed by

$$\hat{\boldsymbol{u}}_i(k,l)=K_i\boldsymbol{x}(k,l)\tag{73}$$

and

$$\hat{\boldsymbol{w}}_i(k,l)=L_i\boldsymbol{x}(k,l)\tag{74}$$

The output of critic network is expressed by

$$\boldsymbol{z}_i^{\mathrm{T}}(k,l)H_i\boldsymbol{z}_i(k,l)=\boldsymbol{h}_i^{\mathrm{T}}\bar{\boldsymbol{z}}_i\tag{75}$$

where $\boldsymbol{z}_i(k,l)=\left[\boldsymbol{x}^{\mathrm{T}}(k,l)\,\boldsymbol{u}_i^{\mathrm{T}}(k,l)\,\boldsymbol{w}_i^{\mathrm{T}}(k,l)\right]^{\mathrm{T}}$, $\boldsymbol{z}_i(k,l)\in\mathbf{R}^{n_1+n_2+m_1+m_2=q}$, $\bar{\boldsymbol{z}}_i=(z_1^2,z_1z_2,\cdots,z_1z_q,z_2^2,z_2z_3,\cdots,z_2z_q,\cdots,z_{q-1}z_q,z_q^2)$ is the Kronecker product quadratic polynomial basis vector[11], and $\boldsymbol{h}=\boldsymbol{v}(H)$ with $\boldsymbol{v}(\cdot)$ being a vector function that acts on $q\times q$ matrix and gives a $[q(q+1)]/2\times 1$ column vector.

To solve $H_{i+1}(k,l)$, the right side of (65) can be written as

$$\begin{aligned}\mathrm{d}(\boldsymbol{z}_i(k,l),H_i)=&\left[\boldsymbol{x}^{\mathrm{T}}(k,l)\,\boldsymbol{u}_i^{\mathrm{T}}(k,l)\,\boldsymbol{w}_i^{\mathrm{T}}(k,l)\right]\times\\&\begin{bmatrix}Q&0&0\\0&R&0\\0&0&S\end{bmatrix}\begin{bmatrix}\boldsymbol{x}(k,l)\\ \boldsymbol{u}_i(k,l)\\ \boldsymbol{w}_i(k,l)\end{bmatrix}+H_i^+(k,l)\end{aligned}\tag{76}$$

which can be regarded as the desired target function satisfying

$$\boldsymbol{h}_{i+1}^{\mathrm{T}}\bar{\boldsymbol{z}}_i(k,l)=\mathrm{d}(\boldsymbol{z}_i(k,l),H_i)\tag{77}$$

So, we can obtain

$$\boldsymbol{h}_{i+1}=(\bar{\boldsymbol{z}}_i(k,l)\bar{\boldsymbol{z}}_i^{\mathrm{T}}(k,l))^{-1}\bar{\boldsymbol{z}}_i(k,l)\mathrm{d}(\boldsymbol{z}_i(k,l),H_i)\tag{78}$$

Remark 1. In (78), we can see that the matrix $\bar{\boldsymbol{z}}_i(k,l)\bar{\boldsymbol{z}}_i^{\mathrm{T}}(k,l)$ is generally invertible. To overcome this problem, two methods are proposed. First, we can compute $(\bar{\boldsymbol{z}}_i(k,l)\bar{\boldsymbol{z}}_i^{\mathrm{T}}(k,l))^{-1}$ by the Moore-Penrose pseudoinverse technique[38], where $\bar{\boldsymbol{z}}_i(k,l)\bar{\boldsymbol{z}}_i^{\mathrm{T}}(k,l)\neq 0$, for $\forall k,l$. Second, we can use the least-squares technique to obtain the inverse of matrix $\bar{\boldsymbol{z}}_i(k,l)\bar{\boldsymbol{z}}_i^{\mathrm{T}}(k,l)$. In this paper, we adopt the second method.

To implement the least-squares method, white noise is added into the controls (50) and (51), respectively. Then, we have

$$\tilde{\boldsymbol{u}}_i(k,l)=K_i\boldsymbol{x}(k,l)+\boldsymbol{\xi}_1\tag{79}$$

and

$$\tilde{\boldsymbol{w}}_i(k,l)=L_i\boldsymbol{x}(k,l)+\boldsymbol{\xi}_2\tag{80}$$

where $\boldsymbol{\xi}_1(0,\sigma_1^2)$ and $\boldsymbol{\xi}_2(0,\sigma_2^2)$ are both zero-mean white noise with variances σ_1^2 and σ_2^2, respectively. So, $\boldsymbol{z}_i(k,l)$ in (75) can be written as

$$\tilde{\boldsymbol{z}}_i(k,l)=\begin{bmatrix}\boldsymbol{x}(k,l)\\ \tilde{\boldsymbol{u}}_i(k,l)\\ \tilde{\boldsymbol{w}}_i(k,l)\end{bmatrix}=\begin{bmatrix}\boldsymbol{x}(k,l)\\ \boldsymbol{u}_i(k,l)+\boldsymbol{\xi}_1\\ \boldsymbol{w}_i(k,l)+\boldsymbol{\xi}_2\end{bmatrix}=\begin{bmatrix}\boldsymbol{x}(k,l)\\ \boldsymbol{u}_i(k,l)\\ \boldsymbol{w}_i(k,l)\end{bmatrix}+\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{\xi}_1\\ \boldsymbol{\xi}_2\end{bmatrix}\tag{81}$$

Evaluating $\boldsymbol{h}_{i+1}$ at N points $p_1,p_2,\cdots,p_N$, we have

$$\boldsymbol{h}_{i+1}=(\boldsymbol{Z}_N\boldsymbol{Z}_N^{\mathrm{T}})^{-1}\boldsymbol{Z}_N\hat{\boldsymbol{Y}}_N\tag{82}$$

where

$$\boldsymbol{Z}_N=[\bar{\boldsymbol{z}}(p_1)\ \ \bar{\boldsymbol{z}}(p_2)\ \ \cdots\ \ \bar{\boldsymbol{z}}(p_N)]\tag{83}$$

and

$$\hat{\boldsymbol{Y}}_N=[\mathrm{d}(\boldsymbol{z}(p_1),H_i)\ \ \mathrm{d}(\boldsymbol{z}(p_2),H_i)\ \ \cdots\ \ \mathrm{d}(\boldsymbol{z}(p_N),H_i)]^{\mathrm{T}}\tag{84}$$

Then, we can obtain

$$H_{i+1}=\bar{g}(\boldsymbol{h}_{i+1})\tag{85}$$

through the Kronecker method and the feedback control laws K_{i+1} and L_{i+1} can be obtained according to (48) and (49), respectively. According to the condition of the least-squares solution, the number of sampling points N should satisfy the following inequality.

$$N\geq\frac{1}{2}(2q\times(2q+1))\tag{86}$$

The least-squares method in (82) can be solved in real-time by collecting enough data points generated from $\mathrm{d}(\boldsymbol{z}_i(k,l),H_i)$ in (76). What we require to know is the state and control data $\boldsymbol{x}(k,l),\boldsymbol{u}_i(k,l),\boldsymbol{w}_i(k,l)$, and $H_i^+(k,l)$. Therefore, in the proposed iterative ACD method, the model of the system is not required to update the critic and the action network.

3.3 Summarization of data-based iterative ACD algorithm

Given the above preparation, now the data-based iterative ACD algorithm proposed in this paper is summarized as follows:

Step 1. Give the boundary condition $\boldsymbol{x}^{\mathrm{h}}(0,l)=\boldsymbol{f}(l)$ and $\boldsymbol{x}^{\mathrm{v}}(k,0)=\boldsymbol{g}(k)$. Let $P_0=0$, $K_0=0$, and $L_0=0$. Give the computation accuracy ε.

Step 2. According to the N sampling points, compute $\boldsymbol{Z}_N$ and $\hat{\boldsymbol{Y}}_N$ according to (83) and (84).

Step 3. Compute $\boldsymbol{h}_i$ according to (82) and H_i according to (84) through the Kronecker method.

Step 4. Compute the feedback control laws by

$$K_{i+1}=(H_{uu}^i-H_{uw}^i(H_{ww}^i)^{-1}H_{wu}^i)^{-1}\times (H_{uw}^i(H_{ww}^i)^{-1}H_{wx}^i-H_{ux}^i) \tag{87}$$

and

$$L_{i+1}=(H_{ww}^i-H_{wu}^i(H_{uu}^i)^{-1}H_{uw}^i)^{-1}\times (H_{wu}^i(H_{uu}^i)^{-1}H_{ux}^i-H_{wx}^i) \tag{88}$$

Step 5. If

$$\| \boldsymbol{h}_{i+1}-\boldsymbol{h}_i \|\leq \varepsilon \tag{89}$$

exit; otherwise, go to Step 6.

Step 6. Set $i=i+1$, go to Step 2.

4 Neural network implementation

In this subsection, neural networks are constructed to implement the iterative ACD algorithm. There are several ACD structures that can be chosen[27]. As HDP structure is basic and convenient to realize, we will use it to implement the iterative ACD algorithm.

Assume the number of hidden layer neurons is denoted by l, the weight matrix between the input layer and hidden layer is denoted by V, and the weight matrix between the hidden layer and output layer is denoted by W. Then, the output of three-layer neural network is represented by

$$\hat{F}(\boldsymbol{X},V,W)=W^{\mathrm{T}}\boldsymbol{\sigma}(V^{\mathrm{T}}\boldsymbol{X}) \tag{90}$$

where $\boldsymbol{\sigma}(V^{\mathrm{T}}\boldsymbol{X})\in\mathbf{R}^l$, $[\boldsymbol{\sigma}(\boldsymbol{z})]_i=\dfrac{\mathrm{e}^{z_i}-\mathrm{e}^{-z_i}}{\mathrm{e}^{z_i}+\mathrm{e}^{-z_i}}$, $i=1,\cdots,l$, are the activation function.

The neural network estimation error can be expressed by

$$F(\boldsymbol{X})=F(\boldsymbol{X},V^*,W^*)+\varepsilon(\boldsymbol{X}) \tag{91}$$

where V^* and W^* are the ideal weight parameters, and $\varepsilon(\boldsymbol{X})$ is the reconstruction error.

Here, there are three neural networks, which are critic network, action network $\boldsymbol{u}$, and action network $\boldsymbol{w}$. All the neural networks are chosen as three-layer feedforward networks. The whole structure diagram is shown in Fig. 1. The utility term in the figure denotes $\boldsymbol{x}^{\mathrm{T}}(k,l)Q\boldsymbol{x}(k,l)+\boldsymbol{u}^{\mathrm{T}}(k,l)R\boldsymbol{u}(k,l)+\boldsymbol{w}^{\mathrm{T}}(k,l)S\boldsymbol{w}(k,l)$.

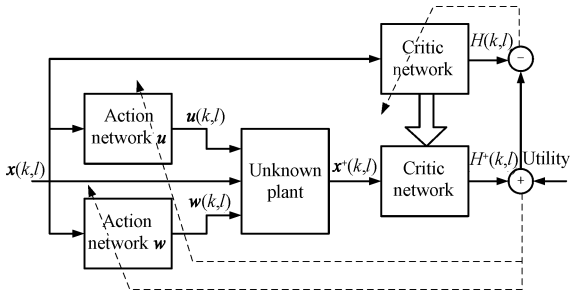


Fig. 1 The structure diagram of the algorithm

4.1 The critic network

The critic network is used to approximate the Hamilton function $H(k,l)$. The output of the critic network is denoted as

$$\hat{H}_i(k,l)=W_{ci}^{\mathrm{T}}\boldsymbol{\sigma}(V_{ci}^{\mathrm{T}}\boldsymbol{x}(k,l)) \tag{92}$$

The target function can be written as

$$H_{i+1}(k,l)=H_i^+(k,l)+\begin{bmatrix}\boldsymbol{x}^{\mathrm{T}}(k,l) & \boldsymbol{u}_i^{\mathrm{T}}(k,l) & \boldsymbol{w}_i^{\mathrm{T}}(k,l)\end{bmatrix}\begin{bmatrix}Q&0&0\\0&R&0\\0&0&S\end{bmatrix}\begin{bmatrix}\boldsymbol{x}(k,l)\\\boldsymbol{u}_i(k,l)\\\boldsymbol{w}_i(k,l)\end{bmatrix} \tag{93}$$

Then, we define the error function for the critic network as

$$e_{ci}(k,l)=\hat{H}_{i+1}(k,l)-H_{i+1}(k,l) \tag{94}$$

And, the objective function to be minimized in the critic network is

$$E_{ci}(k,l)=\frac{1}{2}e_{ci}^2(k,l) \tag{95}$$

So the gradient-based weight updating rule[39] for the critic network is given by

$$w_{c(i+1)}(k,l)=w_{ci}(k,l)+\Delta w_{ci}(k,l) \tag{96}$$

$$\Delta w_{ci}(k,l)=\alpha_c\left[-\frac{\partial E_{ci}(k,l)}{\partial w_{ci}(k,l)}\right] \tag{97}$$

$$\frac{\partial E_{ci}(k,l)}{\partial w_{ci}(k,l)}=\frac{\partial E_{ci}(k,l)}{\partial \hat{H}_i(k,l)}\frac{\partial \hat{H}_i(k,l)}{\partial w_{ci}(k,l)} \tag{98}$$

where $\alpha_c>0$ is the learning rate of critic network and $\boldsymbol{w}_c(k,l)$ is the weight vector in the critic network.

4.2 The action networks

Action networks are used to approximate the iterative optimal controls. There are two action networks, which are used to approximate the optimal controls $\boldsymbol{u}$ and $\boldsymbol{w}$, respectively.

For the action network that approximates the control $\boldsymbol{u}(k,l)$, state $\boldsymbol{x}(k,l)$ is used as the input to create the optimal control and $\boldsymbol{u}(k,l)$ is used as the output of the network. The output can be formulated as

$$\hat{u}_i(k,l)=W_{ai}^{\mathrm{T}}\boldsymbol{\sigma}(V_{ai}^{\mathrm{T}}\boldsymbol{x}(k,l)) \tag{99}$$

So, we can define the output error of the action network as

$$e_{ai}(k,l)=\hat{u}_i(k,l)-u_i(k,l) \tag{100}$$

where $u_i(k,l)$ is the target function that can be described by

$$u_i(k,l)=(H_{ww}^i-H_{wu}^i(H_{uu}^i)^{-1}H_{uw}^i)^{-1}\times (H_{wu}^i(H_{uu}^i)^{-1}H_{ux}^i-H_{wx}^i)\boldsymbol{x}(k,l) \tag{101}$$

where H_i can be obtained according to Kronecker product in (85).

The weighs in the action network are updated to minimize the following performance error measure:

$$E_{ai}(k,l)=\frac{1}{2}e_{ai}^2 \tag{102}$$

The weight updating algorithm is similar to the one for the critic network. By the gradient descent rule, we can obtain

$$w_{a(i+1)}(k,l) = w_{ai}(k,l) + \Delta w_{ai}(k,l) \tag{103}$$

$$\Delta w_{ai}(k,l) = \beta_a \left[-\frac{\partial E_{ai}(k,l)}{\partial w_{ai}(k,l)} \right] \tag{104}$$

$$\frac{\partial E_{ai}(k,l)}{\partial w_{ai}(k,l)} = \frac{\partial E_{ai}(k,l)}{\partial e_{ai}(k,l)} \frac{\partial e_{ai}(k,l)}{\partial u_i(k,l)} \frac{\partial u_i(k,l)}{\partial w_{ai}(k,l)} \tag{105}$$

where $\beta_a > 0$ is the learning rate of the action network.

For the action network $\boldsymbol{w}$ that approximates the control $\boldsymbol{w}(k,l)$, state $\boldsymbol{x}(k,l)$ is used as the input to create the optimal control and $\boldsymbol{w}(k,l)$ is used as the output of the network. The target of $\boldsymbol{w}$ action network can be expressed as

$$\begin{aligned} w_i(k,l) =& (H^i_{ww} - H^i_{wu}(H^i_{uu})^{-1}H^i_{uw})^{-1} \times \\ & (H^i_{wu}(H^i_{uu})^{-1}H^i_{ux} - H^i_{wx})\boldsymbol{x}(k,l) \end{aligned} \tag{106}$$

All the update rules of $\boldsymbol{w}$ action network are the same as the update rules of $\boldsymbol{u}$ network and it is omitted here.

5 Simulation

In this section, the proposed method is applied to an air drying process control. Our example is a modification of Example 1 in [40] and extends the variable space to the infinite horizon.

The dynamical processes can be described by the following Darboux equation:

$$\begin{aligned} \frac{\partial^2 x(s,t)}{\partial s \partial t} =& a_1 \frac{\partial x(s,t)}{\partial t} + a_2 \frac{\partial x(s,t)}{\partial x} + a_0 x(s,t) + \\ & bu(s,t) + cw(s,t) \end{aligned} \tag{107}$$

with the initial and boundary conditions

$$x^{\mathrm{h}}(0,t) = \begin{cases} 0.5, & t \le 4 \\ 0, & t > 4 \end{cases}, \; x^{\mathrm{v}}(s,0) = \begin{cases} 1, & s \le 4 \\ 0, & s > 4 \end{cases} \tag{108}$$

where $x(s,t)$ is an unknown function, a_0, a_1, a_2, b, and c are real coefficients, $u(s,t)$ and $w(s,t)$ are the input functions. The variable x means the humidity, which is the system state, s means the location of the air, and t is the processing time.

Let $a_0 = 0.2$, $a_1 = 0.3$, $a_2 = 0.1$, $b = 0.3$, and $c = 0.25$. The quadratic performance index function is formulated as

$$V = \int_{t=0}^{\infty} \int_{s=0}^{\infty} \left\{ Qx^2(s,t) + Ru^2(s,t) + Sw^2(s,t) \right\} \mathrm{d}s\mathrm{d}t \tag{109}$$

The discretization method for system (107) is similar to the method in [40]. Suppose that the sampling periods of the digital control system are chosen as $X = 0.1\,\mathrm{cm}$ and $T = 0.1\,\mathrm{s}$. Following the methodology presented in [40], we can compute the discretized system equation (1) as

$$\begin{aligned} \begin{bmatrix} x^{\mathrm{h}}((k+1)X, lT) \\ x^{\mathrm{v}}(kX, (l+1)T) \end{bmatrix} =& \begin{bmatrix} 0.7408 & 0.2765 \\ 0.0952 & 0.9048 \end{bmatrix} \begin{bmatrix} x^{\mathrm{h}}(kX, lT) \\ x^{\mathrm{v}}(kX, lT) \end{bmatrix} + \\ & \begin{bmatrix} 0.0259 \\ 0 \end{bmatrix} \times u(kX, lT) + \begin{bmatrix} 0 \\ 0.0564 \end{bmatrix} w(kX, lT) \end{aligned} \tag{110}$$

with the boundary conditions

$$x^{\mathrm{h}}(0, lT) = \begin{cases} 0.5, & l \le 40 \\ 0, & l > 40 \end{cases}, \; x^{\mathrm{v}}(kX, 0) = \begin{cases} 1, & k \le 40 \\ 0, & k > 40 \end{cases} \tag{111}$$

and the discredited performance index function as

$$V = \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \left\{ Qx^2(Xk, Tl) + Ru^2(Xk, Tl) + Sw^2(Xk, Tl) \right\} \tag{112}$$

We implement the iterative algorithm at $(k,l) = (0,0)$. We choose three-layer neural networks as the critic network, the action network u, the action network w with the structures 2-8-1, 2-8-1, and 2-8-1, respectively. The initial weights of action networks and critic network are all set to be random in $[-0.5, 0.5]$. Then, the critic network and the action network are trained for $i = 50$ times so that the given accuracy $\varepsilon = 10^{-6}$ is reached. In the training process, the learning rate $\beta_a = \alpha_c = 0.05$. The evaluating point number $N = 40$ for every iteration and choose the small white noise as $\xi_1(0, 0.01)$ and $\xi_2(0, 0.01)$. The convergence curve of the performance index function is shown in Fig. 2. Then, we apply the optimal control to the system for $k = 40$, $l = 40$ time steps and obtain the following results. The state trajectories are given as Figs. 3 and 4. The control curves are given as Figs. 5 and 6, respectively.

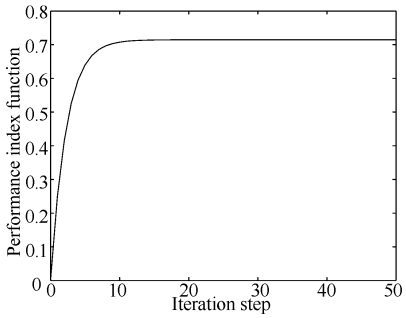


Fig. 2 The convergence of performance index function

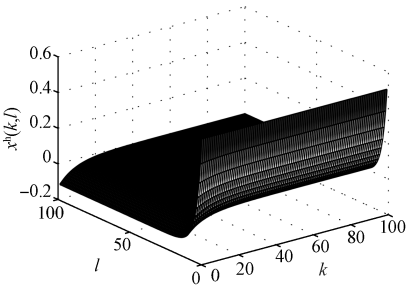


Fig. 3 The state variables x^{h} trajectories

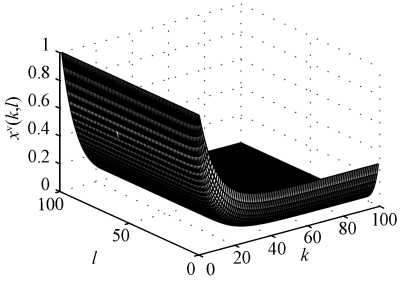


Fig. 4 The state variables x^{v} trajectories

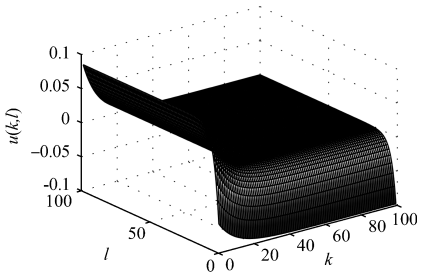


Fig. 5 The optimal control u trajectories

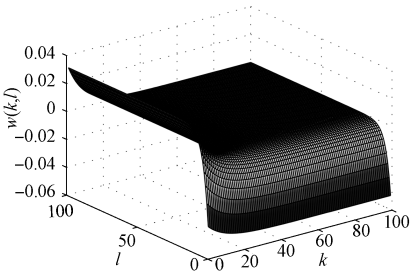


Fig. 6 The optimal control w trajectories

From the simulation results, we can see that the proposed iterative ACD algorithm in this paper obtains good effects. In [40], Tsai just studied the model-based optimal control in the finite horizon. In this paper, using the iterative ACD algorithm, the optimal control scheme for 2-D system in the infinite horizon can also be obtained without the system model. So the proposed algorithm in this paper is more effective than the method in [40] for industry process control.

6 Conclusion

In this paper, we proposed an effective iterative algorithm to find the optimal controller of a class of discrete-time two-person zero-sum games for Roesser types 2-D systems. The proposed ACD algorithm allows to be implemented without the system model. Stability analysis of the 2-D systems was presented and the convergence property of the performance index function was also proved. The simulation study has successfully demonstrated the upstanding performance of the proposed optimal control scheme for the 2-D systems.

References

1 Jamshidi M. *Large Scale Systems: Modeling, Control, and Fuzzy Logics*. Amsterdam: The Netherlands Press, 1982

2 Chang H S, Marcus S I. Two-person zero-sum Markov games: receding horizon approach. *IEEE Transactions on Automatic Control*, 2003, **48**(11): 1951−1961

3 Chen B S, Tseng C S, Uang H J. Fuzzy differential games for nonlinear stochastic systems: suboptimal approach. *IEEE Transactions on Fuzzy Systems*, 2002, **10**(2): 222−233

4 Nian Xiao-Hong, Cao Li. Design of optimal observer and optimal feedback controller based on differential game theory. *Acta Automatica Sinica*, 2006, **32**(5): 807−812 (in Chinese)

5 Nian Xiao-Hong. Suboptimal strategies of linear quadratic closed-loop differential games: a BMI approach. *Acta Automatica Sinica*, 2005, **31**(2): 216−222

6 Bertsekas D P. *Convex Analysis and Optimization*. Boston: Athena Scientific, 2003

7 Goebel R. Convexity in zero-sum differential games. *SIAM Journal of Control and Optimization*, 2001, **40**(5): 1491−1504

8 Altman E, Basar T. Multiuser rate-based flow control. *IEEE Transactions on Communications*, 1998, **46**(7): 940−949

9 Basar T, Olsder G J. *Dynamic Noncooperative Game Theory*. New York: Academic Press, 1982

10 Basar T, Bernhard P. H_∞ *Optimal Control and Related Minimax Design Problems*. Boston: Birkhauser Press, 1995

11 Hua X, Mizukami K. Linear-quadratic zero-sum differential games for generalized state space systems. *IEEE Transactions on Automatic Control*, 1994, **39**(1): 143−147

12 Wei G, Feng G, Wang Z. Robust H_∞ control for discrete-time fuzzy systems with infinite-distributed delays. *IEEE Transactions on Fuzzy Systems*, 2009, **17**(1): 224−232

13 Werbos P J. Approximate dynamic programming for real-time control and neural modeling. *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*. New York: Van Nostrand Reinhold, 1992

14 Xu Jian-Ming, Yu Li. H_∞ control for 2-D discrete state delayed systems in the second FM model. *Acta Automatica Sinica*, 2008, **34**(7): 809−813

15 Uetake Y. Optimal smoothing for noncausal 2-D systems based on a descriptor model. *IEEE Transactions on Automatic Control*, 1992, **37**(11): 1840−1845

16 Owens D H, Amann N, Rogers E, French M. Analysis of linear iterative learning control schemes — a 2D systems/repetitive processes approach. *Multidimensional Systems and Signal Processing*, 2000, **11**(1-2): 125−177

17 Sulikowski B, Galkowski K, Rogers E, Owens D H. Output feedback control of discrete linear repetitive processes. *Automatica*, 2004, **40**(12): 2167−2173

18 Li C J, Fadali M S. Optimal control of 2-D systems. *IEEE Transactions on Automatic Control*, 1991, **36**(2): 223−228

19 Liu D R, Javaherian H, Kovalenko O, Huang T. Adaptive critic learning techniques for engine torque and air-fuel ratio control. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2008, **38**(4): 988−993

20 Al-Tamimi A, Abu-Khalaf M, Lewis F L. Adaptive critic designs for discrete-time zero-sum games with application to H_∞ control. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2007, **37**(1): 240−247

21 Liu De-Rong. Approximate dynamic programming for self-learning control. *Acta Automatica Sinica*, 2005, **31**(1): 13−18

22 Ray S, Venayagamoorthy G K, Chaudhuri B, Majumder R. Comparison of adaptive critic-based and classical wide-area controllers for power systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2008, **38**(4): 1002−1007

23 Watkins C. Learning from Delayed Rewards [Ph. D. dissertation], Cambridge University, USA, 1989

24 Werbos P J. A menu of designs for reinforcement learning over time. *Neural Networks for Control*. Cambridge: MIT Press, 1991. 67−95

25 Widrow B, Gupta N K, Maitra S. Punish/reward: learning with a critic in adaptive threshold systems. *IEEE Transactions on Systems, Man, Cybernetics*, 1973, **3**(5): 455−465

26 Prokhorov D V, Wunsch D C. Adaptive critic designs. *IEEE Transactions on Neural Networks*, 1997, **8**(5): 997−1007

27 Murray J J, Cox C J, Lendaris G G, Saeks R. Adaptive dynamic programming. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 2002, **32**(2): 140−153

28 Zhang H G, Wei Q L, Luo Y H. A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm. *IEEE Transactions on Systems, Man, Cybernetics, Part B: Cybernetics*, 2008, **38**(4): 937−942

29 Liu D R, Zhang H G. A neural dynamic programming approach for learning control of failure avoidance problems. *International Journal of Intelligent Control and Systems*, 2005, **10**(1): 21−32

30 Liu D R, Zhang Y, Zhang H G. A self-learning call admission control scheme for CDMA cellular networks. *IEEE Transactions on Neural Networks*, 2005, **16**(5): 1219−1228

31 Al-Tamimi A, Lewis F L. Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2008, **38**(4): 943−949

32 Chen Z, Jagannathan S. Generalized Hamilton-Jacobi-Bellman formulation-based neural network control of affine nonlinear discretetime systems. *IEEE Transactions on Neural Networks*, 2008, **19**(1): 90−106

33 Ferrari S, Steck J E, Chandramohan R. Adaptive feedback control by constrained approximate dynamic programming. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2008, **38**(4): 982−987

34 Balakrishnan S N, Ding J, Lewis F L. Issues on stability of ADP feedback controllers for dynamical systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2008, **38**(4): 913−917

35 Seiffertt J, Sanyal S, Wunsch D C. Hamilton-Jacobi-Bellman equations and approximate dynamic programming on time scales. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2008, **38**(4): 918−923

36 Al-Tamimi A, Lewis F L, Abu-Khalaf M. Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control. *Automatica*, 2007, **43**(3): 473−481

37 Luenberger D G. *Optimization by Vector Space Methods*. New York: Wiley, 1969

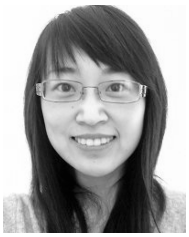
38 Zhang Xian-Da. *Matrix Analysis and Applications*. Beijing: Tsinghua University Press, 2004 (in Chinese)

39 Si J, Wang Y T. On-line learning control by association and reinforcement. *IEEE Transactions on Neural Networks*, 2001, **12**(2): 264−276

40 Tsai J S H, Li J S, Shieh L S. Discretized quadratic optimal control for continuous-time two-dimensional systems. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 2002, **49**(1): 116−125

WEI Qing-Lai Received his bachelor degree in automation control, master and Ph. D. degrees in control theory and control engineering from Northeastern University in 2002, 2005, and 2008, respectively. He is currently a postdoctor with the Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences. His research interest covers neural-networks-based control, nonlinear control, adaptive dynamic programming, and their industrial application. Corresponding author of this paper. E-mail: qinglaiwei@gmail.com

ZHANG Hua-Guang Received his Ph. D. degree from Southeast University in 1991. He is currently a professor. His research interest covers fuzzy system theory, fuzzy control, neural network-based control, adaptive control, chaotic control, complex industry process automation, electric power system automation, and motor driving system automation.
E-mail: z-hg@tom.com



CUI Li-Li Ph. D. candidate at the Institute of Information Science and Engineering, Northeastern University. Her research interest covers adaptive dynamic programming, neural networks, and optimal control. E-mail: cuilili8396@163.com