J Syst Sci Complex 20XX, XX( ): 1–16

On-policy and Off-policy Value Iteration Algorithms for Stochastic Zero-Sum Dynamic Games*

GUO Liangyuan · WANG Bing-Chang · ZHANG Ji-Feng

DOI:
Received: x x 20xx / Revised: x x 20xx

Abstract This paper considers the value iteration algorithms of stochastic zero-sum linear quadratic games with unkown dynamics. On-policy and off-policy learning algorithms are developed to solve the stochastic zero-sum games, where the system dynamics is not required. By analyzing the value function iterations, the convergence of the model-based algorithm is shown. The equivalence of several types of value iteration algorithms is established. The effectiveness of model-free algorithms is demonstrated by a numerical example.

Keywords Approximate dynamic programming, Stochastic zero-sum games, Value iteration.

1 Introduction

The zero-sum game is an important type of game, describing the decision-making process of two players [1], where one player's gain is equal to the other player's loss and hence the total gain/loss for both players always equal to zero. This type of game is widely used in the real world, such as economic competition, military strategy, sports competitions, and so on. Zero-sum games are closely related to H_∞ optimal control [2], which is a robust optimal control method that relies on solving the Hamilton-Jacobi-Bellman (HJB) equation or the Riccati equation. For nonlinear systems, one requires to solve the HJB equation, while for linear systems, it is generally needed to solve the Riccati equation [3].

GUO Liangyuan · WANG Bing-Chang (Corresponding author)
School of Control Science and Engineering, Shandong University, Jinan 250063, China;
Email: lyguo@mail.sdu.edu.cn; bcwang@sdu.edu.cn.
ZHANG Ji-Feng
School of Automation and Electrical Engineering, Zhongyuan University of Technology, Zhengzhou 450007, China; and Key Laboratory of Systems and Control, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China.
Email: jif@iss.ac.cn.
*This research was supported by the National Natural Science Foundation of China under Grant No.62122043, 62433020, T2293770, and the Innovative Research Groups of National Natural Science Foundation of China under Grant No.61821004.
◇ *This paper was recommended for publication by Editor .*

In recent years, there has been tremendous interest in reinforcement learning. Approximate dynamic programming (ADP) is a typical class of practical reinforcement learning methods for obtaining the optimal control solution without the knowledge of system dynamics [4]. The ADP method was proposed by [5], [6] and others, which aims to determine the optimal control solution by finding an approximate value function. Compared with traditional dynamic programming, ADP can circumvent the intractable computational problems [7].

For ADP, the iterative procedure involves two parts: the policy evaluation and the policy improvement [8]. In general, ADP can be divided into value iteration [9] and policy iteration [4]. For policy iteration, the iteration starts with a stabilizing initial control, while value iteration can be initiated with an arbitrary policy [10]. The convergence of the value iteration method was discussed in [11]. When the iteration sequence begins with a zero initial value function, it will converge to the optimal performance index.

Most policy iteration and value iteration methods require some knowledge of the model parameters. Accurately determining the model parameters is a challenging and time-consuming process, especially in a complex environment. Consequently, model-free methods have been proposed and applied for solving the problem in an uncertain environment (e.g. [12], [13], [14]). In practice, it is widely applied in complex situations, like handling variable road conditions in autonomous driving. In [4], the learning algorithm is implemented only using measured input/output data of the system. In [15], the policy optimization methods are developed to obtain the Nash equilibria. A novel policy iteration approach was developed in [16], which solves the algebraic Riccati equation using the online information. In [17], an off-policy algorithm is proposed using real data.

The work mentioned above mostly focused on deterministic models, which have already been widely studied and applied in numerous fields. However, due to the inherent uncertainty and complexity of stochastic systems, different methods are required to be developed. In this context, model-free reinforcement learning has gained popularity for stochastic systems, as it avoids model bias issues [14]. In [18], [19], the ADP algorithm is developed to solve the stochastic optimal control problem, which focuses on the dynamically perturbed stochastic systems. [20] designs a reinforcement learning method to solve the continuous-time stochastic linear quadratic problem, where the performance is simulated by computing the average value based on multiple sample paths.

However, there are few related works regarding the design of ADP methods for discrete-time stochastic systems.

The main contribution of the paper includes the following two-fold:

- In this paper, two model-free value iteration algorithms are developed to solve the discrete-time infinite-horizon stochastic zero-sum linear quadratic games, without using system dynamics information.
- The convergence of the value function iterations is first presented for the model-based value iteration algorithm, and then the convergence of the model-free algorithms is given by establishing the equivalence of model-based and model-free value iteration algorithms.

The equivalence of on-policy and off-policy model-free value iteration algorithms is further shown.

Notation. For matrix $X = (x_{ij})_{n\times n}$, $tr(X)$ is trace, $diag(\cdot)$ represents a diagonal matrix. $vech(X) \triangleq [x_{11}, x_{12}, ..., x_{1n}, x_{22}, x_{23}, ..., x_{n-1,n}, x_{n,n}]^T$, $vecs(X) \triangleq [x_{11}, 2x_{12}, ..., 2x_{1n}, x_{22}, 2x_{23}, ..., 2x_{n-1,n}, x_{n,n}]^T$. $\mathbb{L}$ is Banach space.

2 Problem Formulation

Consider a discrete-time stochastic system as follows:

$$x_{k+1} = Ax_k + Bu_k + Cw_k + (Dx_k + Eu_k + Fw_k)d_k, \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the state, $u_k \in \mathbb{R}^{m_1}$, $w_k \in \mathbb{R}^{m_2}$ are the control input and disturbance input, respectively. $d_k \in \mathbb{R}^1$ follows the standard normal distribution. The matrices A, B, C, D, E, F have proper dimensions. The associated cost function is given by

$$J(x_k) = \sum_{i=k}^{\infty} r^{i-k}\mathbb{E}[c(x_i, u_i, w_i)], \tag{2}$$

where r is discount factor. $c(x_i, u_i, w_i) = x_i^T Rx_i + u_i^T u_i - \gamma^2 w_i^T w_i$.

The zero-sum game problem is to find proper u_k and v_k to minmax the cost function (2). Thus, the value fuction is given by

$$V(x_k) = \min_{u_k}\max_{w_k} J(x_k), \tag{3}$$

According to Bellman's optimality principle [4], [14], the value function may be determined using the HJB equation

$$V(x_k) = \min_{u_k}\max_{w_k} \mathbb{E}[c(x_k, u_k, w_k)] + rV(x_{k+1}). \tag{4}$$

Lemma 2.1 *The value function can be rewritten as*

$$V(x_k) = \mathbb{E}(x_k^T Px_k). \tag{5}$$

where $P \geq 0$ satisfies the algebraic Riccati equation (6)

$$\begin{aligned} P =& R + rA^{\mathrm{T}}PA + rD^{\mathrm{T}}PD - [r(A^{\mathrm{T}}PB + D^{\mathrm{T}}PE) \quad r(A^{\mathrm{T}}PC + D^{\mathrm{T}}PF)] \\ &\times \begin{bmatrix} I + r(B^{\mathrm{T}}PB + E^{\mathrm{T}}PE) & r(B^{\mathrm{T}}PC + E^{\mathrm{T}}PF) \\ r(C^{\mathrm{T}}PB + F^{\mathrm{T}}PE) & r(C^{\mathrm{T}}PC + F^{\mathrm{T}}PF) - \gamma^2 I \end{bmatrix}^{-1} \begin{bmatrix} r(B^{\mathrm{T}}PA + E^{\mathrm{T}}PD) \\ r(C^{\mathrm{T}}PA + F^{\mathrm{T}}PD) \end{bmatrix}. \end{aligned} \tag{6}$$

Proof Since

$$\begin{aligned} &\mathbb{E}(x_{k+1}^T Px_{k+1}) = \mathbb{E}[\mathbb{E}(x_{k+1}^T Px_{k+1}|x_k)] \\ =&\mathbb{E}\Big[x_k^T[(A + BL + CK)^T P(A + BL + CK) + (D + EL + FK)^T P(D + EL + FK)]x_k\Big], \end{aligned}$$

we have

$$\begin{aligned} & r\mathbb{E}(x_{k+1}^T P x_{k+1}) - \mathbb{E}(x_k^T P x_k) \\ =& -\mathbb{E}[x_k^T (R + L^T L - \gamma^2 K^T K) x_k] \\ =& -\mathbb{E}[c(x_k, u_k, w_k)]. \end{aligned}$$

This can be written as

$$\mathbb{E}[c(x_k, u_k, w_k)] = \mathbb{E}(x_k^T P x_k) - r\mathbb{E}(x_{k+1}^T P x_{k+1}). \tag{7}$$

Substituting (7) into (3) results in

$$\begin{aligned} V(x_k) &= \sum_{i=k}^{\infty} r^{i-k} \mathbb{E}[c(x_i, u_i, w_i)] \\ &= \mathbb{E}(x_k^T P x_k) - \lim_{i\to\infty} r^i \mathbb{E}(x_{k+i}^T P x_{k+i}) \\ &= \mathbb{E}(x_k^T P x_k). \end{aligned}$$

■

The corresponding Hamiltonian function is defined as

$$H(x_k, L, K) = \mathbb{E}[c(x_k, u_k, w_k)] + r\mathbb{E}(x_{k+1}^T P x_{k+1}) - \mathbb{E}(x_k^T P x_k).$$

By the first-order necessary condition for optimality [14], we obtain the minimax gains

$$\begin{cases} L = [I + r(B^T P B + E^T P E) - rC^T P B \times (r(C^T P C + F^T P F) - \gamma^2)^{-1} r B^T P C]^{-1} \\ \quad \times [-rB^T P A + rC^T P A (r(C^T P C + F^T P F) - \gamma^2)^{-1} r B^T P C], \\ K = [r(C^T P C + F^T P F) - \gamma^2 - rB^T P C (I + r(B^T P B + E^T P E))^{-1} r C^T P B]^{-1} \\ \quad \times [-rC^T P A + rB^T P A (I + r(B^T P B + E^T P E))^{-1} r C^T P B]. \end{cases} \tag{8}$$

The saddle-point policies can be determined by

$$\begin{cases} u_k = L x_k, \\ w_k = K x_k. \end{cases}$$

Remark 2.2 The Riccati equation (6) can be expressed as the Lyapunov equation (9)

$$\begin{aligned} P =& R + L^T L - \gamma^2 K^T K + r[(A + BL + CK)^T P (A + BL + CK) \\ & + (D + EL + FK)^T P (D + EL + FK)]. \end{aligned} \tag{9}$$

Inserting the minimax gains (8) into (9), we get the Riccati equation (6).

The existence of a solution to the zero-sum games is guaranteed by Theorem 2.1 of [21]. Assume (C, A) is detectable and (A, B) is stabilizable. Then the two-person zero-sum games exist solution if and only if $I - C^*[\mathcal{L}P_0^+ + P_0^+\mathcal{L}^* - P_0^+\mathcal{L}^* M_0 \mathcal{L} P_0^+]C > 0$, where P_0^+ is the positive definite solution when $w_k \equiv 0$, and T^* is the conjugate transpose of T. Here, the operators $\mathcal{L}$ and $\mathcal{L}^*$ are defined as follows: $(\mathcal{L}f)(t) = \int_t^\infty e^{F'(s-t)} f(s)\, ds$, $(\mathcal{L}^* f)(t) = \int_0^t e^{F(t-s)} f(s)\, ds$.

We now turn to develop the model-based value iteration algorithm, including policy evaluation

$$\begin{aligned} P^{(i)} &= R + (L^{(i-1)})^T L^{(i-1)} - \gamma^2 (K^{(i-1)})^T K^{(i-1)} \\ &\quad + r[(A+BL^{(i-1)}+CK^{(i-1)})^T P^{(i-1)} (A+BL^{(i-1)}+CK^{(i-1)}) \\ &\quad + (D+EL^{(i-1)}+FK^{(i-1)})^T P^{(i-1)} (D+EL^{(i-1)}+FK^{(i-1)})], \end{aligned} \tag{10}$$

and policy improvement

$$\begin{aligned} &\begin{bmatrix} I + r(B^T P^{(i)} B + E^T P^{(i)} E) & r(B^T P^{(i)} C + E^T P^{(i)} F) \\ r(C^T P^{(i)} B + F^T P^{(i)} E) & r(C^T P^{(i)} C + F^T P^{(i)} F) - \gamma^2 \end{bmatrix} \\ &\times \begin{bmatrix} L^{(i+1)} \\ K^{(i+1)} \end{bmatrix} = \begin{bmatrix} -r(B^T P^{(i)} A + E^T P^{(i)} D) \\ -r(C^T P^{(i)} A + F^T P^{(i)} D) \end{bmatrix}. \end{aligned} \tag{11}$$

Algorithm 1: Model-based value iteration

Input: Initial state x_0, simulation stop time k_{end}, ε.

Output: The estimation of minimax gains $\widehat{L}, \widehat{K}$.

```
Initialization:i = 1, L^(1), K^(1)
for i = 2 : k_end do
    Obtain P^(i) according to (10).
    Update L^(i+1), K^(i+1) as (11) or (8).
    if ||L^(i+1) − L^(i)|| < ε and ||K^(i+1) − K^(i)|| < ε, then
        Break
    else
        i = i + 1
    end
end
L̂ = L^(i+1), K̂ = K^(i+1)
```

The convergence of model-based value iteration (Algorithm 1) is given in the following lemma.

Lemma 2.3 *Assume that there exist* $1 \leq \beta < \infty$, $0 \leq \underline{m} \leq 1$, *and* $1 \leq \overline{m} < \infty$ *such that* $0 \leq rV^*(x_{k+1}) \leq \beta \mathbb{E}[c(x_k, u_k, w_k)]$ *and* $0 \leq \underline{m} V^*(x_k) \leq V_0(x_k) \leq \overline{m} V^*(x_k)$. *The sequences* $\{L_i\}, \{K_i\}, \{V_i(\cdot)\}$ *are iteratively updated using the model-based value iteration algorithm. Then, the value function* $V_i(\cdot)$ *converges towards* $V^*(\cdot)$ *according to the ensuing set of inequalities:*

$$\left[1 + \frac{\underline{m} - 1}{(1+\beta^{-1})^i}\right] V^*(x) \leq V_i(x) \leq \left[1 + \frac{\overline{m} - 1}{(1+\beta^{-1})^i}\right] V^*(x). \tag{12}$$

Then,

$$\lim_{i \to \infty} V_i(x_k) = V^*(x_k), \lim_{i \to \infty} L_i = L^*, \lim_{i \to \infty} K_i = K^*.$$

Proof Since $0 \le rV^*(x_{k+1}) \le \beta\mathbb{E}[c(x_k,u_k,w_k)]$ and $0 \le \underline{m}V^*(x_k) \le V_0(x_k) \le \overline{m}V^*(x_k)$, we have

$$\frac{\underline{m}-1}{1+\beta}\Big\{\beta\mathbb{E}[c(x_k,u_k,w_k)] - rV^*(x_{k+1})\Big\} \le 0,$$
$$\frac{\overline{m}-1}{1+\beta}\Big\{\beta\mathbb{E}[c(x_k,u_k,w_k)] - rV^*(x_{k+1})\Big\} \ge 0.$$

For $i=0$, (12) holds. Let $i=1$,

$$\begin{aligned} V_1(x_k) &= \mathbb{E}[c(x_k,u_k,w_k)] + rV_0(x_{k+1}) \\ &\le \mathbb{E}[c(x_k,u_k,w_k)] + r\overline{m}V^*(x_{k+1}) \\ &\le \mathbb{E}[c(x_k,u_k,w_k)] + r\overline{m}V^*(x_{k+1}) \\ &+ \frac{\overline{m}-1}{1+\beta}\Big\{\beta\mathbb{E}[c(x_k,u_k,w_k)] - rV^*(x_{k+1})\Big\} \\ &= \frac{1+\overline{m}\beta}{1+\beta}\Big[\mathbb{E}[c(x_k,u_k,w_k)] + rV^*(x_{k+1})\Big] \\ &= \big[1+\frac{\overline{m}-1}{1+\beta^{-1}}\big]V^*(x_k). \end{aligned} \tag{13}$$

Suppose for $i=j-1$, $j=1,2,3,...$, the conclusion still holds, i.e.

$$V_{j-1}(x_k) \le \left[1+\frac{\overline{m}-1}{(1+\beta^{-1})^{j-1}}\right]V^*(x_k).$$

Then, for $i=j$, one has

$$\begin{aligned} V_j(x_k) &= \mathbb{E}[c(x_k,u_k,w_k)] + rV_{j-1}(x_{k+1}) \\ &\le \mathbb{E}[c(x_k,u_k,w_k)] + r\left[1+\frac{\overline{m}-1}{(1+\beta^{-1})^{j-1}}\right]V^*(x_{k+1}) \\ &\le \mathbb{E}[c(x_k,u_k,w_k)] + r\left[1+\frac{\overline{m}-1}{(1+\beta^{-1})^{j-1}}\right]V^*(x_{k+1}) \\ &+ \frac{(\overline{m}-1)\beta^{j-1}}{(1+\beta)^j}\Big\{\beta\mathbb{E}[c(x_k,u_k,w_k)] - rV^*(x_{k+1})\Big\} \\ &= \left[1+\frac{(\overline{m}-1)\beta^j}{(1+\beta)^j}\right]\mathbb{E}[c(x_k,u_k,w_k)] \\ &+ r\left[1+\frac{(\overline{m}-1)}{(1+\beta^{-1})^{j-1}} - \frac{(\overline{m}-1)\beta^{j-1}}{(1+\beta)^j}\right]V^*(x_{k+1}) \\ &= \left[1+\frac{\overline{m}-1}{(1+\beta^{-1})^j}\right]\Big[\mathbb{E}[c(x_k,u_k,w_k)] + rV^*(x_{k+1})\Big] \\ &= \left[1+\frac{\overline{m}-1}{(1+\beta^{-1})^j}\right]V^*(x_k). \end{aligned}$$

The right hand of the inequality (12) is proved.

As for the left hand of (12), let $i = 1$,

$$\begin{aligned} V_1(x_k) &= \mathbb{E}[c(x_k, u_k, w_k)] + rV_0(x_{k+1}) \\ &\geq \mathbb{E}[c(x_k, u_k, w_k)] + r\underline{m}V^*(x_{k+1}) \\ &\geq \mathbb{E}[c(x_k, u_k, w_k)] + r\underline{m}V^*(x_{k+1}) \\ &+ \frac{\underline{m}-1}{1+\beta}\Big\{\beta\mathbb{E}[c(x_k, u_k, w_k)] - rV^*(x_{k+1})\Big\} \\ &= \frac{1+\underline{m}\beta}{1+\beta}\Big[\mathbb{E}[c(x_k, u_k, w_k)] + rV^*(x_{k+1})\Big] \\ &= \big[1 + \frac{\underline{m}-1}{1+\beta^{-1}}\big]V^*(x_k). \end{aligned}$$

Suppose for $i = j-1$, $j = 1, 2, 3, ...$, the conclusion still holds, i.e.

$$V_{j-1}(x_k) \geq \left[1 + \frac{\underline{m}-1}{(1+\beta^{-1})^{j-1}}\right] V^*(x_k).$$

Then, for $i = j$, one has

$$\begin{aligned} V_j(x_k) &= \mathbb{E}[c(x_k, u_k, w_k)] + rV_{j-1}(x_{k+1}) \\ &\geq \mathbb{E}[c(x_k, u_k, w_k)] + r\left[1 + \frac{\underline{m}-1}{(1+\beta^{-1})^{j-1}}\right] V^*(x_{k+1}) \\ &\geq \mathbb{E}[c(x_k, u_k, w_k)] + r\left[1 + \frac{\underline{m}-1}{(1+\beta^{-1})^{j-1}}\right] V^*(x_{k+1}) \\ &+ \frac{(\underline{m}-1)\beta^{j-1}}{(1+\beta)^j}\Big\{\beta\mathbb{E}[c(x_k, u_k, w_k)] - rV^*(x_{k+1})\Big\} \\ &= \left[1 + \frac{(\underline{m}-1)\beta^j}{(1+\beta)^j}\right] \mathbb{E}[c(x_k, u_k, w_k)] \\ &+ r\left[1 + \frac{(\underline{m}-1)}{(1+\beta^{-1})^{j-1}} - \frac{(\underline{m}-1)\beta^{j-1}}{(1+\beta)^j}\right] V^*(x_{k+1}) \\ &= \left[1 + \frac{\underline{m}-1}{(1+\beta^{-1})^j}\right] \Big[\mathbb{E}[c(x_k, u_k, w_k)] + rV^*(x_{k+1})\Big] \\ &= \left[1 + \frac{\underline{m}-1}{(1+\beta^{-1})^j}\right] V^*(x_k). \end{aligned}$$

The left hand of the inequality (12) is proved.

By taking the limit of inequality (12), one has

$$\begin{aligned} \lim_{i\to\infty}\left\{\left[1 + \frac{\underline{m}-1}{(1+\beta^{-1})^i}\right] V^*(x_k)\right\} &= V^*(x_k), \\ \lim_{i\to\infty}\left\{\left[1 + \frac{\overline{m}-1}{(1+\beta^{-1})^i}\right] V^*(x_k)\right\} &= V^*(x_k). \end{aligned}$$

Therefore, $\lim_{i\to\infty} V_i(x_k) = V^*(x_k)$. Combining $V^*(x_k)$ and the limit of (11), one has $L_\infty = L^*, K_\infty = K^*$. This completes the proof. ■

Remark 2.4 Lemma 2.3 is proposed for stochastic situations. A similar result was provided by [22], which is used for the deterministic nonlinear systems.

3 Model-free On-policy value iteration Algorithm

Define the Q-function

$$Q(x_k,u_k,w_k)\triangleq\mathbb{E}[c(x_k,u_k,w_k)]+rV(x_{k+1}),\tag{14}$$

or more compactly as

$$Q(x_k,u_k,w_k)=\mathbb{E}(z_k^THz_k),\tag{15}$$

where $z_k\triangleq[x_k^T\ u_k^T\ w_k^T]^T$, $H=\begin{bmatrix}H_{xx}&H_{xu}&H_{xw}\\H_{ux}&H_{uu}&H_{uw}\\H_{wx}&H_{wu}&H_{ww}\end{bmatrix}\triangleq\begin{bmatrix}r(A^TPA+D^TPD)+R&r(A^TPB+D^TPE)&r(A^TPC+D^TPF)\\r(B^TPA+E^TPD)&r(B^TPB+E^TPE)+I&r(B^TPC+E^TPF)\\r(C^TPA+F^TPD)&r(C^TPB+F^TPE)&-\gamma^2+r(C^TPC+F^TPF)\end{bmatrix}$.

According to the first-order necessary condition for optimality [14], one has

$$\begin{cases}L=[(H_{uu}+H_{uu}^T)-(H_{uw}^T+H_{wu})(H_{ww}+H_{ww}^T)^{-1}(H_{wu}^T+H_{uw})]^{-1}\\\quad\times[(H_{xw}^T+H_{wx})(H_{ww}+H_{ww}^T)^{-1}(H_{uw}+H_{wu}^T)-(H_{ux}+H_{xu}^T)],\\K=[(H_{ww}+H_{ww}^T)-(H_{wu}^T+H_{uw})(H_{uu}+H_{uu}^T)^{-1}(H_{uw}^T+H_{wu})]^{-1}\\\quad\times[(H_{xu}^T+H_{ux})(H_{uu}+H_{uu}^T)^{-1}(H_{uw}^T+H_{wu})-(H_{xw}^T+H_{wx})].\end{cases}\tag{16}$$

From (16), the iteration can learn the minmax gains without relying on the knowledge of system dynamics. Next, we develop a model-free value iteration method to learn the Q-function (i.e. the H matrix):

$$Q^{(i+1)}(x_k,u_k^{(i)},w_k^{(i)})=\mathbb{E}[c(x_k,u_k^{(i)},w_k^{(i)})]+rQ^{(i)}(x_{k+1},u_k^{(i)},w_k^{(i)}).\tag{17}$$

Note that there is no need to solve the Riccati equations or Lyapunov equations in each iteration. Define the following iteration, including policy evaluation

$$\mathbb{E}\big(vech\big(z_k^{(i)}(z_k^{(i)})^T\big)\big)^Th^{(i+1)}=\ \mathbb{E}\big(c(x_k,u_k^{(i)},w_k^{(i)})\big)+\Big(r\mathbb{E}\big(vech\big(z_{k+1}^{(i)}(z_{k+1}^{(i)})^T\big)^T\big)\Big)h^{(i)},\tag{18}$$

and policy improvement

$$\begin{cases}L^{(i+1)}=[(H_{uu}^{(i+1)}+H_{uu}^{(i+1)^T})-(H_{uw}^{(i+1)^T}+H_{wu}^{(i+1)})(H_{ww}^{(i+1)}+H_{ww}^{(i+1)^T})^{-1}\\\quad\times(H_{wu}^{(i+1)^T}+H_{uw}^{(i+1)})]^{-1}[(H_{xw}^{(i+1)^T}+H_{wx}^{(i+1)})(H_{ww}^{(i+1)}+H_{ww}^{(i+1)^T})^{-1}\\\quad\times(H_{uw}^{(i+1)}+H_{wu}^{(i+1)^T})-(H_{ux}^{(i+1)}+H_{xu}^{(i+1)^T})],\\K^{(i+1)}=[(H_{ww}^{(i+1)}+H_{ww}^{(i+1)^T})-(H_{wu}^{(i+1)^T}+H_{uw}^{(i+1)})(H_{uu}^{(i+1)}+H_{uu}^{(i+1)^T})^{-1}\\\quad\times(H_{uw}^{(i+1)^T}+H_{wu}^{(i+1)})]^{-1}[(H_{xu}^{(i+1)^T}+H_{ux}^{(i+1)})(H_{uu}^{(i+1)}+H_{uu}^{(i+1)^T})^{-1}\\\quad\times(H_{uw}^{(i+1)^T}+H_{wu}^{(i+1)})-(H_{xw}^{(i+1)^T}+H_{wx}^{(i+1)})].\end{cases}\tag{19}$$

Then, the iteration of policy evaluation can be rewritten as

$$\mathbb{E}(\Phi^{(i)})^Th^{(i+1)}=\mathbb{E}(Y^{(i)})+\big(r\mathbb{E}(\widetilde{\Phi}^{(i)})^T\big)h^{(i)},$$

where

$$\begin{cases}\phi_k^{(i)}=vech(z_k^{(i)}(z_k^{(i)})^T),\\\Phi^{(i)}=[\phi_0^{(i)},\phi_1^{(i)},...,\phi_N^{(i)}],\\\widetilde{\Phi}^{(i)}=[\phi_1^{(i)},\phi_1^{(i)},...,\phi_{N+1}^{(i)}],\\Y^{(i)}=[c(x_0,u_0^{(i)},w_0^{(i)}),c(x_1,u_1^{(i)},w_1^{(i)}),...,c(x_N,u_N^{(i)},w_N^{(i)})].\end{cases}$$

Using the least squares method, we have the estimate of $h^{(i)}$

$$h^{(i+1)} = \left(\mathbb{E}(\Phi^{(i)})\mathbb{E}(\Phi^{(i)})^T\right)^{-1}\mathbb{E}(\Phi^{(i)})\left[\mathbb{E}(Y^{(i)}) + \left(r\mathbb{E}(\widetilde{\Phi}^{(i)})^T\right)h^{(i)}\right]. \tag{20}$$

The model-free Q-learning value iteration algorithm is concluded in Algorithm 2.

Algorithm 2: On-policy model-free Q-learning value iteration

Input: Discount factor r, least squares data volume N, x_0, k_{end}, ε.

Output: The estimation of minimax gain $\widehat{L}, \widehat{K}$.

initialization: $i = 0$, L_0, K_0, $h_0 = v(H_0) = 0$,
for $k = 1 : k_{end}$ **do**
  Collect sample points for batch i (N sample points per batch).
  Policy evaluation:
    Estimate $h^{(i)}$ according to (20).
    $H^{(i)} = f(h^{(i)})$.
  Policy improvement:
    Use $H^{(i)}$ update $L^{(i+1)}$, $K^{(i+1)}$ as (19).
  if $||L^{(i+1)} - L^{(i)}|| < \varepsilon$ *and* $||K^{(i+1)} - K^{(i)}|| < \varepsilon$, **then**
    Break
  else
    $i = i + 1$
  end
end
$\widehat{L} = L^{(i+1)}$, $\widehat{K} = K^{(i+1)}$

The equivalence between model-free on-policy value iteration algorithm and model-based algorithm is given in the following theorem.

Theorem 3.1 *Assume that there exists a positive definite solution to the ARE (6). Then, the model-free on-policy value iteration algorithm (Algorithm 2) and the model-based algorithm (Algorithm 1) are equivalent.*

Proof Combining (4), (5), (14), and (15), we can obtain

$$\mathbb{E}(z_k^T H z_k) = \mathbb{E}(x_k^T P x_k), \tag{21}$$

By vectorization, (21) becomes

$$\mathbb{E}[vech(x_k x_k^T)]^T vecs(\widetilde{K}^T H \widetilde{K}) = \mathbb{E}[vech(x_k x_k^T)]^T vecs(P),$$

which can be written as

$$\widetilde{K}^T H \widetilde{K} = P, \tag{22}$$

where $vecs(X) = [x_{11}, 2x_{12}, ..., 2x_{1n}, x_{22}, 2x_{23}, ..., 2x_{n-1,n}, x_{n,n}]^T$, $\widetilde{K} \triangleq [I\ L^T\ K^T]^T$.

Taking (17), (5), (14), and (22) into consideration, we have the following iteration equaiton

$$\mathbb{E}((z_k^{(i)})^T H^{(i+1)} z_k^{(i)}) = \mathbb{E}[c(x_k, u_k^{(i)}, w_k^{(i)})] + r\mathbb{E}((z_{k+1}^{(i)})^T H^{(i)} z_{k+1}^{(i)}). \tag{23}$$

From (23), we have (18). Thus, the iterative equations (18)-(19) and (10)-(11) are equivalent. The proof of the theorem is now completed. ■

Remark 3.2 Notice that in Theorem 3.1, the equivalence of the model-free on-policy value iteration and the model-based value iteration is shown. By Lemma 2.3, we can obtain the convergence of model-free on-policy value iteration algorithm.

4 Model-free Off-policy value iteration Algorithm

For the off-policy situation, the Q-function is defined as

$$Q(x_k, u^{off}, w^{off}) = \mathbb{E}[c(x_k, u^{off}, w^{off})] + rV(x_{k+1}), \tag{24}$$

where u^{off}, w^{off} are the arbitrary behavior policies. When the policies are optimal, we have the optimal Q-function

$$Q^*(x_k, u^{off}, w^{off}) \triangleq Q_{(u^*, w^*)}(x_k, u^{off}, w^{off}),$$

which can be written as

$$\begin{aligned} Q^{(i)}(x_k, u^{off}, w^{off}) &= \mathbb{E}[c(x_k, u^{off}, w^{off})] + rV^{(i)}(x_{k+1}) \\ &= \mathbb{E}[c(x_k, u^{off}, w^{off})] + r\mathbb{E}(x_{k+1}^T P^{(i)} x_{k+1}) \\ &= \mathbb{E}(z_k^T H^{(i)} z_k), \end{aligned} \tag{25}$$

where $z_k \triangleq [x_k^T \ {u^{off}}^T \ {w^{off}}^T]^T$. Additionally, the value function is equal to the Q-function [9], we get

$$\mathbb{E}(x^T P^{(i)} x) = \mathbb{E}\Big(\begin{bmatrix} x \\ u^{(i)}(x) \\ w^{(i)}(x) \end{bmatrix}^T H^{(i)} \begin{bmatrix} x \\ u^{(i)}(x) \\ w^{(i)}(x) \end{bmatrix} \Big),$$

i.e.

$$P^{(i)} = \Big(\begin{bmatrix} I \\ L^{(i)} \\ K^{(i)} \end{bmatrix}^T H^{(i)} \begin{bmatrix} I \\ L^{(i)} \\ K^{(i)} \end{bmatrix} \Big). \tag{26}$$

Combining (17), (5), (24), and (26), one has

$$\mathbb{E}(z_k^T H^{(i+1)} z_k) = \mathbb{E}[c(x_k, u^{off}, w^{off})] + r\mathbb{E}\Big(\begin{bmatrix} x_{k+1} \\ L^{(i)} x_{k+1} \\ K^{(i)} x_{k+1} \end{bmatrix}^T H^{(i)} \begin{bmatrix} x_{k+1} \\ L^{(i)} x_{k+1} \\ K^{(i)} x_{k+1} \end{bmatrix} \Big).$$

This can be written as

$$\mathbb{E}[vech(z_k z_k^T)]^T h^{(i+1)} = \mathbb{E}[c(x_k, u^{off}, w^{off})] + r\ \mathbb{E}[vech(\begin{bmatrix} x_{k+1} \\ L^{(i)}x_{k+1} \\ K^{(i)}x_{k+1} \end{bmatrix} \begin{bmatrix} x_{k+1} \\ L^{(i)}x_{k+1} \\ K^{(i)}x_{k+1} \end{bmatrix}^T)]^T h^{(i)}. \tag{27}$$

We use the arbitrary behavior policies to generate enough data points and collect them. By (27), we can obtain

$$\mathbb{E}(\Phi^{(i)})^T h^{(i+1)} = \mathbb{E}(Y^{(i)}) + \mathbb{E}(\Psi^{(i)})^T h^{(i)},$$

where

$$\begin{cases} \phi_k^{(i)} = vech(z_k z_k^T), \\ \psi_k^{(i)} = r\ vech(\begin{bmatrix} x_{k+1} \\ L^{(i)}x_{k+1} \\ K^{(i)}x_{k+1} \end{bmatrix} \begin{bmatrix} x_{k+1} \\ L^{(i)}x_{k+1} \\ K^{(i)}x_{k+1} \end{bmatrix}^T), \\ \Phi^{(i)} = [\phi_0^{(i)}, \phi_1^{(i)}, ..., \phi_N^{(i)}], \\ \Psi^{(i)} = [\psi_0^{(i)}, \psi_1^{(i)}, ..., \psi_N^{(i)}], \\ Y^{(i)} = [c(x_0, u^{off}, w^{off}), c(x_1, u^{off}, w^{off}), ..., c(x_N, u^{off}, w^{off})]. \end{cases}$$

The estimate of $h^{(i)}$ can be computed with the following least-square scheme

$$h^{(i+1)} = \left[\mathbb{E}(\Phi^{(i)})\mathbb{E}(\Phi^{(i)})^T\right]^{-1} \mathbb{E}(\Phi^{(i)}) \left[\mathbb{E}(Y^{(i)}) + \mathbb{E}(\Psi^{(i)})^T h^{(i)}\right]. \tag{28}$$

The equivalence between the model-free off-policy value iteration algorithm and the model-based algorithm is given in the following theorem.

Theorem 4.1 *Assume that there exists a positive definite solution to the ARE (6). Then, the off-policy value iteration algorithm (Algorithm 3) and the model-based algorithm (Algorithm 1) are equivalent.*

Proof Combining (27) and (25), one has

$$P^{(i+1)} = (\begin{bmatrix} I \\ L^{(i)} \\ K^{(i)} \end{bmatrix}^T H^{(i)} \begin{bmatrix} I \\ L^{(i)} \\ K^{(i)} \end{bmatrix}). \tag{29}$$

Note that (29) is equivalent to (10). Then, the iterative equations (27)-(19) and (10)-(11) are equivalent. This completes the proof. ■

Remark 4.2 In Theorem 4.1, the equivalence of the model-free off-policy value iteration and the model-based value iteration is shown. By Theorem 3.1, we establish the equivalence of the model-free off-policy value iteration and the model-based value iteration algorithms. This together with Theorem 4.1 implies the equivalence of the off-policy and on-policy model-free value iteration algorithms.

Algorithm 3: Off-policy model-free Q-learning value iteration

Input: Discount factor r, least squares data volume N, x_0, k_{end}, ε.

Output: The estimation of minimax gain $\widehat{L}, \widehat{K}$.

initialization: $i = 0$, L_0, K_0, $h_0 = v(H_0) = 0$,
for $k = 1 : k_{end}$ **do**
  Collect sample points for batch i by behavior policy u^{off} *and* w^{off} (N sample points per batch).
  Policy evaluation:
    Estimate $h^{(i)}$ according to (28).
    $H^{(i)} = f(h^{(i)})$.
  Policy improvement:
    Use $H^{(i)}$ update $L^{(i+1)}$, $K^{(i+1)}$ as (19).
  if $||L^{(i+1)} - L^{(i)}|| < \varepsilon$ *and* $||K^{(i+1)} - K^{(i)}|| < \varepsilon$, **then**
    Break
  else
    $i = i + 1$
  end
end
$\widehat{L} = L^{(i+1)}$, $\widehat{K} = K^{(i+1)}$

5 Simulation Study

To demonstrate the effectiveness of the proposed algorithms in a stochastic case, we present a simulation example that focuses on the design of the F-16 aircraft autopilot [9]. Consider a system with following parameters: $\gamma = 1, r = 0.001, R = \mathrm{diag}\,(1,1,1)\,, x_0 = [10 \quad 5 \quad -2]^T$, $B = \begin{bmatrix} -0.00150808 & -0.0096 & 0.867345 \end{bmatrix}^T$, $C = \begin{bmatrix} 0.00951892 & 0.00038373 & 0 \end{bmatrix}^T$,

$$A = \begin{bmatrix} 0.906488 & 0.0816012 & -0.0005 \\ 0.0741349 & 0.90121 & -0.000708383 \\ 0 & 0 & 0.132655 \end{bmatrix}.$$

By using Algorithm 1, after two iterations, we obtain

$$P^* = \begin{bmatrix} 1.0016609406 & 0.0002862002 & -0.0000009506 \\ 0.0002862002 & 1.0016441664 & -0.0000012947 \\ -0.0000009506 & -0.0000012947 & 1.0000351456 \end{bmatrix}.$$

By Algorithm 2, after seven iterations, we have

$$\widehat{P}^* = \begin{bmatrix} 0.9925631866 & 0.0385674270 & -0.0160499187 \\ 0.0385674270 & 0.9665930972 & 0.0305363862 \\ -0.0160499187 & 0.0305363862 & 1.00016064817 \end{bmatrix}.$$

By Algorithm 3, after seven iterations, we obtain

$$\widehat{P}^* = \begin{bmatrix} 1.2180699288 & -0.1239634402 & 0.0124082424 \\ -0.1239634402 & 1.0695248090 & -0.0080062414 \\ 0.0124082424 & -0.0080062414 & 1.0002895411 \end{bmatrix}.$$

These indicate that the proposed algorithms in this paper perform well.

The iterative result of the on-policy value iteration algorithm (Algorithm 2) is demonstrated in Figures 1-2, including the state trajectories and the iterative curve of P_i. Meanwhile, the off-policy value iteration algorithm (Algorithm 3) is demonstrated in Figures 3-4. These figures show that both Algorithms 2 and 3 are convergent.

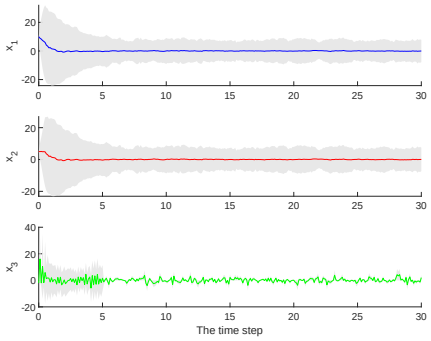

Figure 1: State trajectories (Algorithm 2).

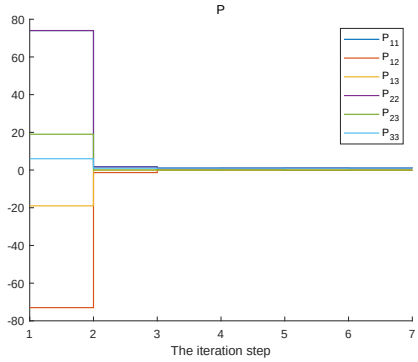

Figure 2: The iterative curve of P_i (Algorithm 2).

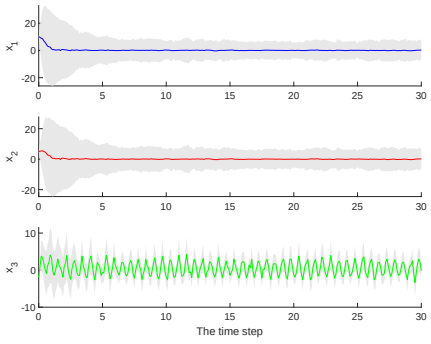

Figure 3: State trajectories (Algorithm 3).

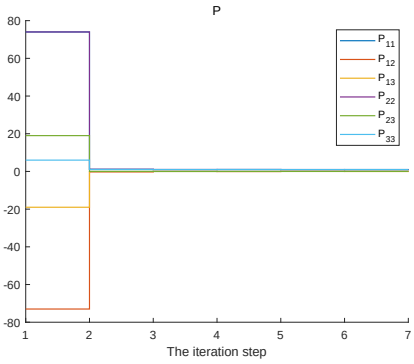

Figure 4: The iterative curve of P_i (Algorithm 3).

6 Conclusion

In this paper, two model-free value iteration algorithms have been developed to solve the discrete-time infinite-horizon stochastic zero-sum linear quadratic games. The convergence of the algorithms is provided. By the model-free algorithms, two saddle-point policies are obtained by using interactive data without system parameters. A numerical example is provided to demonstrate the performance of proposed algorithms.

References

[1] Pachter M, Pham K D, Discrete-time linear-quadratic dynamic games, *Journal of Optimization Theory and Applications*, 2010, **146**: 151–179.

[2] Rizvi S A A, Lin Z, Output feedback Q-learning for discrete-time linear zero-sum games with application to the H-infinity control, *Automatica*, 2018, **95**: 213–221.

[3] Kiumarsi B, Vamvoudakis K G, Modares H, Lewis F L, Optimal and autonomous control using reinforcement learning: A survey, *IEEE transactions on neural networks and learning systems*, 2017, **29**(6): 2042–2062.

[4] Lewis F L, Vamvoudakis K G, Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2010, **41**(1): 14–25.

[5] Werbos P, Approximate dynamic programming for real-time control and neural modeling, *Handbook of intelligent control*, 1992.

[6] Barto A G, Bradtke S J, Singh S P, Learning to act using real-time dynamic programming, *Artificial intelligence*, 1995, **72**(1 - 2): 81–138.

[7] Liu D, Wei Q, Wang D, Yang X, Li H, *Adaptive dynamic programming with applications in optimal control*, Springer, 2017.

[8] Lewis F L, Liu D, *Reinforcement learning and approximate dynamic programming for feedback control*, John Wiley & Sons, 2013.

[9] 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.

[10] Heydari A, Stability analysis of optimal adaptive control under value iteration using a stabilizing initial policy, *IEEE transactions on neural networks and learning systems*, 2017, **29**(9): 4522–4527.

[11] Al-Tamimi A, Lewis F L, Abu-Khalaf M, 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.

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

[13] 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.

[14] Lai J, Xiong J, Shu Z, Model-free optimal control of discrete-time systems with additive and multiplicative noises, *Automatica*, 2023, **147**: 110685.

[15] Zhang K, Yang Z, Basar T, Policy optimization provably converges to Nash equilibria in zero-sum linear quadratic games, *Advances in Neural Information Processing Systems*, 2019, **32**.

[16] Jiang Y, Jiang Z - P, Computational adaptive optimal control for continuous - time linear systems with completely unknown dynamics, *Automatica*, 2012, **48**(10): 2699–2704.

[17] Luo B, Yang Y, Liu D, Policy iteration Q-learning for data-based two-player zero-sum game of linear discrete-time systems, *IEEE Transactions on Cybernetics*, 2020, **51**(7): 3630–3640.

[18] Bian T, Jiang Y, Jiang Z - P, Adaptive dynamic programming for stochastic systems with state and control dependent noise, *IEEE Transactions on Automatic control*, 2016, **61**(12): 4170–4175.

[19] Bian T, Jiang Z - P, Continuous - Time Robust Dynamic Programming, *SIAM Journal on Control and Optimization*, 2019, **57**(6): 4150–4174.

[20] Li N, Li X, Peng J, Xu Z Q, Stochastic linear quadratic optimal control problem: a reinforcement learning method, *IEEE Transactions on Automatic Control*, 2022, **67**(9): 5009–5016.

[21] Chen S, Necessary and sufficient conditions for the existence of positive solutions to algebraic

Riccati equations with indefinite quadratic term, *Applied Mathematics and Optimization*, 1992, **26**: 95–110.

[22] Liu D, Li H, Wang D, Neural-network-based zero-sum game for discrete-time nonlinear systems via iterative adaptive dynamic programming algorithm, *Neurocomputing*, 2013, **110**: 92–100.