Relaxed Policy Iteration Algorithm for Nonlinear Zero-Sum Games With Application to H-Infinity Control

Jie Li, Shengbo Eben Li, Jingliang Duan, Yao Lyu, Wenjun Zou, Yang Guan, and Yuming Yin

***Abstract*—Though policy evaluation error profoundly affects the direction of policy optimization and the convergence property, it is usually ignored in policy iteration methods. This work incorporates the practical inexact policy evaluation into a simultaneous policy update paradigm to reach the Nash equilibrium of the nonlinear zero-sum games. In the proposed algorithm, the restriction of precise policy evaluation is removed by bounded evaluation error characterized by Hamiltonian without sacrificing convergence guarantees. By exploiting Fréchet differential, the practical iterative process of value function with estimation error is converted into the Newton's method with variable steps, which are inversely proportional to evaluation errors. Accordingly, we construct a monotone scalar sequence that shares the same Newton's method with the value sequence to bound the error of the value function, which enjoys an exponential convergence rate. Numerical results show its convergence in affine systems, and the potential to cope with general nonlinear plants.**

***Index Terms*—Hamilton–Jacobi–Isaacs (HJI) equation, Newton's method, policy iteration, zero-sum game.**

I. Introduction

Zero-sum games have attracted much attention in the field of control in the past few years. The intention of a zero-sum game is to solve Nash equilibria, at which each player loses what the other gains in its performance [1]. H_∞ control problems are typical zero-sum game problems from the view of minimax optimization, where external disturbances can be viewed as opponent players [1], [2]. The key to coping with nonlinear dynamics is to find the solution to the Hamilton–Jacobi–Isaacs (HJI) equation [3], [4], [5], which is a nonlinear partial differential equation that is hard to solve analytically.

Approximate dynamic programming (ADP) is a class of evolving computational methods that learn control policies via interacting with the environment [6]. Among all variants of ADP methods, policy iteration is the most commonly used because of its theoretical completeness [7]. It includes two alternating steps of policy evaluation and policy improvement, where the former seeks to find the value function of the current control policy, and the latter optimizes a better policy guided by the learned value function. Recently, policy iteration has been applied to solving the HJI equation of nonlinear zero-sum games or H_∞ control problems [8], [9]. Abu-Khalaf et al. [10], [11] extended policy iteration to two-player zero-sum games. A two-loop policy iteration method with uniform convergence guarantees was developed, where the disturbance policy and its matching value function were approximately updated with neural networks in the inner loop [11], and the control policy was updated in the outer loop asynchronously. For simplification, a simultaneous policy update paradigm was then developed, which involved only one iterative loop [12], [13], and employed three neural networks to implement the proposed model-free off-policy algorithm [13]. Afterward, parameter tuning algorithms with only one critic network were designed to further reduce the computational burden and eliminate approximation errors in policy networks [14], [15].

In policy evaluation, attaining an exact closed-form solution remains difficult [11], [16]. For polynomial systems, one approach is to use an auxiliary optimization problem to find suboptimal solutions through semidefinite programming. In practice, another common scheme introduces neural networks with basis functions [14] or multilayer perceptron (MLP) [17], [18] to estimate the exact solution. Gradient descent can be employed to update the weights of hidden layers, but the value function may not converge to its solution by performing finite gradient steps in the application process [19]. The least squares method can conveniently minimize the residual error of the basis function weights [11], and a large number of basis functions, whose types are not easily specified, are required to achieve minor approximation errors [12], [16]. Under the assumption that the value function is uniformly approximated, the convergence of algorithms can be obtained. The uniform convergence of asynchronous methods is analyzed by proving that the value sequence is monotonic [10, Th.3], [11, Th.1]. The simultaneous policy update algorithm is demonstrated to be equivalent to finding a fixed point in the Newton's method, where the Kantorovtich's theorem assures convergence [12, Th.1], [13, Th.5]. The learned weights are also proven to ultimately converge to a neighborhood of ideal weights by the Lyapunov theory [14], [15], where the ultimate bound is related to the reconstruction error of the approximator. However, few studies have investigated the impact of policy evaluation errors caused by the actual implementation of the algorithm on convergence rate and finite-time analysis, which are also worth studying for efficient algorithm development.

This work aims to explore whether policy iteration can tolerate policy evaluation errors, and to prove that the alternate iterations as a whole are still convergent. The main contributions of this article are as follows. 1) We first establish the equivalence between relaxed policy iteration with bounded policy evaluation error, described by the

Manuscript received 24 December 2022; revised 17 March 2023; accepted 29 March 2023. Date of publication 11 April 2023; date of current version 29 December 2023. This work was supported in part by the National Key R&D Program of China under Grant 2022YFB2502901, in part by the Tsinghua University-Toyota Joint Research Center for AI Technology of Automated Vehicle, and in part by the National Natural Science Foundation of China under Grant 52202487. Recommended by Associate Editor S. S. Saab. *(Corresponding author: Shengbo Eben Li.)*

Jie Li, Shengbo Eben Li, Yao Lyu, Wenjun Zou, Yang Guan, and Yuming Yin are with the School of Vehicle and Mobility, Tsinghua University, Beijing 100084, China (e-mail: jie-li18@mails.tsinghua.edu.cn; lisb04@gmail.com; y-lv19@mails.tsinghua.edu.cn; zouwj20@mails.tsinghua.edu.cn; guany17@mails.tsinghua.edu.cn; yinyuming@zjut.edu.cn).

Jingliang Duan is with the School of Mechanical Engineering, University of Science and Technology Beijing, Beijing 100083, China (e-mail: duanjl15@163.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TAC.2023.3266277.

Digital Object Identifier 10.1109/TAC.2023.3266277

ratio of Hamiltonian, and the Newton's method about value function, with the step size being inversely proportional to the evaluation error. This enables policy iteration to tolerate certain evaluation errors while maintaining the overall convergence. 2) Next, we construct a scalar sequence sharing the same Newton's method with the value sequence to give an upper error bound of the value function of the whole iterative process. By deriving the convergence rate of the scalar sequence, the error of value function is proven to decay exponentially with Newton iterations, i.e., an ϵ-optimal solution can be obtained with a Newton iteration complexity of $\mathcal{O}(\log(\epsilon^{-1}))$.

The rest of this article is organized as follows. In Section II, a description of the zero-sum game and its HJI equation is given. Section III describes the proposed algorithm, whose convergence mechanism is analyzed in Section IV. Simulation results are shown in Section V. Finally, Section VI concludes this article.

II. Problem Description

Consider an affine nonlinear plant with known dynamics

$$\dot{x} = f(x) + g(x)u + k(x)w \tag{1}$$

where $x \in \Omega \subseteq \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ and $w \in \mathbb{R}^q$ are control input and external disturbance. $f(x) \in \mathbb{R}^n$, $g(x) \in \mathbb{R}^{n\times m}$, and $k(x) \in \mathbb{R}^{n\times q}$ are known continuous nonlinear functions defined in a state set Ω containing the origin.

Define the performance index as

$$J(u,w) \triangleq \int_t^\infty l(x,u,w)\,d\tau \tag{2}$$

where t is the initial time, $l(x,u,w) \triangleq x^TQx + u^TRu - \gamma^2 w^Tw$ is the utility function, $Q \geq 0$ and $R > 0$ are weighting matrices, and γ denotes the attenuation level. Assume that there exists an admissible control policy such that the system is asymptotically stable and has a finite performance.

For control policy $u(x)$ and disturbance policy $w(x)$, define the value function as

$$V(x) \triangleq \int_t^\infty \left(x^TQx + u^TRu - \gamma^2 w^Tw\right) d\tau. \tag{3}$$

Taking the partial derivative of t for the value function (3), one can formulate an equation about Hamiltonian

$$H\left(x,u,w,\frac{\partial V(x)}{\partial x}\right) \triangleq x^TQx + u^TRu - \gamma^2 w^Tw + \frac{\partial V(x)}{\partial x^T}\left(f(x)+g(x)u+k(x)w\right) = 0 \tag{4}$$

which is widely applied in algorithm design. The solution of the differential equation (4) is the value function of policies.

The two-player zero-sum differential game is derived as

$$V^*(x) = J(u^*,w^*) = \min_u \max_w J(u,w) \tag{5}$$

where $V^*(x)$ is the optimal value function or Nash value of the zero-sum game, and $J(u^*,w^*)$ is the optimal performance function. Upon combining Nash condition and Isaacs' condition, we will show how to solve the given zero-sum game from the perspective of partial differential equations.

Remark 1: The solvability of H_∞ suboptimal control problem $\|T_{zw}\|_\infty^2 = \sup_w \frac{\|z\|_2^2}{\|w\|_2^2} < \gamma^2$ is equivalent to that of a zero-sum game [1]. In H_∞ control problems, z denotes the objective output, and $1/\gamma$ represents the boundary of model uncertainty.

A. HJI Equation

The zero-sum differential game (5) may have no solution or multiple solutions [20]. For simplicity, we assume that there exists a unique solution. The well-known Nash condition provides a sufficient condition for the uniqueness of the solution, requiring a saddle point [21], i.e.,

$$J(u^*,w) \leq J(u^*,w^*) \leq J(u,w^*)\ \forall u,w \in L_2[t\ ,\infty) \tag{6}$$

where u^* and w^* are at equilibrium. Both players have no motivation to change to make their performance better. A necessary condition for Nash condition is Isaacs' condition [21], which can be seen as an extension of Pontryagin maximum principle, i.e.,

$$\min_u \max_w H\left(x,u,w,\frac{\partial V^*(x)}{\partial x}\right) = \max_w \min_u H\left(x,u,w,\frac{\partial V^*(x)}{\partial x}\right). \tag{7}$$

Applying two stationarity conditions $\partial H/\partial u = 0$ and $\partial H/\partial w = 0$ to the Hamiltonian $H\left(x,u,w,\frac{\partial V^*(x)}{\partial x}\right)$ gives

$$u = \arg\min_u H\left(x,u,w,\frac{\partial V^*(x)}{\partial x}\right) = -\frac{1}{2}R^{-1}g^T(x)\frac{\partial V^*(x)}{\partial x} \tag{8}$$

$$w = \arg\max_w H\left(x,u,w,\frac{\partial V^*(x)}{\partial x}\right) = \frac{1}{2\gamma^2}k^T(x)\frac{\partial V^*(x)}{\partial x}. \tag{9}$$

Substituting them into (4) yields the HJI equation

$$G(V^*) \triangleq x^TQx + \frac{\partial V^*(x)}{\partial x^T}f(x) - \frac{1}{4}\frac{\partial V^*(x)}{\partial x^T}g(x)R^{-1}g^T(x)\frac{\partial V^*(x)}{\partial x} + \frac{1}{4\gamma^2}\frac{\partial V^*(x)}{\partial x^T}k(x)k^T(x)\frac{\partial V^*(x)}{\partial x} = 0 \tag{10}$$

which is a nonlinear partial differential equation. The boundary condition is $V^*(x_e) = 0$, where x_e is an equilibrium state, usually chosen as a zero vector in regulator problems.

On the other hand, assume that the dynamics (1) is zero-state observable, and suppose that the solution $V^*(x)$ of the HJI equation (10) is smooth positive semidefinite. Then, the derived control policy and disturbance policy are at Nash equilibrium [21]. Thus, Isaacs' condition can be a sufficient condition for Nash condition under mild hypotheses. The HJI equation (10) can be rewritten as a formulation of Hamiltonian

$$\min_u \max_w H\left(x,u,w,\frac{\partial V^*(x)}{\partial x}\right) = 0. \tag{11}$$

So far, the solving process of the zero-sum game (5) has been converted into that of a partial differential equation. However, its analytical solution is difficult to find. Therefore, in the following section, we will introduce some numerical methods to solve the derived HJI equation (11).

Remark 2: Conditions for the existence of smooth positive semidefinite solution of the HJI equation (10) have been provided in [4], [5]. The results of this article are also based on regularity assumptions made in [4] and [5].

B. Existing Policy Iteration Algorithms

Policy iteration is a numerical method widely applied in ADP [6]. It involves alternating iterations between policy evaluation and policy improvement. In the policy evaluation step, the value function $V^{k+1}(x)$

is updated by solving the differential equation

$$H\left(x,u^k,w^k,\frac{\partial V^{k+1}(x)}{\partial x}\right)=l\left(x,u^k,w^k\right)+\frac{\partial V^{k+1}(x)}{\partial x^T}\left(f(x)+g(x)u^k+k(x)w^k\right)=0 \tag{12}$$

which makes policies u^k and w^k no longer greedy. In policy improvement steps, policies u^{k+1} and w^{k+1} are improved by minimizing or maximizing the abovementioned Hamiltonian

$$u^{k+1}=-\frac{1}{2}R^{-1}g^T(x)\frac{\partial V^{k+1}(x)}{\partial x} \tag{13}$$

$$w^{k+1}=\frac{1}{2\gamma^2}k^T(x)\frac{\partial V^{k+1}(x)}{\partial x} \tag{14}$$

making the value function incorrect for the updated policies.

It is a commonly employed algorithm framework to iterate the abovementioned two steps until the value function meets some termination conditions, e.g., $|V^{k+1}(x)-V^k(x)|\le\epsilon$. The simultaneous policy update paradigm [12] has been proven to have a relationship with the Newton's method.

Lemma 1 [12]: Consider a Banach space $\mathbb{V}\subset\{V(x)|V(x):\Omega\to\mathbb{R},\ V(0)=0\}$ equipped with a norm $\|\cdot\|_\Omega$. With the definition of Gâteaux derivative, the Gâteaux and Fréchet differential of $G(V)$ at V can be derived as

$$G'(V)W=\frac{\partial W}{\partial x^T}f-\frac{1}{2}\frac{\partial W}{\partial x^T}gR^{-1}g^T\frac{\partial V}{\partial x}+\frac{1}{2\gamma^2}\frac{\partial W}{\partial x^T}kk^T\frac{\partial V}{\partial x}. \tag{15}$$

Then, the iteration process (12)–(14) is equivalent to the following Newton's method

$$V^{k+1}=V^k-\left[G'\left(V^k\right)\right]^{-1}G\left(V^k\right). \tag{16}$$

The transformation of the Newton's method is valid only if the value function is uniformly approximated in policy evaluation [12, Th.1]. Besides, from the global point of view, policy evaluation errors bring mismatch and confusion to the iterative process. Not only is the updated value function incorrect for both policies, but policies are no longer optimal. Nevertheless, few studies have explored the effect of evaluation errors caused by calculation in practice, such as finite gradient steps, on algorithm convergence. In the following section, the evaluation error will be directly incorporated into the algorithm design.

III. Relaxed Policy Iteration Algorithm

In this section, we will introduce relaxed policy iteration, which can tolerate evaluation errors while maintaining the overall convergence. First, an easy-to-implement termination condition of policy evaluation is developed. Then, the intuitive understanding of the proposed algorithm will be presented.

Neural networks with polynomial or MLP can be used to approximate the value function $V(x;\omega)$, denoted as the critic network with weight ω. Given the improved policies u^k and w^k, the value function $V(x;\omega^{k+1})$ will be updated based on the approximated Hamiltonian

$$H\left(x,u^k,w^k,\frac{\partial V\left(x;\omega^{k+1}\right)}{\partial x}\right)=l\left(x,u^k,w^k\right)+\frac{\partial V\left(x;\omega^{k+1}\right)}{\partial x^T}\left(f(x)+g(x)u^k+k(x)w^k\right). \tag{17}$$

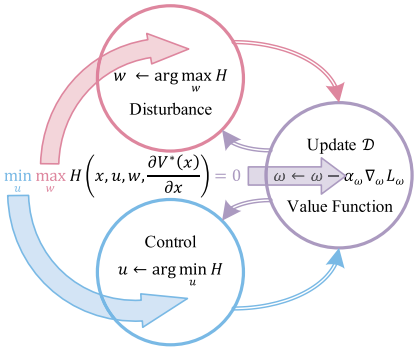


Fig. 1. Relaxed policy iteration algorithm.

In the extreme case, when there is no change in the value function $V(x;\omega^{k+1})$, the Hamiltonian (17) remains the same. Ideally, the updated value function makes the Hamiltonian equal to zero. Actually, due to finite computation and limited approximation abilities of neural networks, the parameterized value function has residual errors, making it difficult for the Hamiltonian to be precisely zero. Between extreme and ideal cases, the ratio of Hamiltonian provides a practical relaxation and quantitative assessment of accuracy for policy evaluation

$$\left|H\left(x,u^k,w^k,\frac{\partial V\left(x;\omega^{k+1}\right)}{\partial x}\right)\right|\le\left(1-\alpha^k\right)\left|H\left(x,u^k,w^k,\frac{\partial V\left(x;\omega^k\right)}{\partial x}\right)\right| \tag{18}$$

where $\alpha^k\in(0,\ 1]$ characterizes the degree of relaxation. Note that α^k is strictly greater than 0, which avoids the above extreme case of invariant Hamiltonian and stagnant learning. When α^k gradually approaches 1 from the negative direction, the right-hand side of the abovementioned inequality will tend to 0, and the policy will be evaluated more and more accurately. Therefore, the update of the value function can tolerate certain errors. In the theoretical analysis of the following section, we will see α^k exactly controls the step size of each Newton iteration.

During implementation, policy evaluation is performed by repeatedly applying gradient descent, such as Adam method [19], to an over-parameterized critic network to find such a value function $V(x;\omega^{k+1})$. In order to satisfy the persistency of excitation condition [22], parallel agents are employed to explore different parts of state space to improve the diversity of the training data. Besides, policy improvement directly applies the closed-form expressions shown in (13) and (14) to the affine system. Note that the algorithm can be generalized to other cost functions than quadratic functions as long as an analytical solution to the optimal policy is available. The pseudocode of the offline algorithm is shown in Algorithm 1, and the corresponding procedure is presented in Fig. 1.

Remark 3: Excessive relaxation leads to slow convergence of the algorithm, especially for linear systems, since existing least squares methods can yield relatively accurate quadratic value functions. On the other hand, deficient relaxation requires extra gradient steps to improve the accuracy of policy evaluation and introduce computational burden. Therefore, the degree of relaxation balances the rate of convergence and the total gradient steps applied in the policy evaluation process.

Remark 4: Initial admissible control policy is also required. It can be guaranteed if the Hamiltonian of the initial value function, found by

Algorithm 1: Relaxed Policy Iteration (RPI).

Initialization: Initial value function $V(x;\omega^0)$.
Parameters: Learning rate of value function is λ_ω, step size of Newton's method is α^k, stopping criterion is $\epsilon > 0$.

0. Generate data by applying control policy u^k and disturbance policy w^k to the dynamic system (1).
1. Update the value function via gradient descent $\omega \leftarrow \omega - \lambda_\omega \frac{\partial \left| H\left(x,u^k,w^k,\frac{\partial V(x;\omega)}{\partial x}\right)\right|}{\partial \omega}$, and output $V(x;\omega^{k+1})$ until (18) holds.
2. Given the value function $V(x;\omega^{k+1})$, derive policies u^{k+1} and w^{k+1} by (13) and (14).
3. Set $k \leftarrow k+1$. If $H(x,u^k,w^k,V(x;\omega^k))=0$ or $|V(x;\omega^k)-V(x;\omega^{k-1})| \le \epsilon$, stop and output value function $V(x;\omega^k)$ and control policy u^k, otherwise go back to step 0 and continue.

numerical experiments or semidefinite programming based on the sum of squares decomposition, satisfies the inequality condition [16].

The proposed method is inherently tolerant of evaluation errors during practical implementation. In the next section, the convergence guarantees and the relationship between convergence speed and evaluation error will be introduced.

IV. Convergence Analysis

In this section, we will show the convergence mechanism of the proposed relaxed policy iteration algorithm. Although evaluation errors introduce mismatch in the entire algorithm, Fréchet differential analysis first shows that the overall algorithm is equivalent to a variant of the Newton's method with variable steps. Referring to the induction idea in the Kantorovich's theorem [23], a scalar iterative process is then constructed to guarantee convergence. The auxiliary iteration shares the same Newton's method with the value function, and gives a boundary to its error. Finally, the convergence speed is obtained by deriving the upper error bound. The abovementioned proving process will be presented in the following theorems. In light of existing studies, we consider the Banach space defined in Lemma 1, and leverage the mapping $G: \mathbb{V} \to \mathbb{V}$ defined in (10).

Theorem 1: Let $V^{k+1}(x)$ satisfy (18) in Algorithm 1. Then, the iteration process can be transformed into the Newton's method with mixed step sizes α^k and $2-\alpha^k$.

Proof: Based on the Fréchet differential derived in (15), the updated Hamiltonian in the termination condition (18) is

$$H^{k+1} \triangleq G\left(V^k\right) - G'\left(V^k\right)V^k + G'\left(V^k\right)V^{k+1}.$$

When the learning rate of gradient descent method is relatively small, the updated value function $V^{k+1}(x)$ satisfies

$$\left|G\left(V^k\right) - G'\left(V^k\right)V^k + G'\left(V^k\right)V^{k+1}\right| = \left(1-\alpha^k\right)\left|G\left(V^k\right)\right|$$

where the variable step size $\alpha^k \in (0,\ 1]$. The updating process of the value function can be denoted as the following two cases depending on whether the sign of Hamiltonian changes or not

$$V^{k+1} = \begin{cases} T\left(V^k\right) \triangleq V^k - \alpha^k\left[G'\left(V^k\right)\right]^{-1}G\left(V^k\right), & H^{k+1}G\left(V^k\right) \ge 0 \\ V^k - \left(2-\alpha^k\right)\left[G'\left(V^k\right)\right]^{-1}G\left(V^k\right), & H^{k+1}G\left(V^k\right) < 0 \end{cases} \tag{19}$$

where the mixed step sizes $\alpha^k \in (0,\ 1]$ and $2-\alpha^k \in [1,\ 2)$, and $T(\cdot)$ can be regarded as the operator of the Newton's method with variable steps. ■

The auxiliary iterative process introduced in the following derivation with mixed step sizes α^k and $2-\alpha^k$ can be employed to characterize the convergence property of the transformed Newton's method (19). The convergence of the auxiliary iteration can be obtained by using the symmetric point found by the step size $\alpha^k = 2$ to prove that the range of oscillation shrinks. The iterative process with the step size α^k provides an intuitive and practical approximation for the convergence rate of the iterative process with mixed step size. Prior to analyzing the convergence of Newton's method with variable steps α^k, we have the following lemmas.

Lemma 2: Consider a modified Newton's method

$$V^{k+1} = S\left(V^k\right) \triangleq V^k - \alpha\Gamma^0 G\left(V^k\right) = V^k - \alpha\left[G'\left(V^0\right)\right]^{-1}G\left(V^k\right)$$

equipped with a specific policy iteration whose termination condition (18) is fixed, where $\alpha \in (0,\ 1]$. Assume that the related operator $S \in \mathrm{C}^1(\Omega_0)$, where $\Omega_0 = \{V \| \|V - V^0\| < r\}$. Create an auxiliary scalar iterative process $\phi \in \mathrm{C}^1[t^0,\ t']$

$$t^{k+1} = \phi\left(t^k\right) \triangleq t^k + \alpha c^0 \psi\left(t^k\right) = t^k - \alpha\psi\left(t^k\right)/\psi'\left(t^0\right)$$

to find the root $t^* \in [t^0,\ t']$ of the equation $\psi(t^*) = 0$, where $t' = t^0 + r$. Suppose the following conditions are satisfied:

1) $\Gamma^0 = [G'(V^0)]^{-1}$ is a continuous linear operator;
2) $c^0 = -\frac{1}{\psi'(t^0)} > 0$;
3) $\|\Gamma^0 G(V^0)\| \le c^0\psi(t^0)$;
4) $\|\Gamma^0 G''(V)\| \le c^0\psi''(t)$, if $\|V - V^0\| \le t - t^0 \le r$.

Then, $\|V^{k+1} - V^k\| \le t^{k+1} - t^k$, and the value sequence generated by S has a limit V^* that satisfies $V^* = S(V^*)$.

Proof: According to condition 3)

$$0 \le \|V^1 - V^0\| = \|-\alpha\Gamma^0 G\left(V^0\right)\| \le \alpha c^0\psi\left(t^0\right) = t^1 - t^0.$$

For the operator S, $S'(V) = I - \alpha\Gamma^0 G'(V)$, $S''(V) = -\alpha\Gamma^0 G''(V)$. For the real function ϕ, $\phi'(t) = 1 + \alpha c^0\psi'(t)$. Based on condition 4), when $\|V - V^0\| \le t - t^0 \le r$

$$\begin{aligned} 0 \le \|S'(V)\| &\le \|S'(V) - S'\left(V^0\right)\| + \|S'\left(V^0\right)\| \\ &= \left\|\int_{V^0}^{V} -\alpha\Gamma^0 G''(w)\,dw\right\| + \|I - \alpha\left[G'\left(V^0\right)\right]^{-1}G'\left(V^0\right)\| \\ &\le \int_{t^0}^{t} \alpha c^0\psi''(\tau)\,d\tau + \|(1-\alpha)I\| = \phi'(t). \end{aligned}$$

Note that $\phi'(t) \ge 0$, for $t^{k-1} \le t^k$, we have $t^k = \phi(t^{k-1}) \le \phi(t^k) = t^{k+1}$. So, the sequence $\{t^k\}$ generated by $t^{k+1} = \phi(t^k)$ is monotone. For $t^0 \le t^*$, $t^1 = \phi(t^0) \le \phi(t^*) = t^*$. By induction, $t^k \le t^*$ holds for all $k \in \mathbb{N}$. Thus, $\lim_{k\to\infty} t^k = t^*$, and t^* is the smallest root of $t = \phi(t)$ in $[t^0,\ t']$.

Based on $\|V^1 - V^0\| \le t^1 - t^0$, we know that $V^1 \in \Omega_0$. Suppose that we have verified that $V^k \in \Omega_0$ and $\|V^k - V^0\| \le t^k - t^0$, $k = 1,\ \ldots,\ n$, for the corresponding points in $[V^{n-1},\ V^n]$ and $[t^{n-1},\ t^n]$, *i.e.*, $V = V^{n-1} + \lambda(V^n - V^{n-1})$, $t = t^{n-1} + \lambda(t^n - t^{n-1})$, it can be derived that $\|V - V^0\| \le t - t^0$. Therefore, $\|S'(V)\| \le \phi'(t)$. For

$k = n+1$, we have

$$\|V^{n+1}-V^n\| = \|S(V^n)-S(V^{n-1})\| = \left\|\int_{V^{n-1}}^{V^n} S'(w)\,dw\right\|$$
$$\leq \int_{t^{n-1}}^{t^n} \phi'(\tau)\,d\tau = \phi(t^n)-\phi(t^{n-1}).$$

Moreover, $V^{n+1}\in\Omega_0$ since $\|V^{n+1}-V^0\|\leq t^{n+1}-t^0\leq r$.

Hence, $\forall k\in\mathbb{N}$, $V^k\in\Omega_0$ and $\|V^{k+1}-V^k\|\leq t^{k+1}-t^k$. It can be proven that $\{V^k\}$ generated by $V^{k+1}=S(V^k)$ is a Cauchy sequence and has a limit V^* satisfying $V^*=S(V^*)$. ■

Lemma 3: The value function V^1 and parallel scalar t^1 are derived by the Newton's methods S and ϕ, where the step size is denoted as α^0. Then, $\|\Gamma^0 G(V^1)\|\leq c^0\psi(t^1)$.

Proof: According to Taylor's expansion, we have

$$\Gamma^0 G(V^1) = (1-\alpha^0)\Gamma^0 G(V^0)+\int_{V^0}^{V^1}\Gamma^0 G''(V)(V^1-V)\,dV$$
$$c^0\psi(t^1) = (1-\alpha^0)\,c^0\psi(t^0)+\int_{t^0}^{t^1} c^0\psi''(t)(t^1-t)\,dt.$$

Based on the boundary of the first value function V^1, for the corresponding values in $[V^0,\ V^1]$ and scalars in $[t^0,\ t^1]$, i.e., $V=V^0+\lambda(V^1-V^0)$, $t=t^0+\lambda(t^1-t^0)$, it can be derived that $\|V-V^0\|\leq t-t^0\leq r$ and $\|V^1-V\|\leq(1-\lambda)(t^1-t^0)=t^1-t$. Therefore, conditions 3) and 4) can be employed

$$\|\Gamma^0 G(V^1)\|$$
$$\leq (1-\alpha^0)\|\Gamma^0 G(V^0)\|+\int_{V^0}^{V^1}\|\Gamma^0 G''(V)(V^1-V)\|dV$$
$$\leq (1-\alpha^0)c^0\psi(t^0)+\int_{t^0}^{t^1} c^0\psi''(t)(t^1-t)\,dt = c^0\psi(t^1)$$

which completes the proof. ■

Then, we will show the main results of theoretical analysis. The following provides proof of convergence and derivation of convergence rate for the overall iteration procedure.

Theorem 2: For the operator $T(\cdot)$ defined in (19), suppose all conditions in Lemma 2 hold. Then, the Newton's method with variable steps converges to the root of $G(V)=0$, i.e., iterative value function V^k converges to the Nash value.

Proof: Consider the auxiliary Newton's method of a scalar function ψ, whose step size $\alpha^k\in(0,\ 1]$ is variable

$$t^{k+1}=\phi(t^k)\triangleq t^k+\alpha^k c^k\psi(t^k)=t^k-\alpha^k\psi(t^k)/\psi'(t^k)$$

where $\phi'(t)=1-\alpha^k+\alpha^k\psi(t)\psi''(t)/[\psi'(t)]^2$. Linking conditions 2) and 4), we have $\phi'(t)\geq 0$.

The first step of Newton's method $T(\cdot)$ with variable steps is the same as that of the modified Newton's method $S(\cdot)$ in Lemma 2. So, the value function V^1 is bounded, i.e., $\|V^1-V^0\|\leq t^1-t^0$. It can be proven by induction that the sequence $\{t^k\}$ generated by $t^{k+1}=\phi(t^k)$ is monotone, and $\lim_{k\to\infty}t^k=t^*$, where $t^*\in[t^0,\ t']$ is the root of $t=\phi(t)$.

Now, we are ready to present by induction that all four conditions about the first value function V^1 still hold. Consider the operator $\Gamma^0 G'(V^1)$

$$\|I-\Gamma^0 G'(V^1)\| = \left\|\Gamma^0\int_{V^0}^{V^1} G''(V)\,dV\right\|\leq c^0\psi'(t^1)+1.$$

It can be calculated that $\psi'(t^1)<0$, so condition 2) can be derived by induction. According to the Banach's theorem [23], $\Gamma^0 G'(V^1)$ has an inverse $U^0=[I-(I-\Gamma^0 G'(V^1))]^{-1}=[\Gamma^0 G'(V^1)]^{-1}$, and $\|U^0\|\leq\frac{1}{-c^0\psi'(t^1)}=\frac{\psi'(t^0)}{\psi'(t^1)}=\frac{c^1}{c^0}$. Hence, there exists a continuous linear operator

$$\Gamma^1=[G'(V^1)]^{-1}=U^0\Gamma^0$$

and according to Lemma 3

$$\|\Gamma^1 G(V^1)\|\leq\|U^0\|\|\Gamma^0 G(V^1)\|\leq c^1\psi(t^1).$$

It means that conditions 1) and 3) can be derived by induction. Note that if $\|V-V^1\|\leq t-t^1$, then $\|V-V^0\|\leq\|V-V^1\|+\|V^1-V^0\|\leq t-t^1+t^1-t^0\leq t-t^0$. Therefore, condition 4) can be proven to hold:

$$\|\Gamma^1 G''(V)\|\leq\|U^0\|\|\Gamma^0 G''(V)\|\leq c^1\psi''(t).$$

Through mathematical induction $\forall k\in\mathbb{N}$, we have

$$\|V^{k+1}-V^k\|\leq t^{k+1}-t^k.$$

Note that the sequence $\{t^k\}$ is increasing and bounded, so

$$\|V^k-V^*\|\leq t^*-t^k \tag{20}$$

and the value function sequence $\{V^k\}$ also has a limit $V^*=\lim_{k\to\infty}V^k$, which is a root of $G(V^*)=0$. Therefore, the iterative value function converges to the Nash value. ■

Theorem 3: Consider a quadratic function in $[t^0,\ t']$

$$\psi(t)\triangleq Kt^2-2t+2\eta$$

where $t^0=0$, $t'=2\eta$, $t^*=(1-\sqrt{1-2K\eta})/K$ is the root of the equation $\psi(t)=0$, and $2K\eta<1$. Assume that all the conditions in Theorem 2 hold. Then, the convergence rate of the value function of the whole iterative process is as follows:

$$\|V^k-V^*\|<\prod_{i=0}^{k-1}\left[2-\alpha^i\left(1+\sqrt{1-2K\eta}\right)\right]\frac{1-\sqrt{1-2K\eta}}{2^kK}.$$

Proof: From the quadratic function, it can be derived that

$$\psi'(t)=2Kt-2,\ \psi''(t)=2K,\psi(t^0)=2\eta,\ \psi'(t^0)=-2$$
$$\psi(t^*)=0,\ t^*=\phi(t^*).$$

The scalar Newton's method ϕ in Theorem 2 is expressed as

$$t^{k+1}=\phi(t^k)=\left(1-\frac{\alpha^k}{2}\right)t^k+\frac{\alpha^k}{2K}-\alpha^k\frac{-\frac{1}{K}+2\eta}{2Kt^k-2}.$$

Thus, the error of t^{k+1} is constrained by that of t^k

$$t^*-t^{k+1}=\left[\left(1-\frac{\alpha^k}{2}\right)-\frac{\alpha^k(1-2K\eta)}{2(1-Kt^*)(1-Kt^k)}\right](t^*-t^k)$$
$$<\frac{2-\alpha^k\left(1+\sqrt{1-2K\eta}\right)}{2}(t^*-t^k).$$

Therefore, the convergence rate of the scalar sequence is

$$t^*-t^k<\prod_{i=0}^{k-1}\frac{2-\alpha^i\left(1+\sqrt{1-2K\eta}\right)}{2}(t^*-t^0)$$
$$=\prod_{i=0}^{k-1}\left[2-\alpha^i\left(1+\sqrt{1-2K\eta}\right)\right]\frac{1-\sqrt{1-2K\eta}}{2^kK}.$$

Substituting the abovementioned inequality into (20), we can obtain the convergence speed of the value function. Suppose variable step size

α^k is constant, i.e., $\alpha^k \equiv \alpha$. Given an ϵ-optimal solution $\|V^k - V^*\| < \epsilon$, the required Newton iteration

$$k > \mathcal{O}\left(\log_{1-\alpha}\epsilon\right) = \frac{\mathcal{O}\left(\log\left(\epsilon^{-1}\right)\right)}{\log\left(\left(1-\alpha\right)^{-1}\right)} = \mathcal{O}\left(\log\left(\epsilon^{-1}\right)\right)$$

where the constant term $1/\log((1-\alpha)^{-1})$ is positively related to the policy evaluation error. ■

Remark 5: Hyperparameter α^k can be employed to adjust the convergence rate. Numerical results show that the termination condition (18) can be easily satisfied if α^k is set to a small positive number. However, this will slow down convergence compared to accurate policy evaluation, i.e., $\alpha^k \equiv 1$. In the latter case, the iteration process is equivalent to the traditional policy iteration algorithm, whose convergence rate is a particular case of our results when applying a stricter inequality zoom to iteration increments [12], [23].

Remark 6: The practical implementation of Algorithm 1 is only limited to affine systems because the optimal policy has no analytical solution in general nonlinear systems. In order to exhibit the potential of relaxation thoughts in general nonlinear systems, we try to propose a ternary policy iteration (TPI) algorithm without theoretical convergence guarantees. The policy improvement step is also relaxed by performing single or multiple gradient descent steps:

$$\theta \leftarrow \theta - \lambda_\theta \nabla_\theta L_\theta\left(\omega^{k+1}, \theta^k, \eta^k\right)$$
$$\eta \leftarrow \eta - \lambda_\eta \nabla_\eta L_\eta\left(\omega^{k+1}, \theta^k, \eta^k\right)$$

where control policy $u(x;\theta)$ and disturbance policy $w(x;\eta)$ are approximated via neural networks, θ and η are parameters to be learned. Their loss functions are as follows:

$$L_\theta\left(\omega^{k+1}, \theta^k, \eta^k\right) \triangleq \mathbb{E}_{x\in\mathcal{D}}\left[H\left(x, \theta^k, \eta^k, \omega^{k+1}\right)\right]$$
$$L_\eta\left(\omega^{k+1}, \theta^k, \eta^k\right) \triangleq \mathbb{E}_{x\in\mathcal{D}}\left[-H\left(x, \theta^k, \eta^k, \omega^{k+1}\right)\right]$$

where the Hamiltonian is approximated via ternary parameters

$$H\left(x,\theta,\eta,\omega\right) \triangleq l\left(x, u\left(x;\theta\right), w\left(x;\eta\right)\right) + \frac{\partial V\left(x;\omega\right)}{\partial x^T}\left(f\left(x\right)+g\left(x\right)u\left(x;\theta\right)+k\left(x\right)w\left(x;\eta\right)\right).$$

V. Simulation Results

This section first studies an affine nonlinear model with an analytical solution of the corresponding HJI equation. Polynomial bases and MLP are selected to validate the convergence accuracy achieved by Algorithm 1. Then, a model-free relaxed policy iteration algorithm with system identification embedded in policy training is applied to an unknown affine nonlinear system to compare the control effects of different algorithms. Finally, the TPI algorithm is implemented in a general nonlinear system to show its effectiveness and potential.

A. Oscillator Model

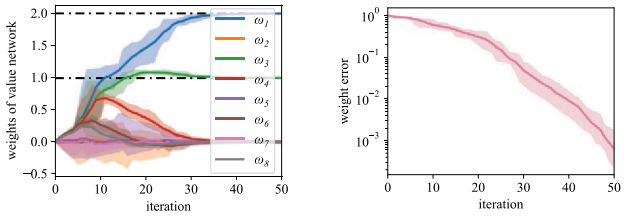
Consider an oscillator model mentioned in [12], where

$$f(x) = \begin{bmatrix} -\frac{1}{4}x_1 \\ \frac{1}{2}x_1^2 x_2 - \frac{1}{2\gamma^2}x_2^3 - \frac{1}{2}x_2 \end{bmatrix},$$
$$g(x) = \begin{bmatrix} 0 \\ x_1 \end{bmatrix}, \quad k(x) = \begin{bmatrix} 0 \\ x_2 \end{bmatrix}.$$

In the utility function, $Q = I$, $R = I$, and $\gamma = 2$. Note that the corresponding HJI equation (10) can be solved analytically, and the Nash value of this problem is $V^*(x) = 2x_1^2 + x_2^2$.

TABLE I
Hamiltonian of Policy Evaluation Step

iteration	1	2	3	4
before update	1.5669	1.4310	0.2174	0.0254
after update	0.2144	0.0875	0.0110	0.0010



(a) (b)

Fig. 2. Weights of relaxed policy iteration algorithm. (a) Weights of value network. (b) Relative error of weights.

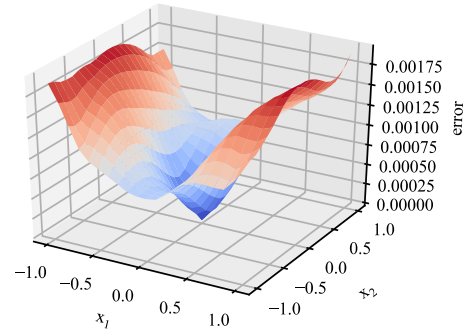


Fig. 3. Absolute error of value.

Choose a fourth-degree polynomial to approximate the value function, i.e., $V\left(x;\omega\right) = \omega^T\ \sigma\ \left(x\right) = \omega_1\ x_1^2 + \omega_2 x_1 x_2 + \omega_3 x_2^2 + \omega_4 x_1^4 + \omega_5 x_1^3 x_2 + \omega_6 x_1^2 x_2^2 + \omega_7 x_1 x_2^3 + \omega_8 x_2^4$. An example of the least squares method is shown in Table I. In the policy evaluation step, the weights of the value network are updated by sampling 2^6 points and minimizing the mean square error. It can be seen that the Hamiltonian (17) after the policy evaluation step is only gradually tending toward zero, which provides evidence that the policy evaluation is biased.

When applying the RPI algorithm, the batch size is set to 2^6, and the learning rate is 10^{-3}. The degree of relaxation α^k in the termination condition (18) is selected as 0.12, which is a tradeoff between the total gradient steps and accuracy in practice. Run the algorithm ten times independently, and the weights of value network are shown in Fig. 2, where dotted lines represent the optimal weights ω^*, colored solid lines and shaded areas represent the mean and range of the learned weights ω, respectively. It can be known from Fig. 2(b) that the relative error of ω can reach 0.1% in the sense of Euclidean norm. Note that $(1-0.12)^{50} \approx 0.2\%$. The upper bound of the weight error is reasonable. Therefore, the theoretical study in the previous section for affine nonlinear systems is validated.

To eliminate the influence of approximator type on the algorithm, an MLP with two hidden layers, each of which contains 2^6 neurons, is employed to approximate the value function. Run the algorithm ten times independently, with the rest of the settings left unchanged. The mean value of the absolute error of the value network in the range $[-1,\ 1] \times [-1,\ 1]$ is shown in Fig. 3, where the error is also around 0.1%. This shows that the selection of the approximation function does not affect the application and accuracy of the algorithm.

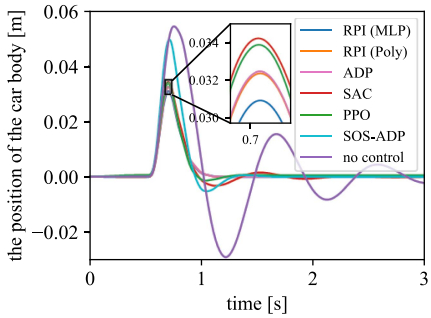


Fig. 4. Control effects of the suspension system.

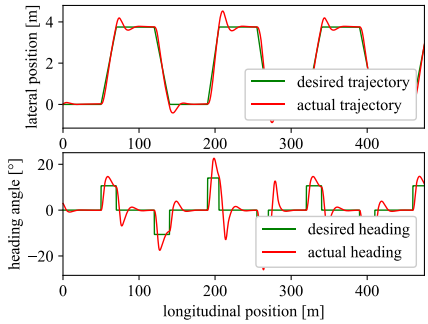


Fig. 5. Tracking effect.

B. Suspension System

Consider an unknown suspension system described in [16], whose parameters and nonlinear dynamics are given by:

$$\dot{x}_1 = x_2, \dot{x}_3 = x_4$$

$$\dot{x}_2 = -\frac{1}{M_b}\left[K_a\left(x_1 - x_3\right) + K_n(x_1 - x_3)^3 + C_a\left(x_2 - x_4\right) - u\right]$$

$$\dot{x}_4 = \frac{1}{M_{us}}\left[-M_b\dot{x}_2 - K_t\left(x_3 - w\right)\right].$$

Set $Q = \mathrm{diag}([\ 1000 \quad 3 \quad 100 \quad 0.1\ ])$, $R = I$, and $\gamma = 30$.

System identification [24], [25] is integrated into the policy training process to form a data-driven algorithm without prior knowledge about the dynamic model, attain excellent sample efficiency, and avoid plain combinations. At each Newton iteration, samples are collected from the actual system, and the approximate dynamic model is updated by performing several gradient descent steps on the supervised learning objective. Historical data are utilized to achieve persistent learning. Meanwhile, the continuously updated approximate dynamic model is adopted to run the RPI algorithm with polynomial and MLP. The used MLP shares the same structure as the previous example. The step size α^k is also selected as 0.12. The batch size is set to 2^6, and the learning rate is 10^{-3}.

In order to evaluate the model-free version of RPI after 50 Newton iterations, a single bump is applied as road disturbance

$$w\ (t) = \begin{cases} 0.038\left(1 - \cos\left(8\pi t\right)\right), & t \in [0.5, 0.75) \\ 0, & t \notin [0.5, 0.75) \end{cases}$$

and the trajectories of the car body are plotted in Fig. 4. For comparison, the results of the latest and best-known RL/ADP methods, including ADP [24], SAC [26], PPO [27], and SOS-ADP [16], are also shown. SOS-ADP solves a transformed relaxed optimization problem, and the approximation may lead to unsatisfactory results. The most commonly used RL algorithms, including stable and effective on-policy method PPO and efficient off-policy method SAC without asymptotic convergence guarantees, achieve better control effect. RPI and ADP algorithms with polynomials attain the same control effect since they solve the original problem with the same approximation structure. The RPI algorithm with MLP shows improvement in control effect over other results due to theoretical guarantees and extended approximation capability.

C. Vehicle Tracking

Consider a dynamic vehicle model with horizontal slope disturbance, which is a general nonlinear system

$$\begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{\omega}_r \\ \dot{\varphi} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} a_x - \frac{F_{yf}\sin\delta}{m} + v_y\omega_r \\ \frac{F_{yf}\cos\delta + F_{yr}}{m} - v_x\omega_r \\ \frac{aF_{yf}\cos\delta - bF_{yr}}{I_{zz}} \\ \omega_r \\ v_x\sin\varphi + v_y\cos\varphi \end{bmatrix} + \begin{bmatrix} g\sin\varphi \\ g\cos\varphi \\ 0 \\ 0 \\ 0 \end{bmatrix}\sin\beta$$

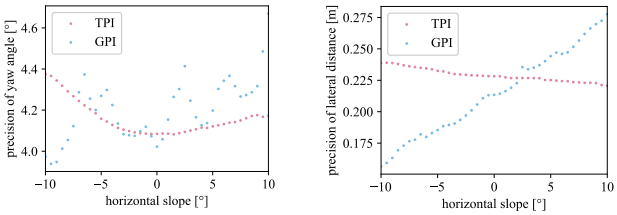
where v_x and v_y are longitudinal and lateral velocities, ω_r is the yaw rate, φ and y are the yaw angle and lateral distance between the vehicle and the reference trajectory, respectively. The reference trajectory is a periodic double-lane change, as shown in Fig. 5. The control input is $u = [\delta\ a_x]^T$, where δ is the steering angle and a_x is the longitudinal acceleration. The disturbance w is a sine function of the horizontal slope β. The lateral tire force is approximated via the Fiala tire model [28]. All parameters are taken from [28]. The utility function is

$$l\left(x, u, w\right) = 16(v_x - 12)^2 + 0.02\omega_r^2 + 18\varphi^2 + 40y^2 + 0.1\delta^2 + 0.3a_x^2 - 5^2 w^T w$$

where the desired longitudinal velocity is 12 m/s.

To implement the TPI algorithm, three MLPs with five hidden layers, each of which has 2^5 neurons, are employed. The activation functions of hidden layers are ELU, and those of the output layers of the value network, policy network, and disturbance network are selected as softplus, tanh, and tanh, respectively. The output of the policy network multiplies $[\pi/9\ 3]^T$ to adjust the amplitude of control. The batch size is set to 2^8. Learning rates of networks are 4×10^{-5}, 10^{-5}, and 10^{-5}, and Adam method is used to update networks [19].

After 400 thousand gradient steps, the trajectory tracking effect of the vehicle is shown in Fig. 5, where the horizontal slope is set to $w = \sin 1^\circ$. To further verify the robustness of the TPI algorithm, a comparison simulation is performed with the GPI algorithm [28], whose dynamic parameters, utility function, network framework, and learning process are the same as those of our algorithm, apart from the omission of disturbance training. The root mean square error from the reference trajectory is applied to represent the control precision. It can be found from Fig. 6 that the precision of the TPI algorithm changes slightly within the range of slope angles from - 10° to 10° compared with the GPI algorithm. Therefore, the TPI algorithm with adversarial training has a better robust performance in this general nonlinear case.



(a) (b)

Fig. 6. Robustness of TPI algorithm. (a) Precision of yaw angle. (b) Precision of lateral distance.

VI. Conclusion

A relaxed policy iteration algorithm is presented in this article to solve nonlinear zero-sum games. The issue that policies are difficult to evaluate accurately in the actual deployment of algorithms is emphasized. Therefore, precise policy evaluation is relaxed by an inequality termination condition. Convergence properties are proven by leveraging an equivalent Newton's method, and exponential convergence speed is derived. Simulation results demonstrate that the algorithm converges to the Nash solution for affine plants with around thousandths accuracy after 50 iterations, keeping pace with theoretical derivation. Moreover, the obtained solution has better resistance to disturbances for general nonlinear plants. The convergence and computational efficiency studies for general nonlinear systems are fascinating and challenging topics that we will leave to future work.

References

[1] T. Basar and P. Bernhard, *H∞ Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*. Berlin, Germany: Springer, 2008.

[2] G. Zames, "Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses," *IEEE Trans. Autom. Control*, vol. AC-26, no. 2, pp. 301–320, Apr. 1981.

[3] J. A. Ball and J. W. Helton, "Nonlinear H∞ control theory for stable plants," *Math. Control, Signals Syst.*, vol. 5, no. 3, pp. 233–261, 1992.

[4] A. J. Van Der Schaft, "L_2-gain analysis of nonlinear systems and nonlinear state feedback H∞ control," *IEEE Trans. Autom. Control*, vol. 37, no. 6, pp. 770–784, Jun. 1992.

[5] A. Isidori and A. Astolfi, "Disturbance attenuation and H∞-control via measurement feedback in nonlinear systems," *IEEE Trans. Autom. Control*, vol. 37, no. 9, pp. 1283–1293, Sep. 1992.

[6] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[7] D. Liu, Q. Wei, D. Wang, X. Yang, and H. Li, *Adaptive Dynamic Programming With Applications in Optimal Control*. Berlin, Germany: Springer, 2017.

[8] Z. Jiang and Y. Jiang, "Robust adaptive dynamic programming for linear and nonlinear systems: An overview," *Eur. J. Control*, vol. 19, no. 5, pp. 417–425, Sep. 2013.

[9] D. Wang, H. He, and D. Liu, "Adaptive critic nonlinear robust control: A survey," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3429–3451, Oct. 2017.

[10] M. Abu-Khalaf, F. L. Lewis, and J. Huang, "Policy iterations on the Hamilton-Jacobi-Isaacs equation for H∞ state feedback control with input saturation," *IEEE Trans. Autom. Control*, vol. 51, no. 12, pp. 1989–1995, Dec. 2006.

[11] M. Abu-Khalaf, F. L. Lewis, and J. Huang, "Neurodynamic program-ming and zero-sum games for constrained control systems," *IEEE Trans. Neural Netw.*, vol. 19, no. 7, pp. 1243–1252, Jul. 2008.

[12] H.-N. Wu and B. Luo, "Neural network based online simultaneous policy update algorithm for solving the HJI equation in nonlinear H∞ control," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 12, pp. 1884–1895, Dec. 2012.

[13] H. Modares, F. L. Lewis, and Z.-P. Jiang, "H∞ tracking control of completely unknown continuous-time systems via off-policy reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2550–2562, Oct. 2015.

[14] T. Dierks and S. Jagannathan, "Optimal control of affine nonlinear continuous-time systems using an online Hamilton-Jacobi-Isaacs formulation," in *Proc. IEEE 49th Conf. Decis. Control*, 2010, pp. 3048–3053.

[15] D. Wang, H. He, and D. Liu, "Improving the critic learning for event-based nonlinear H∞ control design," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3417–3428, Oct. 2017.

[16] Y. Zhu, D. Zhao, X. Yang, and Q. Zhang, "Policy iteration for H∞ optimal control of polynomial nonlinear systems via sum of squares programming," *IEEE Trans. Cybern.*, vol. 48, no. 2, pp. 500–509, Feb. 2018.

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

[18] Q. Wei and D. Liu, "Adaptive dynamic programming for optimal tracking control of unknown nonlinear systems with application to coal gasification," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 4, pp. 1020–1036, Oct. 2014.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015, pp. 1–9.

[20] H. Zhang, Q. Wei, and D. Liu, "An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games," *Automatica*, vol. 47, no. 1, pp. 207–214, 2011.

[21] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*. Hoboken, NJ, USA: Wiley, 2012.

[22] G. Chowdhary and E. Johnson, "Concurrent learning for convergence in adaptive control without persistency of excitation," in *Proc. IEEE 49th Conf. Decis. Control*, 2010, pp. 3674–3679.

[23] L. V. Kantorovich and G. P. Akilov, *Functional Analysis*, 2nd ed. New York, NY, USA: Pergamon, 1982.

[24] S. Xue, B. Luo, D. Liu, and Y. Yang, "Constrained event-triggered H∞ control based on adaptive dynamic programming with concurrent learning," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 52, no. 1, pp. 357–369, Jan. 2022.

[25] X. Yang, H. He, Q. Wei, and B. Luo, "Reinforcement learning for robust adaptive control of partially unknown nonlinear systems subject to unmatched uncertainties," *Inf. Sci.*, vol. 463, pp. 307–322, 2018.

[26] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.

[27] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017. [Online]. Available: http://arxiv.org/abs/1707.06347

[28] J. Duan, Z. Liu, S. E. Li, Q. Sun, Z. Jia, and B. Cheng, "Adaptive dynamic programming for nonaffine nonlinear optimal control problem with state constraints," *Neurocomputing*, vol. 484, pp. 128–141, 2022.