WARWICK
THE UNIVERSITY OF WARWICK

Manuscript version: Author's Accepted Manuscript
The version presented in WRAP is the author's accepted manuscript and may differ from the published version or Version of Record.

Persistent WRAP URL:
http://wrap.warwick.ac.uk/182612

How to cite:
Please refer to published version for the most recent bibliographic citation information. If a published version is known of, the repository item page linked to above, will contain details on accessing it.

Copyright and reuse:
The Warwick Research Archive Portal (WRAP) makes this work by researchers of the University of Warwick available open access under the following conditions.

Copyright © and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable the material made available in WRAP has been checked for eligibility before being made available.

Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

Publisher's statement:
Please refer to the repository item page, publisher's statement section, for further information.

For more information, please contact the WRAP Team at: wrap@warwick.ac.uk.

warwick.ac.uk/lib-publications

ADP-Based Optimal Control for Discrete-Time Systems with Safe Constraints and Disturbances

Jun Ye, Hongyang Dong, Yougang Bian, *Member, IEEE*, Hongmao Qin, and Xiaowei Zhao*

***Abstract*—In this paper, a novel adaptive dynamic programming (ADP)-based optimal control method is developed for discrete-time systems subject to constraints and disturbances. Particularly, a safe policy iteration scheme is designed to handle state and input constraints, including both hard and soft constraints, by converting the original policy improvement strategy into a constrained optimization problem with a prescribed state cost function. After that, an actor-critic-disturbance framework is introduced to address the constrained optimal control problem. The robust safety against disturbances is treated as a two-player zero-sum game, where the actor and disturbance neural networks are used to approximate the optimal control input and the disturbance policy, respectively. The convergence property of the proposed algorithm is analyzed, and the multi-step version of the proposed ADP scheme is derived based on this property. Simulation results are demonstrated and discussed to validate the effectiveness and performance of the proposed method.**

***Note to Practitioners*—Addressing constraints in optimal control problems is essential for guaranteeing the safe operation of controlled systems. However, conventional ADP algorithms struggle to simultaneously manage state and control input constraints during the search for the optimal solution. In real-world applications, another critical and common issue is the presence of external disturbances, where disturbances that cause the control object to deviate from the safe region must be constrained while seeking an optimal control policy. Bearing these factors in mind, this study presents a novel ADP scheme for solving optimal control problems of discrete-time systems, taking into account state and control constraints as well as the impact of disturbances. Moreover, the convergence analysis of the proposed SADP scheme is provided, offering a powerful theoretical foundation for guaranteeing the safety and feasibility of the controlled system during operation.**

***Index Terms*—Adaptive dynamic programming, discrete-time systems, optimal control, constrained control, zero-sum game.**

I. Introduction

With the ever-accelerated updating of computation and control technology, various traditional intelligent control methods, such as sliding mode control [1], model predictive control (MPC) [2], [3], linear quadratic regulator control [4] and so on, are achieved significant development recently. As an important branch of advanced control theory, optimal control also has been thriving in recent years. This kind of control method focuses on obtaining an optimal admissible control policy so that the pre-defined performance index function can reach maximum or minimum value with the prerequisite of stabilizing controlled systems. In real applications, optimal control methods have received widespread attention in various fields, such as intelligent manufacturing, smart grid [5], [6], self-driving cars [7], [8], waverider vehicles (WVs) [9], [10], hypersonic flight vehicles (HFVs) [11], [12], and so on. As a typical optimal control strategy, dynamic programming (DP) has solid theoretical foundations. The core idea of DP is decoupling a complicated problem into some subproblems. Nevertheless, DP is not suitable for dealing with high-dimensional systems due to the underlying Curse of Dimensionality issue. To mitigate this issue, Werbos [13] proposed various adaptive dynamic programming (ADP) schemes that can obtain optimal solution using approximated technology, projected forward in time. Today, ADP, as an intelligent control technology that combines optimal control theory and reinforcement learning (RL), has been widely applied to solve a diverse range of complex nonlinear optimization and decision-making problems [14]–[17].

Ensuring the safety in control under dynamic environments is crucial for real-world applications. However, there are still noticeable shortcomings when applying ADP methods to cases that necessitate consideration of constraints and therefore ensure control safety. In fact, nonlinear MPC (NMPC) is a closed-loop optimal control scheme based on nonlinear models, which serves as a powerful tool for addressing nonlinear optimal problems under constraints. For instance, Shen [18] proposed an innovative distributed NMPC method for AUV tracking by effectively decomposing the original optimization problems into smaller-sized subproblems and solving them in a distributed fashion. Since MPC is a technique for solving optimization problems online, it inherently involves a significant computational burden, which becomes increasingly pronounced as system complexity increases. In contrast, ADP schemes offer a notable advantage in terms of computational efficiency [19]. This is because applying iterative control inputs derived from either online or offline ADP techniques to control systems involves only simple arithmetic operations. Consequently, developing a safe and efficient scheme based on the ADP technique is both essential and urgent.

Both input and state constraints are important in guaranteeing control safety. Input constraints typically pertain to the

This work has received funding from the UK Engineering and Physical Sciences Research Council under Grant EP/S001905/1. *(Corresponding author: Xiaowei Zhao.)*

Jun Ye is with the College of Mechanical and Vehicle Engineering, Hunan University, Changsha 410082, China, and also with the Intelligent Control and Smart Energy (ICSE) Research Group, School of Engineering, University of Warwick, Coventry CV4 7AL, UK. (e-mail: junye@hnu.edu.cn).

Hongyang Dong and Xiaowei Zhao are with the Intelligent Control and Smart Energy (ICSE) Research Group, School of Engineering, University of Warwick, Coventry CV4 7AL, UK. (e-mails: hongyang.dong@warwick.ac.uk, xiaowei.zhao@warwick.ac.uk).

Yougang Bian and Hongmao Qin are with the College of Mechanical and Vehicle Engineering, Hunan University, Changsha 410082, China (e-mails: byg10@foxmail.com, qinhongmao@hnu.edu.cn).

operational limitations of actuators, while state constraints often relate to user-defined safe regions during the control phase [20]. In the context of ADP techniques, addressing control input constraints can be achieved by modifying the utility function's form or by utilizing a specific saturation function at the output of the actor neural network. For instance, a novel non-quadratic performance function is introduced in [21]–[23] to accommodate various input constraints, and the convergence of these iterative ADP algorithms is also analyzed. In [24], a novel goal representation adaptive critic design (ACD) is proposed in the event-triggered context for uncertain discrete-time systems with input saturation, saving unnecessary resource consumption while guaranteeing control performance. Handling state constraints within ADP methods is more challenging than addressing control input constraints, primarily due to the use of policy gradient algorithms. Neglecting state constraints during the design process can lead to suboptimal performance and reduced safety in practical applications, potentially resulting in unforeseen consequences. To overcome this issue, some researchers [25]–[27] have employed methods that incorporate additional terms into the cost function, striking a balance between safety and optimality.

However, these methods cannot strictly guarantee that systems consistently comply with safety constraints, as there is an inherent trade-off between optimality and safety [28]. Recently, constrained policy optimization (CPO) [29] has been introduced, building upon trust region policy optimization [30]. With this new technique, Duan [31] proposed a constrained generalized policy iteration (CGPI) scheme to tackle optimal control problems with state constraints by constructing a trust region optimization problem within the policy improvement (PIM) framework. Nevertheless, this scheme may encounter situations where no feasible policy can satisfy various constraints while keeping the cost function bounded.

By considering actual scenarios, some safety constraints are often not very strict and can be slightly violated, i.e., treated as soft constraints. However, certain safety constraints, such as collision distance restrictions and control limits, must be maintained within the safety margin [26]. Based on this understanding, this study develops a practical ADP scheme to address both soft and hard constraints in accordance with real-world requirements, achieving a balance between safety and control performance.

Moreover, in reality, complex environments are often dynamic and time-varying, introducing random external disturbances into the dynamic system. As a result, it becomes difficult to establish an accurate mathematical model, leading to a mismatch between the dynamic model and the controller. This mismatch can have a detrimental impact on control performance [32]. To address external disturbances, Che [33] developed a novel ADP framework to estimate rudder faults and ocean current disturbance by introducing neural network estimators and integrating both rudder faults and external disturbances within the utility function. Duan [34] presented a model-free scheme that introduced a bounded L_2 -gain architecture to learn the optimal solution of the designed Hamilton-Jacobi-Isaac equation for a vehicle system in the presence of unknown disturbances. Carlucho [35] proposed an adaptive control framework based on an actor-critic goal-oriented deep RL structure for vehicle systems, with the feasibility of this framework demonstrated through real experiments. Drawing inspiration from these works, we develop an effective actor-critic-disturbance ADP framework in this work to mitigate the negative effects caused by imprecise model information while considering input and state constraints.

In this paper, a novel ADP scheme is proposed for solving optimal control problems of discrete-time systems with state and control constraints under the influence of disturbances. The main contribution of this paper can be summarized as follow:

1) Based on the practical characteristics of hard and soft constraints, an innovative safe ADP (SADP) scheme with a policy iteration (PI) method is proposed. This approach considers both state and control input constraints during the search for the optimal control policy. In comparison to common techniques that embed barrier functions into the cost function [25]–[27], the proposed algorithm ensures excellent balance between optimality and safety. Furthermore, when compared to the multi-state constraint form [31], the proposed scheme significantly reduces computational burden by employing only a single inequality to represent aggregated state constraints.
2) The optimal control problem subject to disturbances is formulated as a two-player zero-sum game problem. In contrast to prevailing methods that solely treat disturbances as an additional policy interacting with the optimal control policy in this type of non-cooperative game problems [36]–[39], the state and input constraints are persistently incorporated throughout the game process. This ensures that scenarios leading to the control object deviating from the safe region are effectively constrained during the search for the optimal control policy under the influence of disturbances. The convergence property of SADP subject to disturbances is analyzed. Furthermore, the feasibility of the multi-step SADP scheme is also illustrated.
3) The proposed algorithm possesses wide flexibility and applicability to various variants. Its adaptability to operate both with and without accurate model information highlights its robustness in addressing various real-world scenarios. By strategically leveraging model information to derive the future state of the control object, the algorithm achieves enhanced control performance and safety. Furthermore, when faced with unavailable or inaccurate model information, the algorithm can adopt data-driven techniques, liberating it from reliance on the model and enabling seamless adaptation to real-world conditions. This multifaceted approach enhances the algorithm's versatility and effectiveness in tackling different control challenges.

The rest of the paper is organized as follows. In Section II, the optimal control problem is described, and safe constraint forms are formulated based on practical requirements. In

Section III, the iteration steps of SADP algorithm are derived, and the convergence property is analyzed. In Section IV, the implementation details are developed. In Section V, simulation studies and analyses are given to demonstrate the effectiveness of the proposed scheme. Finally, in Section VI, conclusions are drawn, and future work is outlined.

II. Problem Formulation

A. Optimal Control Problem

Consider the discrete-time nonlinear system with disturbances:

$$x_{t+1} = f(x_t) + g(x_t)u_t + w(x_t)\omega_t, t = 0,1,2,\dots \quad (1)$$

where $x_t \in \mathbb{R}^m$ denotes the state variable, $u_t \in \mathbb{R}^n$ denotes the control input, and $\omega_t \in \mathbb{R}^n$ denotes the disturbance term, with n and m represent the dimensions of state and control spaces, respectively. The functions $f(\cdot)$, $g(\cdot)$ and $w(\cdot)$ are considered to be Lipschitz continuous on a compact set Ω that includes the origin point. It can be assumed that system (1) is stabilizable, i.e., there exists a set of effective control policies that can asymptotically stabilize the system in (1) on Ω. As for the disturbance policy ω_t, its amplitude is assumed to be within a bound ω_m, i.e., $\|\omega_t\| \le \omega_m$ and $\omega_t \in \Psi_\omega$, where Ψ_ω is a set that contains all disturbance policies.

For the optimal control problem, the cost function can be constructed as

$$\mathbb{V}(x_t) = \sum_{l=t}^{\infty} \gamma^{l-t} U(x_l, u_l, \omega_l), \quad (2)$$

where $0 < \gamma \le 1$ denotes the discount factor, x_l represents the state vector beginning at a specific state x_t, and u_l and ω_l represent the corresponding control policy and disturbance policy, respectively.

In (2), the utility function can be expressed as

$$U(x_t, u_t, \omega_t) = x_t^T Q x_t + u_t^T R u_t - \beta \omega_t^T \omega_t, \quad (3)$$

where Q and R are positive-definite weighting matrices for the state and input, respectively, and β is a positive constant denoting the user-determined resistant degree with respect to the disturbance.

Based on (2) and Bellman's optimality principle, the optimal cost function can be expressed as

$$\mathbb{V}^*(x_t) = \min_{u_t} \max_{\omega_t} \big(U(x_t, u_t, \omega_t) + \gamma \mathbb{V}^*(x_{t+1})\big). \quad (4)$$

Then, the control object is to search an optimal control policy:

$$u_t^* = \arg\min_{u_t} \big(U(x_t, u_t, \omega_t) + \gamma \mathbb{V}^*(x_{t+1})\big). \quad (5)$$

Such an optimal control problem considering disturbances can be regarded as a two-player zero-sum game, in which the worst-case disturbance can be given as

$$\omega_t^* = \arg\max_{\omega_t} \big(U(x_t, u_t, \omega_t) + \gamma \mathbb{V}^*(x_{t+1})\big), \quad (6)$$

B. Safe Constraints Handling

For discrete-time general optimal control problems, [40], [41] present that the optimal control policy can be achieved by implementing ADP techniques. Further, in order to deal with optimal control problems subject to disturbances, [36]–[39] shows that (4)-(6) can be solved based on the actor-critic-disturbance architecture. However, in actual fact, most systems are safety-critical in practical applications, such as vehicle systems [42], spacecraft [43], and robot [44], but all the ADP schemes above cannot guarantee safety during the search for the optimal solution.

Particularly, to achieve an excellent balance between optimality and control performance, control inputs are constrained by using a smooth and saturated function based on practical requirements. This means that the constrained control input is set boundedly, i.e., $\Psi_u = \{u \big| |u_p| \le \bar{u}_p, p = 1,2,\dots,n\}$, where $\bar{u}_p$ is the bound for the p-th control input.

Remark 1: Although differentiating the performance function accumulated by non-quadratic utility with respect to the control input can effectively confine the control input within the safe region [25]–[27], this approach is only compatible with incorporating state penalty terms into the performance function to guarantee safety. However, the penalty degree is empirically self-defined, leading to a lack of solid theoretical foundation and design standards for balancing safety abilities and control performance [25], [28]. As a result, the incorporating method is not an excellent choice to deal with state and control input constraints while guaranteeing optimality.

After finalizing the handling of control input constraints, the specific way of coping with state constraints should be determined further. Motivated by [45], [46], the control barrier function (CBF) can be employed to handle state constraints. First, a safe set $\mathcal{C}_s$ is defined as

$$\mathcal{C}_s = \{x \in \mathbb{R}^m | h(x) \le 0\}, \quad (7)$$

$$\partial\mathcal{C}_s = \{x \in \mathbb{R}^m | h(x) = 0\}, \quad (8)$$

$$Int(\mathcal{C}_s) = \{x \in \mathbb{R}^m | h(x) > 0\}, \quad (9)$$

where $h(\cdot)$ is a continuously differentiable function, $h(\cdot) \le 0$ represents the safe region of constrained states. Then, the unsafe set $Int(\mathcal{C}_s)$ can be defined as the complementary set of $\mathcal{C}_s$. In this study, the purpose is to design an optimal control policy to guarantee that a sequence of state $\{x | x_1^u, x_2^u, x_3^u \dots\}$ will not enter $Int(\mathcal{C}_s)$ even under the influence of disturbances.

CBFs have the potential to effectively restrict states within the safe region [45]. Motivated by [47], a generalized CBF is adopted in this study

$$h(x_{c,t+1}) \le (1 - \gamma_c) h(x_{c,t}), \quad (10)$$

where $x_{c,t}$ represents the constrained state at time t, and $h(x_{c,t}) \le 0$. $0 < \gamma_c \le 1$ represents the conservativeness coefficient.

Moreover, based on (10), a safe cost function $J_c(x_{c,t})$ with the generalized CBF is introduced as follows

$$\begin{aligned} J_c(x_{c,t}^{N_c}) &= \sum_{l=t}^{t+N_c-1} h(x_{c,l}) \\ &\le \sum_{l=t}^{t+N_c-1} (1-\gamma_c)^{l-t} h(x_{c,t}) \le 0. \end{aligned} \quad (11)$$

where N_c denotes the predicted step for the constrained state. The implementation of the conservativeness coefficient also ensures that immediate constraints are more crucial than guaranteeing constraints in future steps.

Remark 2: It is worth noting that the safe control region

$\mathcal{U}_x \in \Psi_u$ is a set related to the control input. For each state variable x, there exists a corresponding safe set $\mathcal{U}_x$ ensuring that the system can reach the safe region.

Remark 3: In reality, a single-step constraint on states may not achieve an excellent trade-off between control performance and safety. For instance, in the control process of autonomous vehicles, if safety constraints are considered for only one step (i.e., the desired trajectory and boundary information are not taken into account in future steps), there is a high probability that the system will enter an unsafe region for those states that need to be constrained and are already close to the safety boundary. This is because it may not be possible to determine the appropriate control input to be implemented on the vehicle system in the next step. In other words, $\mathcal{U}_x$ may be an empty set. Moreover, even if an admissible control policy is achievable after solving, it may still lead to unsafe effects in future steps. Drawing from human-driving behaviors, the feasible driving region is evaluated over a certain distance, meaning that the current driving behavior is indeed influenced by future states. If the vehicle cannot safely pass through the feasible region, drivers will take action, such as braking ahead of time. That's why we consider future steps in Eq. (11).

Based on Eq. (11), to obtain a far-sighted effect, the safe state constraint is given as

$$J_c\left(x_{c,t}^{N_c}\right) \le 0, \tag{12}$$

Compared with the state function defined in [31], which requires [$N_c \times$ (the number of constraints)] inequations to describe a similar constrained optimal control problem as in this paper, we just need an aggregated inequation, e.g. Eq. (12), to describe the constraint handling requirements. This aggregated expression can also significantly improve the solving efficiency and effectiveness.

Moreover, for the existing safe-certified ADP method with CBF [26], [48], the admissible control space based on CBF should be added as the extra constraint into the optimal control problem, and this kind of constraint is associated with model information closely. Both of these factors will bring extra computation burden in the process of solving the specific optimization problem. Meanwhile, the Bellman equation used for solving the optimal control problem should be relaxed by introducing an iterative slack variable in this method, which will also bring the challenge of balancing optimality and safety, and further exacerbate the computational burden. Therefore, based on these analyses above, the constraint about admissible control space derived from designed CBF is not applied. In this study, the CBF is applied for designing the safe cost function. Based on the safe cost function, the specific method of handling state constraints will be presented in Section IV-B.

Subsequently, the control object becomes to get a constrained optimal control policy, such that

$$\begin{aligned} u_t^* &= \arg\min_{u_t}\left(U(x_t,u_t,\omega_t) + \gamma\mathbb{V}^*(x_{t+1})\right) \\ &\text{s.t. } J_c\left(x_{c,t}^{N_c}\right) \le 0. \end{aligned} \tag{13}$$

Conventional ADP schemes are inapplicable for such a complex constrained optimal control problem, not to mention the DP method. To address this issue, a safe ADP (SADP) scheme based on the CBF and the trust region concept is proposed. Key details are presented in the following section.

III. Safe Adaptive Dynamic Programming

A. Derivation of PI Safe Adaptive Dynamic Programming

To ensure a more stable implementation of SADP in practical applications, the PI method is adopted in this study. This choice is made because the control policy is consistently admissible during the learning process of PI compared to the value iteration method [40], [41], [49]. To further analyze the iterative properties of the proposed scheme, an action-disturbance-dependent function is introduced as follows:

$$\begin{aligned} V(x_t,u_t,\omega_t) &= U(x_t,u_t,\omega_t) + \sum_{l=t+1}^{\infty}\gamma^{l-t}U(x_l,u_l,\omega_l) \\ &= U(x_t,u_t,\omega_t) + \gamma\mathbb{V}(x_{t+1}). \end{aligned} \tag{14}$$

It is noted that $V(x_t,u_t,\omega_t)$ is related to the state, the control policy, and the disturbance. Moreover, one can see that, in (14), the behavior policy is same with the evaluation policy. Therefore, Eq. (14) actually indicates an on-policy control scheme. Then, according to (4) and (14), the optimal cost function $V^*(x_t,u_t,\omega_t)$ can be expressed as

$$\begin{aligned} V^*(x_t,u_t,\omega_t) &= \min_{u_t}\max_{\omega_t}\left(U(x_t,u_t,\omega_t) + \gamma\mathbb{V}^*(x_{t+1})\right) \\ &= \min_{u_t}\max_{\omega_t}\big(U(x_t,u_t,\omega_t) \\ &\quad + \gamma V^*(x_{t+1},u_{t+1},\omega_{t+1})\big). \end{aligned} \tag{15}$$

For zero-sum game problems, according to (5), (6), and (15), the optimal control policy and worst-case disturbance solved by the on-policy method are given by (16) and (17), respectively.

$$\begin{aligned} u_t^* &= \arg\min_{u_t}\left(V^*(x_t,u_t,\omega_t)\right) \\ &\text{s.t. } J_c\left(x_{c,t}^{N_c}\right) \le 0. \end{aligned} \tag{16}$$

$$\omega_t^* = \arg\max_{\omega_t}\left(V^*(x_t,u_t,\omega_t)\right). \tag{17}$$

To obtain u_t^*, the cost function, control policy, and disturbance policy are updated by iterations. The iteration rule can be presented with policy evaluation (PEV) inner loops and PI outer loops. For clarity, the inner loop iteration is represented by j, and the outer loop is represented by i. Both i and j start from zero and continue to infinity.

Given an arbitrary initial admissible control policy $u_{0,t} \in \mathcal{U}_x$ and a disturbance $\omega_{0,t} \in \Psi_\omega$, the iterative cost function $V_0(x_t,u_{0,t},\omega_{0,t})$ can be achieved as

$$\begin{aligned} V_0\left(x_t,u_{0,t},\omega_{0,t}\right) &= U\left(x_t,u_{0,t},\omega_{0,t}\right) \\ &\quad + \gamma V_0\left(x_{t+1},u_{0,t+1},\omega_{0,t+1}\right). \end{aligned} \tag{18}$$

It should be emphasized that $V_0\left(x_t,u_{0,t},\omega_{0,t}\right) = V_{0,\infty}\left(x_t,u_{0,t},\omega_{0,t}\right)$ as $j \to \infty$. Then, the control policy $u_{1,t}$ and disturbance policy $\omega_{1,t}$ can be derived as

$$\begin{aligned} u_{1,t} &= \arg\min_{u_t}\left(V_0\left(x_t,u_t,\omega_{0,t}\right)\right) \\ &\text{s.t. } J_c\left(x_{c,t}^{N_c}\right) \le 0. \end{aligned} \tag{19}$$

$$\omega_{1,t} = \omega_{0,t} + \alpha_\omega\nabla_\omega V_0\left(x_t,u_{1,t},\omega_{0,t}\right)|_{\omega=\omega_0}. \tag{20}$$

Subsequently, for $i = 1,2,3,\ldots$, the iterative cost function can be derived as

$$V_{i,j+1}(x_t,u_{i,t},\omega_{i,t}) = U(x_t,u_{i,t},\omega_{i,t}) + \gamma V_{i,j}(x_{t+1},u_{i,t+1},\omega_{i,t+1}). \quad (21)$$

Moreover, the iterative control and disturbance policies can be denoted as

$$u_{i+1,t} = \arg\min_{u_t}\left(V_i(x_t,u_t,\omega_{i,t})\right) \text{ s.t. } J_c(x_{c,t}^{N_c}) \le 0. \quad (22)$$

$$\omega_{i+1}(x_t) = \omega_i(x_t) + \alpha_\omega \nabla_\omega V_i(x_t,u_{i+1,t},\omega_{i,t})|_{\omega=\omega_i}. \quad (23)$$

For conventional ADP schemes, neural networks (NNs) are typically introduced for updating control policies with gradient descent methods. However, this kind of methods cannot handle optimal control problems with state constraints since conventional PIM is employed only to optimize unconstrained scenarios. Here we develop a novel safe ADP (SADP) scheme to address the constrained optimization problem described in (19) and (22). However, solving (19) and (22) with analytical solutions directly is almost impossible because of the strong nonlinear properties of the objective function and constraints, especially under high-dimensional state & action spaces. To address this challenging problem, we employ the linearization technique. Specifically, the cost function and state constraints are linearized based on the previous control policy. Given this logic, the objective function can be approximated as

$$V_i(x_t,u_t,\omega_{i,t}) = V_i(x_t,u_t,\omega_{i,t})|_{u_t=u_{i,t}} + \left(\nabla_{u_t}V_i(x_t,u_t,\omega_{i,t})|_{u_t=u_{i,t}}\right)^T(u_t - u_{i,t}) + R_1^v(u_t). \quad (24)$$

In fact, $V_i(x_t,u_t,\omega_{i,t})$ can be regarded as a function w.r.t u_t, where u_t is an independent variable. The first term on the right-hand side of (24) represents a specific value of the performance function at $u_{i,t}$. Furthermore, $\left(\nabla_{u_t}V_i(x_t,u_t,\omega_{i,t})|_{u_t=u_{i,t}}\right)$ in the second term denotes the specific derivation information of the i-th performance function at $u_{i,t}$. Additionally, u_t in $(u_t - u_{i,t})$ corresponds to u_t in the left-hand side of (24), which is a control policy that needs to be solved.

Moreover, the constrained cost function is approximated by

$$J_c(x_{c,t}^{N_c}) = J_c(x_{c,t}^{N_c})|_{u_t=u_{i,t}} + \left(\nabla_{u_t}J_c(x_{c,t}^{N_c})|_{u_t=u_{i,t}}\right)^T(u_t - u_{i,t}) + R_1^c(u_t) \quad (25)$$

Here $R_1^v(u_t)$ and $R_1^c(u_t)$ represent residual errors.

Remark 4: It is worth noting that $J_c(x_{c,t}^{N_c})$ is related to the current control policy $u_{i,t}$ instead of only relating the future state. This is because predicted state constraints are derived from initial state x_t with current control policy $u_{i,t}$. In other words, if the executed control policy is changed, the approximate result will be different.

To ensure that the approximation scheme is feasible, the extent of the control policy update should be limited within a small range. In order to reasonably measure the update extent of the control policy, $D(u_t,u_{i,t})$ is introduced to represent the distance between the new and old control policies, which can be defined as follows:

$$D(u_t,u_{i,t}) \triangleq \mathbb{E}_{x_t\sim\vartheta_x}\left[\|u_t - u_{i,t}\|_2^2\right]. \quad (26)$$

where $\mathbb{E}_{x_t\sim\vartheta_x}$ represents the expectation with regard to the state distribution ϑ_x on Ω.

Then, we also define a limited range for $D(u_t,u_{i,t})$ as δ. Therefore, we can get a new constraint:

$$D(u_t,u_{i,t}) \le \delta. \quad (27)$$

$D(u_t,u_{i,t})$ can be approximated by a linearized way:

$$D(u_t,u_{i,t}) = D(u_t,u_{i,t})|_{u_t=u_{i,t}} + \left(\nabla_{u_t}D(u_t,u_{i,t})|_{u_t=u_{i,t}}\right)^T(u_t - u_{i,t}) + \frac{1}{2}(u_t - u_{i,t})^T\left(\nabla_{u_t}^2 D(u_t,u_{i,t})|_{u_t=u_{i,t}}\right)(u_t - u_{i,t}) + R_1^D(u_t). \quad (28)$$

Based on all these analyses, our PIM process can be expressed as

$$\min_{u_t}\left(\nabla_{u_t}V_i(x_t,u_t,\omega_{i,t})|_{u_t=u_{i,t}}\right)^T(u_t - u_{i,t})$$
$$\text{s.t. } J_c(x_{c,t}^{N_c})|_{u_t=u_{i,t}} + \left(\nabla_{u_t}J_c(x_{c,t}^{N_c})|_{u_t=u_{i,t}}\right)^T(u_t - u_{i,t}) \le 0$$
$$\frac{1}{2}(u_t - u_{i,t})^T\left(\nabla_{u_t}^2 D(u_t,u_{i,t})|_{u_t=u_{i,t}}\right)(u_t - u_{i,t}) < \delta. \quad (29)$$

B. Convergence Analysis of PI-based SADP

In this subsection, the convergence property of our PI-based SADP is provided by showing that the cost function $V_{i,j}$, the control policy u_i, and the disturbance policy ω_i can converge to the optimal value as iterations $i,j \to \infty$.

Lemma 1: For any fixed control policy $u_{i,t}$ and disturbance policy $\omega_{i,t}$, $i = 0,1,\ldots$, with the iterative cost function given in (21). Then, for $\forall x_t \in \mathbb{R}^m$, the iterative cost function $V_{i,j}(x_t,u_{i,t},\omega_{i,t})$ will converge to $V_{i,\infty}(x_t,u_{i,t},\omega_{i,t}) = V_i(x_t,u_{i,t},\omega_{i,t})$ as $j \to \infty$.

Proof: The convergence process of policy evaluation (PEV) can be analyzed in two steps.

Step 1: For $j = 0,1,2,\ldots$, the iterative cost function $V_{i,j}(x_t,u_{i,t},\omega_{i,t})$ should satisfy

$$V_{i,j+1}(x_t,u_{i,t},\omega_{i,t}) \le V_{i,j}(x_t,u_{i,t},\omega_{i,t}), \forall j \ge 0. \quad (30)$$

First, for $i = 1$, $j = 0$, and $\forall x_t \in \mathbb{R}^m$,

$$\begin{aligned} V_{1,0}(x_t,u_{1,t},\omega_{1,t}) &= U(x_t,u_{1,t},\omega_{1,t}) + \gamma V_0(x_{t+1},u_{1,t+1},\omega_{1,t+1}) \\ &\le U(x_t,u_{0,t},\omega_{0,t}) + \gamma V_0(x_{t+1},u_{0,t+1},\omega_{0,t+1}) \\ &= V_0(x_t,u_{0,t+1},\omega_{0,t+1}). \end{aligned} \quad (31)$$

For $j = 1$,

$$\begin{aligned} V_{1,1}(x_t,u_{1,t},\omega_{1,t}) &= U(x_t,u_{1,t},\omega_{1,t}) + \gamma V_{1,0}(x_{t+1},u_{1,t+1},\omega_{1,t+1}) \\ &\le U(x_t,u_{1,t},\omega_{1,t}) + \gamma V_0(x_{t+1},u_{1,t+1},\omega_{1,t+1}) \\ &= V_{1,0}(x_t,u_{1,t+1},\omega_{1,t+1}). \end{aligned} \quad (32)$$

Then, for $j = 2,3,\ldots$,

$$\begin{aligned} V_{1,j+1}(x_t,u_{1,t},\omega_{1,t}) &\le U(x_t,u_{1,t},\omega_{1,t}) + \gamma V_{1,j-1}(x_{t+1},u_{1,t+1},\omega_{1,t+1}) \\ &= V_{1,j}(x_t,u_{1,t+1},\omega_{1,t+1}). \end{aligned} \quad (33)$$

Similarly, for $i = 2,3,\ldots$, we can obtain $V_{i,j+1}(x_t,u_{i,t},\omega_{i,t}) \le$

$V_{i,j}(x_t, u_{i,t}, \omega_{i,t})$.

Step 2: According to the monotonically nonincreasing property $V_{i,j+1}(x_t, u_{i,t}, \omega_{i,t}) \le V_{i,j}(x_t, u_{i,t}, \omega_{i,t})$, we can conclude that the cost function $V_{i,j}(x_t, u_{i,t}, \omega_{i,t})$ is bounded for $\forall i, j$. Then, the norm value related to the cost function can be defined as

$$\begin{aligned}\left\|V_{i,j+1}(x_t, u_{i,t}, \omega_{i,t}) - V_{i,j}(x_t, u_{i,t}, \omega_{i,t})\right\|_\infty \\ = \max_{X_t \in \Omega}\left\|V_{i,j+1}(x_t, u_{i,t}, \omega_{i,t})\right. \\ \left. - V_{i,j}(x_t, u_{i,t}, \omega_{i,t})\right\|.\end{aligned} \tag{34}$$

For $j = 0$, the norm value can be denoted as $\Delta_0 = \left\|V_{i,1}(x_t, u_{i,t}, \omega_{i,t}) - V_{i,0}(x_t, u_{i,t}, \omega_{i,t})\right\|$. Then, we can get

$$\begin{aligned}\left\|V_{i,2}(x_t, u_{i,t}, \omega_{i,t}) - V_{i,1}(x_t, u_{i,t}, \omega_{i,t})\right\| \\ = \gamma^N\left\|V_{i,1}(x_{t+N}, u_{i,t+N}, \omega_{i,t+N})\right. \\ \left. - V_{i,0}(x_{t+N}, u_{i,t+N}, \omega_{i,t+N})\right\| \\ \le \gamma^N \Delta_0.\end{aligned} \tag{35}$$

So, as $j \to \infty$, one has

$$\begin{aligned}\left\|V_{i,j+1}(x_t, u_{i,t}, \omega_{i,t}) - V_{i,j}(x_t, u_{i,t}, \omega_{i,t})\right\| \\ = \gamma^N\left\|V_{i,j}(x_{t+N}, u_{i,t+N}, \omega_{i,t+N})\right. \\ \left. - V_{i,j-1}(x_{t+N}, u_{i,t+N}, \omega_{i,t+N})\right\| \\ \le \gamma^{jN} \Delta_0.\end{aligned} \tag{36}$$

We can find that as $j \to \infty$, $\left\|V_{i,j+1}(x_t, u_{i,t}, \omega_{i,t}) - V_{i,j}(x_t, u_{i,t}, \omega_{i,t})\right\|$ will be equivalent to 0. Therefore, the iterative cost function $V_{i,j}(x_t, u_{i,t}, \omega_{i,t})$ will converge to $V_i(x_t, u_{i,t}, \omega_{i,t})$ as $j \to \infty$. ■

Theorem 1: Assume that the constrained optimal control problem in (29) is solvable, and $V_{i,j}(x_t, u_{i,t}, \omega_{i,t})$, $u_{i,t}$, and $\omega_{i,t}$ are given as in (18)-(23), with $u_{i,t} \in \mathcal{U}_x$. Then the iterative policy sequence $u_{i,t}$ and $\omega_{i,t}$ will converge to optimal values u_t^* and ω_t^* for $i = 0,1,2, ...$

Proof: First, according to (21), we have

$$\begin{aligned}V_{i+1,0}(x_t, u_{i+1,t}, \omega_{i+1,t}) \\ = U(x_t, u_{i+1,t}, \omega_{i+1,t}) \\ + \gamma V_{i,\infty}(x_{t+1}, u_{i+1,t+1}, \omega_{i+1,t+1}).\end{aligned} \tag{37}$$

Then, one has

$$\begin{aligned}&V_i(x_t, u_{i,t}, \omega_{i,t}) \\ &= \sum_{l=0}^{N-1}\gamma^l U(x_{t+l}, u_{i,t+l}, \omega_{i,t+l}) \\ &\qquad + \gamma^N V_i(x_{t+N}, u_{i,t+N}, \omega_{i,t+N}) \\ &\ge \sum_{l=0}^{N-1}\gamma^l U(x_{t+l}, u_{i+1,t+l}, \omega_{i+1,t+l}) \\ &\qquad + \gamma^N V_i(x_{t+N}, u_{i,t+N}, \omega_{i,t+N}) \\ &\ge \sum_{l=0}^{2N-1}\gamma^l U(x_{t+l}, u_{i+1,t+l}, \omega_{i+1,t+l}) \\ &\qquad + \gamma^{2N} V_i(x_{t+2N}, u_i(x_{t+2N}), \omega_i(x_{t+2N})) \\ &\vdots \\ &\ge \sum_{l=0}^{\infty}\gamma^l U(x_{t+l}, u_{i+l,t+l}, \omega_{i+1,t+l}) \\ &= V_{i+1}(x_t, u_{i+1,t}, \omega_{i+1,t}).\end{aligned} \tag{38}$$

Thus, the iterative cost function $V_i(x_t, u_{i,t}, \omega_{i,t})$ is monotonically non-increasing.

According to *Lemma 1*, PEV can ultimately lead to a convergent cost value $V_i(x_t, u_{i,t}, \omega_{i,t}) = V_{i,\infty}(x_t, u_{i,t}, \omega_{i,t})$ with specific policies $u_{i,t}$ and $\omega_{i,t}$ under iteration index i as $j \to \infty$, i.e., the accurately cost function with $u_{i,t}$ and $\omega_{i,t}$ can be achieved based on (21) as $j \to \infty$. Then, based on the PIM process given in (29), a minimized iterative cost function with an admissible control policy and a worst disturbance policy is aimed to be achieved. At the same time, according to (38), the iterative cost function $V_i(x_t, u_{i,t}, \omega_{i,t})$ is monotonically non-increasing. Moreover, for $\forall x_t \in \mathbb{R}^m$, $V^*(x_t, u_t^*, \omega_t^*)$ is bounded because u_t^* must be an admissible control policy. On the basis of [41], [50], [51], $V_i(x_t, u_{i,t}, \omega_{i,t})$ will converge to a bounded $V_\infty(x_t, u_{\infty,t}, \omega_{\infty,t})$ as $i \to \infty$. Therefore, we can conclude that $V_\infty(x_t, u_{\infty,t}, \omega_{\infty,t}) = V^*(x_t, u_t^*, \omega_t^*)$ with $u_{i,t} \to u_t^*$ and $\omega_{i,t} \to \omega_t^*$. ■

Remark 5: In this study, to obtain the constrained cost function (11), the state variables should be forwarded for N_c steps with the model information. Therefore, the optimal cost function can also be rewritten as

$$\begin{aligned}V^*(x_t, u_t, \omega_t) = \min_{u_t}\max_{\omega_t}\left(\sum_{l=t}^{t+N_p-1}\gamma^{l-t}U(x_l, u_l, \omega_l)\right. \\ \left. + \gamma^{N_p}V^*\left(x_{N_p+t}, u_{N_p+t}, \omega_{N_p+t}\right)\right).\end{aligned} \tag{39}$$

where N_p denotes the predicted step for the cost function. Then, the iterative rule of the cost function (21) can be rewritten as

$$\begin{aligned}V_{i,j+1}(x_t, u_{i,t}, \omega_{i,t}) \\ = \sum_{l=t}^{t+N_p-1}\gamma^{l-t}U(x_l, u_{i,l}, \omega_{i,l}) \\ + \gamma^{N_p}V_{i,j}\left(x_{N_p+t}, u_{i,N_p+t}, \omega_{i,N_p+t}\right).\end{aligned} \tag{40}$$

In fact, because of the implementation of the on-policy framework, i.e., the behavior policy is same with the evaluation policy, Eq. (40) is also appropriate for *Theorem 1*, and the implementation of multi-step method is capable of improving the accuracy and speed of evaluating the cost function.

Remark 6: Regarding the stability of closed-loop control system, it can be analyzed through the lens of optimality. A controller resulting from a typical adaptive critic technique assures stability since it is essentially an optimal controller. Optimal control guarantees stability with the nonexistence of conjugate points, ensuring that the control objective moves towards the only equilibrium point based on the optimal solution [52]. Furthermore, according to the boundedness of initial admissible control law and the monotonicity of iterative performance function, it can be deduced that the optimal performance function remains finite. This critical property implies that the optimal control law can effectively stabilize the system, as an infinite performance function indicates an inability to achieve stability.

IV. NN Implementation of SADP

In this study, the NN technique is introduced to approximate the cost function $\hat{V}(x_t, \hat{u}_t, \hat{\omega}_t)$, the control policy $\hat{u}_t$, and the disturbance policy $\hat{\omega}_t$ with any specific state x_t. The details are introduced in the following subsections.

A. Implementation of the Critic Network

A three-layer feed-forward NN is constructed to approximate the cost function $\hat{V}(x_t, u_t, \omega_t)$ that is denoted by

$$\hat{V}_{i,j}\left(x_t, \hat{u}_{i,t}, \hat{\omega}_{i,t}\right) = \widehat{W}_{i,j}^{c\,T}\sigma_c\left(Y_C^T\left(x_t, \hat{u}_{i,t}, \hat{\omega}_{i,t}\right)\right). \quad (41)$$

where $\widehat{W}_{i,j}^c \in \mathbb{R}^{(m+2n)\times \hbar_c}$ represents the weight matrix between the hidden and output layers with the inner loop j and the outer loop i, and $\hbar_c$ denotes the number of neurons on the hidden layer. Y_c represents the weight matrix between the hidden and input layers. To approximate the cost function $\hat{V}(x_t, u_t, \omega_t)$ more accurately, a nonlinear activation function $\sigma_c(\cdot)$ is implemented for hidden layer, with an upper bounded σ_{cM}, i.e., $\left\|\sigma_c\left(Y_C^T\left(x_t, \hat{u}_{i,t}, \hat{\omega}_{i,t}\right)\right)\right\| \le \sigma_{cM}$.

In PEV, the accurate cost function should ultimately be obtained by eliminating the evaluation error using the backpropagation mechanism. The evaluation error can be denoted by

$$E_{i,j}^c = \frac{1}{2}\left\|e_{i,j}^c\right\|^2, \quad (42)$$

where

$$\begin{aligned} e_{i,j}^c = {} & \hat{V}_{i,j}\left(x_t, \hat{u}_{i,t}, \hat{\omega}_{i,t}\right) \\ & - \left(\sum_{l=t}^{t+N_p-1}\gamma^{l-t}U\left(x_l, \hat{u}_{i,l}, \hat{\omega}_{i,l}\right)\right. \\ & \left. + \gamma^{N_p}\hat{V}_{i,j-1}\left(x_{N_p+t}, \hat{u}_{i,N_p+t}, \hat{\omega}_{i,N_p+t}\right)\right), \end{aligned} \quad (43)$$

and

$$\begin{aligned} \hat{V}_{i,j,t}^o = {} & \sum_{l=t}^{t+N_p-1}\gamma^{l-t}U\left(x_l, \hat{u}_{i,l}, \hat{\omega}_{i,l}\right) \\ & + \gamma^{N_p}\hat{V}_{i,j-1}\left(x_{N_p+t}, \hat{u}_{i,N_p+t}, \hat{\omega}_{i,N_p+t}\right). \end{aligned} \quad (44)$$

It is worth noting that $\hat{u}_{i,t+1}$ and $\hat{\omega}_{i,t+1}$ represent the control policy and disturbance policies corresponding to x_{t+1} at the outer loop number i.

By applying the backpropagating technique, the update rule of the weight matrix $\widehat{W}_{i,j}^c$ can be denoted as

$$\begin{aligned} & \widehat{W}_{i,j+1}^c = \widehat{W}_{i,j}^c + \Delta\widehat{W}_{i,j}^c \\ & = \widehat{W}_{i,j}^c - \alpha_c\frac{\partial E_{i,j}^c}{\partial e_{i,j}^c}\frac{\partial e_{i,j}^c}{\partial \hat{V}_{i,j}\left(x_t, \hat{u}_{i,t}, \hat{\omega}_{i,t}\right)}\frac{\partial \hat{V}_{i,j}\left(x_t, \hat{u}_{i,t}, \hat{\omega}_{i,t}\right)}{\partial \widehat{W}_{i,j}^c} \\ & = \widehat{W}_{i,j}^c - \alpha_c e_{i,j}^c\sigma_c\left(Y_C^T\left(x_t, \hat{u}_{i,t}, \hat{\omega}_{i,t}\right)\right) \end{aligned} \quad (45)$$

where $0 < \alpha_c \le 1$ represents the learning rate of the critic network.

B. Implementation of the Actor Network

The actor network is introduced to obtain the optimal control policy. The output of the actor network can be represented by

$$\hat{u}_{i,t} = \zeta\left(\widehat{W}_i^{a\,T}\sigma_a\left(Y_a^T(x_t)\right)\right). \quad (46)$$

where $\widehat{W}_i^a \in \mathbb{R}^{m\times \hbar_a}$ and Y_a represent the weight matrix of the actor network, and $\hbar_a$ denotes the number of neurons on the hidden layer. Similar as before, a nonlinear activation function $\sigma_a(\cdot)$ is applied for hidden neurons of the actor network, with $\left\|\sigma_a\left(Y_a^T(x_t)\right)\right\| \le \sigma_{aM}$, where σ_{aM} is a positive constant. In addition, to guarantee saturated constraints of actuators, the nonlinear activation function $\zeta(\cdot)$ is used for the output layer of the actor network, with $\left|\zeta\left(\widehat{W}_i^{a\,T}\sigma_a\left(Y_a^T(x_t)\right)\right)\right| \le \bar{U}$, where $\bar{U} = [\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_n]^T$.

To obtain the updated policy, the traditional PIM process with the gradient descent method is no longer suitable for improving the policy advantage under state constraints. For simplicity, Eq. (29) can be rewritten as

$$\begin{aligned} & \min_u\ g^T\Delta\widehat{W}^a \\ & \text{s.t. } c + b^T\Delta\widehat{W}^a \le 0 \\ & \frac{1}{2}\left(\Delta\widehat{W}^a\right)^T H\left(\Delta\widehat{W}^a\right) < \delta. \end{aligned} \quad (47)$$

where $\Delta\widehat{W}^a = \widehat{W}^a - \widehat{W}_i^a$, and

$$\begin{aligned} g = {} & \widehat{W}^{c\,T}\otimes\left(1 - \tanh\left(Y_C^T\left(x_t, \hat{u}_{i,t}, \hat{\omega}_{i,t}\right)\right)^2\right) \\ & \cdot Y_c(m:m+n,:), \end{aligned} \quad (48)$$

$$b = \left(\nabla_{u_t}J_c\left(x_{c,t}^{N_c}\right)|_{u_t=u_{i,t}}\right)^T \quad (49)$$

$$c = J_c\left(x_{c,t}^{N_c}\right)|_{u_t=u_{i,t}}, \quad (50)$$

$$H = \nabla_{u_t}^2 D\left(u_t, u_{i,t}\right)|_{u_t=u_{i,t}} = \frac{\partial^2 D\left(u_t, u_{i,t}\right)}{\partial\widehat{W}_{i,p}^a\partial\widehat{W}_{i,q}^a}|_{u_t=u_{i,t}}. \quad (51)$$

where both $\widehat{W}_{i,p}^a$ and $\widehat{W}_{i,q}^a$ represent a specific weight element of actor network at i-th iteration. p and q denote specific positions in the weight matrix $\widehat{W}^a$. Please note that the Fisher matrix H is positive-definite, $c, \delta \in \mathbb{R}$, and $\delta > 0$.

In (47), the variable that needs to be optimized is $\Delta\widehat{W}^a$. The network parameter of the iterative control policy can be represented by

$$\widehat{W}_{i+1}^a = \widehat{W}_i^a + \Delta\widehat{W}_i^a. \quad (52)$$

We employ the Lagrange multiplier method to solve the convex optimization problem in (47), First, the Lagrange function can be constructed as

$$\begin{aligned} L\left(\Delta\widehat{W}^a, \lambda_1, \lambda_2\right) = {} & g^T\Delta\widehat{W}^a + \lambda_1\left(c + b^T\Delta\widehat{W}^a\right) \\ & + \lambda_2\left(\frac{1}{2}\left(\Delta\widehat{W}^a\right)^T H\left(\Delta\widehat{W}^a\right) - \delta\right), \end{aligned} \quad (53)$$

where λ_1 and λ_2 are dual variables. Then, the dual function can be constructed as

$$\begin{aligned} g(\lambda_1, \lambda_2) = \inf_{x\in C_s}\left(\right. & g^T\Delta\widehat{W}^a + \lambda_1\left(c + b^T\Delta\widehat{W}^a\right) \\ & \left. + \lambda_2\left(\frac{1}{2}\left(\Delta\widehat{W}^a\right)^T H\left(\Delta\widehat{W}^a\right) - \delta\right)\right). \end{aligned} \quad (54)$$

Subsequently, the dual problem corresponding original optimization problem (47) can be represented as

$$\max_{\lambda_1\ge 0, \lambda_2\ge 0}\ \min_{\Delta\widehat{W}^a} L\left(\Delta\widehat{W}^a, \lambda_1, \lambda_2\right) \quad (55)$$

Then, deriving the gradient information of $L\left(\Delta\widehat{W}^a, \lambda_1, \lambda_2\right)$ with

Algorithm 1: SADP Algorithm

Initialization:

- Initialize an initial admissible control law with $\widehat{W}^a$ and arbitrary $\widehat{W}^\omega$, $\widehat{W}^c$, and $\widehat{W}^s$;
- Given arbitrarily small positive constant $\Delta V_\epsilon, \epsilon_J$;
- Given max iteration j_N and i_N;
- Given the learning rates α_c, α_ω, and α_s;
- Initialize the system dataset $\mathcal{D}_M$ in Ω;

Iteration:

1: For $i = 1$ to i_N do
2: Choose randomly a set of initial states x_t in Ω.
3: Rollout N_c steps from x_t with control policy $u_{i,t}$
4: Update constrained cost function $\hat{J}_c(x_{c,t}^{N_c})$ with (62)
5: PEV:
Compute $\hat{V}_{i,j,t}^o$ and $e_{i,j}^c$ with (44) and (43)
Update the cost function $\hat{V}_{i,j+1}(x_t, \hat{u}_{i,t}, \hat{\omega}_{i,t})$ with (45)
If $|\hat{V}_{i,j+1}(x_t, u_{i,t}, \omega_{i,t}) - \hat{V}_{i,j}(x_t, u_{i,t}, \omega_{i,t})| < \epsilon_J$
break
6: PIM:
Update control policy u_{i+1} with (55)
Update disturbance policy ω_{i+1} with (58)
7: if $|\hat{V}_{i+1}(x_t, u_{i+1,t}, \omega_{i+1,t}) - \hat{V}_i(x_t, u_{i,t}, \omega_{i,t})| < \Delta V_\epsilon$
break
8: $i = i + 1$
9: return $\widehat{W}^{a,*}$, $\widehat{W}^{c,*}$ and $\widehat{W}^{\omega,*}$.

respect to $\Delta\widehat{W}^a$, it can be expressed as

$$\nabla_{\Delta\widehat{W}^a} L(\Delta\widehat{W}^a, \lambda_1, \lambda_2) = g + b\lambda_1 + \lambda_2 H \Delta\widehat{W}^a \tag{56}$$

Let $\nabla_{\Delta\widehat{W}^a} L(\Delta\widehat{W}^a, \lambda_1, \lambda_2) = 0$, and then $\Delta\widehat{W}^a$ can be derived as

$$\Delta\widehat{W}^a = -\frac{1}{\lambda_2} H^{-1}(g + \lambda_1 b) \tag{57}$$

Then, substitute (57) into (55), the dual problem can be reconstructed as

$$\max_{\lambda_1 \geq 0, \lambda_2 \geq 0} -\frac{1}{2\lambda_2}(q + \lambda_1^T s \lambda_1 + 2\lambda_1^T r) - \lambda_2 \delta + \lambda_1^T c \tag{58}$$

where $q = g^T H^{-1} g$, $r = g^T H^{-1} b$, and $s = b^T H^{-1} b$. Subsequently, we have

$$\Delta\widehat{W}^{a,*} = -\frac{1}{\lambda_2^*} H^{-1}(g + \lambda_1^* b) \tag{59}$$

where λ_1^* and λ_2^* are the optimal analytical solutions of dual problem (55).

Remark 7: Because of the inherent approximation errors stemming from linearization, it is possible that the optimal solution of (55) cannot yield an entirely suitable update This could potentially lead to the generation of a new policy that doesn't conform to state constraints. As a consequence, the solution of optimization problem (47) in the subsequent iterations is infeasible. In essence, the feasible region of (47) turns out to be non-existent. In this case, a certain extent relaxation should be implemented [29], [31].

C. Implementation of the Disturbance Network

The disturbance network is used to estimate a near-worst disturbance policy ω_t^*. The iterative disturbance policy $\hat{\omega}_{i,t}$ can be represented as

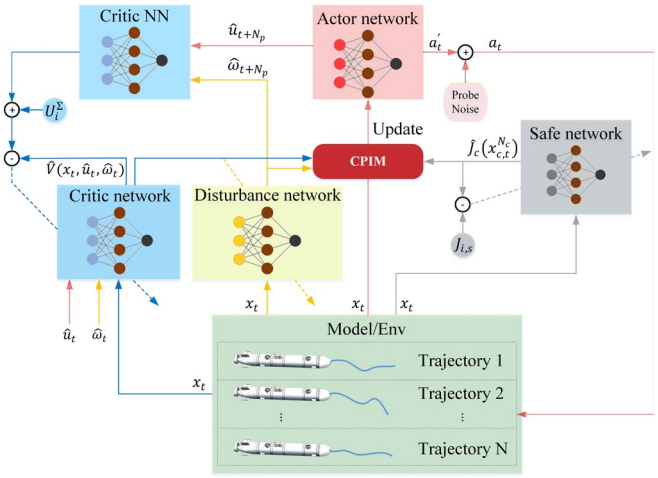

Fig. 1. The architecture of the proposed SADP scheme considering disturbances.

$$\hat{\omega}_{i,t} = \widehat{W}_i^{\omega^T} \sigma_\omega(Y_\omega^T(x_t)). \tag{60}$$

where $\widehat{W}_i^\omega \in \mathbb{R}^{m \times \hbar_\omega}$ signifies the weight matrix connecting the hidden and output layers, specific to the iterative index i, while $\hbar_\omega$ denotes the number of neurons on the hidden layer. Y_ω represents the weight matrix between the hidden and input layers. Similarly, a nonlinear activation function $\sigma_\omega(\cdot)$ is implemented for the hidden layer, with $\|\sigma_\omega(Y_\omega^T(x_t))\| \leq \sigma_{\omega M}$, where $\sigma_{\omega M}$ is a positive constant.

To obtain an optimal control policy under worst disturbance, the policy gradient can be used. Then, the weight update rule of the disturbance network can be expressed by

$$\begin{aligned} \widehat{W}_{i+1}^\omega &= \widehat{W}_i^\omega + \Delta\widehat{W}_i^\omega \\ &= \widehat{W}_i^\omega - \alpha_\omega \left(-\frac{\partial \hat{V}_{i,j}(x_t, \hat{u}_{i,t}, \hat{\omega}_{i,t})}{\partial \hat{\omega}_{i,t}} \frac{\partial \hat{\omega}_{i,t}}{\partial \widehat{W}_i^\omega} \right) \\ &= \widehat{W}_i^\omega + \alpha_\omega \Bigg(\widehat{W}_i^{c^T} \otimes \Big(1 \\ &\qquad - \sigma_{c,i}^2 \left(Y_{c,\hat{\omega}}^T(x_t, \hat{u}_{i,t}, \hat{\omega}_{i,t}) \right) \Big) \\ &\qquad \cdot \sigma_\omega(Y_\omega^T(x_t)) \Bigg) \end{aligned} \tag{61}$$

Where $Y_{c,\hat{\omega}}^T(x_t, \hat{u}_{i,t}, \hat{\omega}_{i,t})$ represents $Y_{c,m+n:m+2n}^T(x_t, \hat{u}_{i,t}, \hat{\omega}_{i,t})$. $0 < \alpha_\omega \leq 1$ represents the learning rate of the disturbance network.

D. Implementation of the Constrained Cost Network

We further introduce a constrained cost network to approximate $J_c(x_{c,t}^{N_c})$. The approximated cost, denoted by $\hat{J}_c(x_{c,t}^{N_c})$, can be represented as

$$\hat{J}_c(x_{c,t}^{N_c}) = \widehat{W}_p^{s^T} \sigma_s(Y_s^T(x_t, \hat{u}_t, \hat{\omega}_t)). \tag{62}$$

where $\widehat{W}_p^s \in \mathbb{R}^{(m+2n) \times \hbar_s}$ and Y_s represent the weight matrix, and $\hbar_s$ denotes the number of neurons on the hidden layer. p denotes the iteration for estimating the constrained cost function. A nonlinear activation function $\sigma_s(\cdot)$ is implemented with $\|\sigma_s(Y_s^T(x_t, \hat{u}_t, \hat{\omega}_t))\| \leq \sigma_{sM}$, where σ_{sM} is a positive constant.

Similarly, the evaluation error can be given as

$$E_p^s = \frac{1}{2}\left\|e_p^s\right\|^2, \tag{63}$$

where

$$e_p^s = \hat{J}_c\left(x_{c,t}^{N_c}\right) - \sum_{l=t}^{t+N_c-1} \gamma_c^{l-t} h\left(x_{c,l}\right). \tag{64}$$

Then, the update rule of the weight matrix $\widehat{W}_p^s$ is set to be

$$\begin{aligned} \widehat{W}_p^s &= \widehat{W}_p^s + \Delta\widehat{W}_p^s \\ &= \widehat{W}_p^s - \alpha_s \frac{\partial E_p^s}{\partial e_p^s}\frac{\partial e_p^s}{\partial \hat{J}_c\left(x_{c,t}^{N_c}\right)}\frac{\partial \hat{J}_c\left(x_{c,t}^{N_c}\right)}{\partial \widehat{W}_p^s} \\ &= \widehat{W}_p^s - \alpha_s e_p^s \sigma_s\left(Y_s^T(x_t)\right). \end{aligned} \tag{65}$$

where $0 < \alpha_s \leq 1$ represents the learning rate of the constrained cost network.

E. SADP Algorithm

The proposed SADP scheme can be used to solve the optimal control problem of discrete-time systems subject to safe constraints and disturbances, as shown in Fig. 1, and the pseudocode of the proposed scheme is given in **Algorithm 1**.

Remark 8: The data utilization method is critical for searching for the optimal control policy. In this study, the selection of N_p and N_c should be considered carefully. To achieve a satisfactory far-sighted effort, N_c can be chosen as an integer that is greater than 1. i.e., $N_c > 1$. For convenience, N_p can be chosen to be equivalent to N_c. Moreover, if $N_p \to \infty$, the Temporal Difference (TD) learning framework will be converted into Monte Carlo (MC) method. In fact, zero-bias property of the MC scheme can provide better convergence during the training process. Meanwhile, to decrease the variance, which is evident in MC, the data can be generated independently and in parallel with a considerable number of different initial states. Additionally, inspired by [30], a data buffer $\mathcal{D}_M$ is introduced to store data transitions, i.e., $\mathcal{D}_M = \{(x_t, u_t, \omega_t, U_t, x_{t+1}), x_t \in \Omega, u_t \in \mathcal{U}_x, \omega_t \in \Psi_\omega\}$, where the number of transitions in dataset is denoted as M.

Remark 9: In Section-III-B, the rigorous convergence of the proposed SADP is analyzed, which means the proposed scheme is feasible to obtain the optimal solution. The application of the NN technique in this context provides a specific implementation. Although there exists approximation error with this method, the over-parameterized NN structure, not limited to the structure shown in this paper, can ensure sufficient precision in approximating the iterative function.

Remark 10: Notably, the proposed method can be implemented in two different ways depending on the availability of accurate model information. In scenarios where accurate model information is accessible, the algorithm can effectively utilize it to derive the future state of the control object, resulting in improved control performance and safety. On the other hand, in cases where model information is not available or cannot be established accurately, the algorithm adopts a data-driven approach, allowing the control object to interact with the environment and collect data to learn an optimal control law. This data-driven strategy liberates the algorithm from depending on an accurate model and facilitates adaptability to real-world conditions. In contrast, the conventional CBF approach necessitates accurate model information to design safe constraints, making it vulnerable to suboptimal performance and safety concerns when faced with inaccuracies in model knowledge.

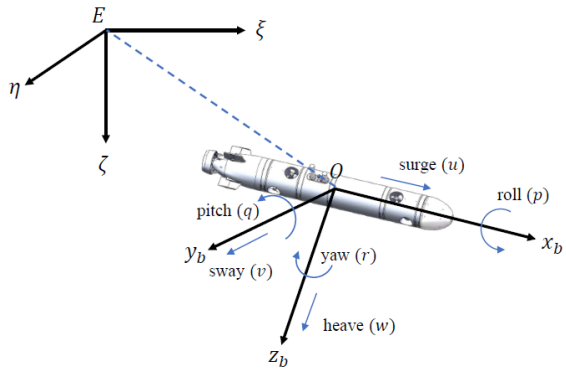

Fig. 2. 6 DOF AUV coordinate systems

Table 1
HYDRODYNAMIC PARAMETERS OF THE AUV

Parameters	Symbol	Value
rotational inertia about z-axis	I_{zz}	11.61
hydrodynamic added masses	$N_{\dot{r}}$	-53.87
nonlinear damping force coefficients	N_r	-13.66
cross flow resistance coefficients	$N_{r\|r\|}$	-10.37
rudder lifting moment coefficients	N_δ	7.103
Disturbance coefficients	N_ω	4.58
weight of the AUV (N)	W	488.84
velocity of surge (m/s)	u_x	1.0
Disturbance	ω	$N(0,0.001)$

V. SIMULATION

In this section, a simulation example is presented to validate the effectiveness and control performance of the proposed SADP algorithm. Specifically, the presented example is related to the course motion, which is quite common in the AUVs' community. It is worth noting that AUVs, as a type of self-executing unmanned underwater vehicles, have recently achieved widespread applications in various fields, such as resource exploration, search and rescue, commercial applications, pipeline inspection, and so on. They are progressively evolving towards more intelligent and digitalized directions in deeper and farther environments [53].

In typical scenarios, an AUV can be treated as a symmetric rigid body, and its operating speed falls within a low range. Consequently, the course dynamics system of an AUV can be decoupled from the 6 DOF model [54], as shown in Fig. 2. The system state variables are the yaw angle ψ and angular velocity r, i.e., $x = [\psi, r]^T$, and the control input variable is denoted as $u = \delta_r$. Therefore, the course dynamics system can be expressed as

$$\begin{cases} \dot{\psi} = r \\ (I_{zz} - N_{\dot{r}})\dot{r} = N_r u_x r + N_\delta u_x^2 \delta_r + N_{r|r|} r|r| + N_\omega \omega \end{cases}, \tag{66}$$

where u_x represents the forward velocity of the AUV. The remaining hydrodynamic parameters are shown in Table 1,

according to the handbook of the ELFIN E200 AUV.

The Euler and trapezoidal approach is adopted to discretize the original continuous-time system (66) with the discrete time $\Delta_t = 0.2s$. As a result, the discrete-time course dynamics model can be given as

$$\begin{bmatrix} x_{1(t+1)} \\ x_{2(t+1)} \end{bmatrix} = \begin{bmatrix} 0.2x_{2t} + x_{1t} \\ 0.958x_{2t} - 0.0316x_{2t}|x_{2t}| \end{bmatrix} + \begin{bmatrix} 0 \\ 0.0216 \end{bmatrix} u_t + \begin{bmatrix} 0 \\ 0.01 \end{bmatrix} \omega_t, \quad (67)$$

where $x_{1t} = \psi$ and $x_{2t} = r$. For the utility function in (3), we set $Q = \begin{bmatrix} 0.5 & 0 \\ 0 & 1.0 \end{bmatrix}$, $R = 1.0$, and $\beta = 0.014$.

In most cases, the state should be limited within a proper range. Otherwise, the dynamics model may become inaccurate because the hydrodynamic parameters will change dynamically under different operating conditions, and the components installed in the AUV should also operate within stable working conditions. Therefore, to ensure that the AUV operates properly and maintains an accurate dynamics model, the state is typically bounded by $|x_t| \leq 0.348 = 20^\circ$. Similarly, based on the input saturation characteristics of the actuators in AUVs, the control input should be constrained within $|u_t| \leq 0.79 = 45^\circ$.

Three 3-layer neural networks (NNs) are employed to approximate the optimal cost function, control input, and disturbance policy, with structures of 4-16-1, 2-16-1, and 2-16-1. The activation functions used in the three NNs vary, depending on the specific training objectives and characteristics. For the critic NN, the activation functions of the hidden layer and output layer are both ReLU functions. As for the actor and disturbance NNs, the activation functions of the hidden layer and output layer are ReLU and tanh functions, respectively. The learning rate α_c and α_ω are both set as 5×10^{-3}, and the Adam update criterion is applied to achieve their respective approximated objectives. Moreover, to ensure the iteration feasibility, the convergence threshold ϵ_J and ΔV_ϵ are both set to 10^{-6}. Additionally, as mentioned in Remark 8, data utilization and comprehensiveness are crucial; thus, in this study, the data buffer $\mathcal{D}_M$ is consists of 10^5 transitions generated by various initial states.

To verify the control performance and constraint effect, the initial state is chosen as $x_0 = [0.3, -0.3]^T$. Subsequently, to verify the constraint handling performance of the proposed algorithm, the iterative convergence trajectories of each PI step generated by SADP and the action-dependent heuristic dynamic programming (ADHDP [55]) are compared in Fig. 3. It should be noted that the forward steps executed via the dynamics model (67) comprise 300 time steps. In Fig. 3 (a), we can find that all states and control inputs are within the required state and action limits. However, in Fig. 3 (c), the system state clearly exceeds the limit, although the control policy remains admissible. Fig. 4 displays the behavior of the final disturbance policy applied to (67) after iterations. Notably, both the control policy and disturbance policy converge to the original point, indicating that the saddle point solutions u_t^* and ω_t^* are ultimately achieved. Additionally, the optimal cost function is displayed in Fig. 5. To validate the approximation accuracy of the optimal cost function, the real cost function with optimal

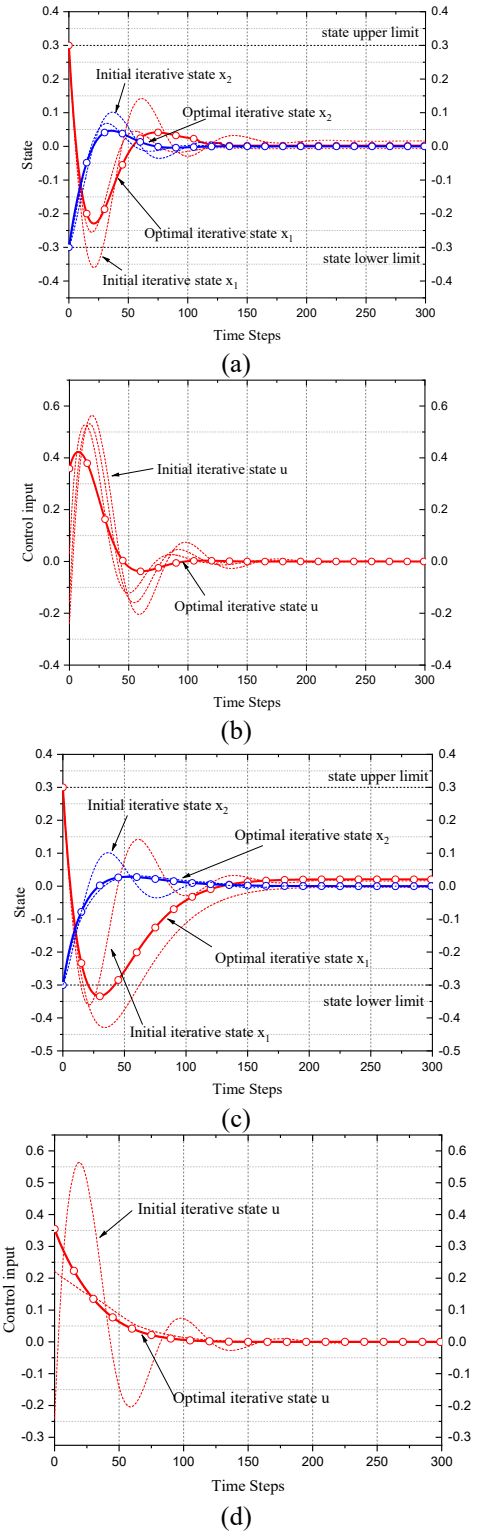

Fig. 3. Simulation results. (a) Iterative states and optimal states under SADP. (b) Iterative control inputs and optimal control inputs under SADP. (c) Iterative states and optimal states under ADHDP. (d) Iterative control inputs and optimal control inputs under ADHDP.

control and disturbance policies are shown as a red line. Through this comparison, it can be concluded that the approximated cost function is nearly identical to the optimal cost function. Therefore, the effectiveness of the SADP algorithm is verified based on the comprehensive analyses

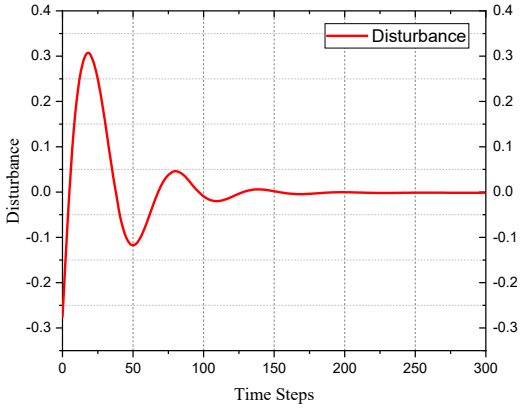

Fig. 4. Convergent behaviors of disturbance policy with SADP

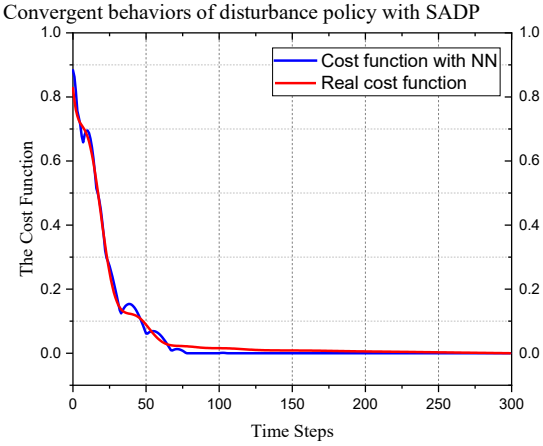

Fig. 5. Convergent behaviors of the optimal cost function with SADP

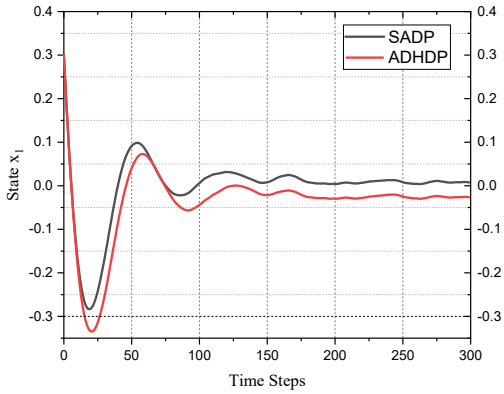

Fig. 6. The convergence behavior of state x_1 under random disturbances

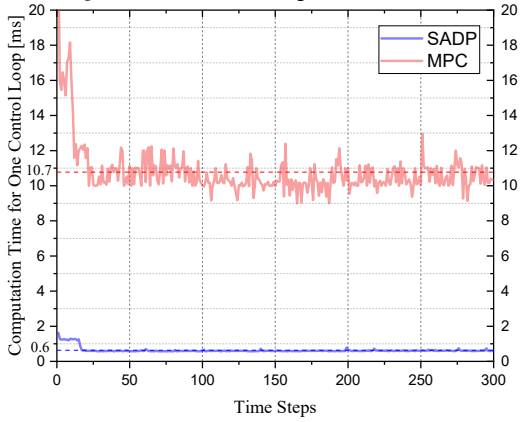

Fig. 7. Computation times for one control loop of SADP and MPC

Table 2
ITERATION TIME FOR EACH PI OF SADP AND MPC [s]

Algorithm	1st	2nd	3rd	4th
SADP	32.69	33.52	37.33	40.92
Model-based ADHDP	68.46	70.02	73.31	63.78

presented above.

The disturbance shown in Fig. 4 ultimately converges to the saddle point. However, the disturbance encountered in real-world systems and environments tends to exhibit a stochastic nature. Therefore, in order to validate the robustness and safety of the proposed SADP more effectively, a random disturbance within the range of $(-0.05, 0.05)$ is implemented in the proposed SADP and the traditional game-based ADHDP. As shown in Fig. 6, compared with the final unsafe performance obtained by the traditional game-based ADHDP, the system state can successfully converge into the safe region by adopting the proposed algorithm. This result signifies that the behavior of control object deviating from the safe bound due to the presence of disturbances is effectively constrained during the search for the optimal control policy.

During the training process, the implementation of model information can facilitate the data utilization, thereby enhancing the convergence performance; however, this does not mean that the model-based version can achieve a faster convergence. This is primarily attributed to the process of solving partial derivative cross dynamics model with backpropagation mechanism is extremely time-consuming. Evidently shown in Table 2, the time consumption within each PI loop reveals a noteworthy discrepancy, with the model-based iteration taking nearly two times longer than the SADP scheme. In fact, this divergence would become particularly significant as the system complexity increases.

In the control community, MPC stands as a potent solution for grappling with nonlinear optimal problems subject to constraints. In Fig. 7, a comparison of the computation time across one control loop based on the proposed SADP and MPC algorithms is depicted. Notably, the average calculation time for the SADP algorithm is 0.6 ms, while the MPC controller necessitates almost 10.7 ms. This striking contrast underscores the exceptional computational efficiency of the SADP algorithm, which outpaces the MPC algorithm by nearly 17.8 times. This compelling distinction not only underscores the computational prowess of the proposed method but also highlights its ability to operate within millisecond-level timeframes, thereby offering real-time solutions in practical applications.

VI. CONCLUSION

In this study, a novel safe ADP is proposed to solve constrained optimal control problems considering disturbances. First, the original PIM process is transformed into a constrained optimization problem, in which only a single inequation representing aggregated state constraints is constructed, reducing the computational burden and improving the feasibility of the scheme. Apart from addressing state and control input constraints from practical perspectives, the

problem caused by disturbances is also considered in constrained optimal control problems. To this end, the robust safety against disturbances is treated as a two-player zero-sum game, where the optimal control policy and worst disturbance policy are approximated by two NNs, respectively. Furthermore, from a theoretical perspective, the convergence properties of the SADP algorithm are analyzed. Finally, simulation results demonstrate the excellent safety and control performance based on a course motion system of AUVs.

In the future, we may focus on developing a model-free off-policy SADP scheme that can be applied to optimal tracking control problems considering disturbances.

REFERENCES

[1] I.-L. G. Borlaug, K. Y. Pettersen, and J. T. Gravdahl, 'Comparison of two second-order sliding mode control algorithms for an articulated intervention AUV: Theory and experimental results', *Ocean Eng.*, vol. 222, p. 108480, Feb. 2021.

[2] Y. Bian, J. Zhang, M. Hu, C. Du, Q. Cui, and R. Ding, 'Self-triggered distributed model predictive control for cooperative diving of multi-AUV system', *Ocean Eng.*, vol. 267, p. 113262, Jan. 2023.

[3] M. Hu, C. Li, Y. Bian, H. Zhang, Z. Qin, and B. Xu, 'Fuel Economy-Oriented Vehicle Platoon Control Using Economic Model Predictive Control', *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 20836–20849, Nov. 2022.

[4] Y. Yang, B. Kiumarsi, H. Modares, and C. Xu, 'Model-Free λ-Policy Iteration for Discrete-Time Linear Quadratic Regulation', *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 2, pp. 635–649, Aug. 2021.

[5] H. Dong and X. Zhao, 'Wind-Farm Power Tracking Via Preview-Based Robust Reinforcement Learning', *IEEE Trans. Ind. Inform.*, vol. 18, no. 3, pp. 1706–1715, Mar. 2022.

[6] H. Dong and X. Zhao, 'Composite Experience Replay-Based Deep Reinforcement Learning with Application in Wind Farm Control', *IEEE Trans. Control Syst. Technol.*, vol. 30, no. 3, pp. 1281–1295, May 2022.

[7] Y. Bian, S. E. Li, W. Ren, J. Wang, K. Li, and H. X. Liu, 'Cooperation of Multiple Connected Vehicles at Unsignalized Intersections: Distributed Observation, Optimization, and Control', *IEEE Trans. Ind. Electron.*, vol. 67, no. 12, pp. 10744–10754, Dec. 2020.

[8] Y. Bian, C. Du, M. Hu, S. E. Li, H. Liu, and C. Li, 'Fuel Economy Optimization for Platooning Vehicle Swarms via Distributed Economic Model Predictive Control', *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 4, pp. 2711–2723, Oct. 2022.

[9] X. Bu, B. Jiang, and H. Lei, 'Low-Complexity Fuzzy Neural Control of Constrained Waverider Vehicles via Fragility-Free Prescribed Performance Approach', *IEEE Trans. Fuzzy Syst.*, vol. 31, no. 7, pp. 2127–2139, Jul. 2023.

[10] X. Bu, Q. Qi, and B. Jiang, 'A Simplified Finite-Time Fuzzy Neural Controller With Prescribed Performance Applied to Waverider Aircraft', *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 7, pp. 2529–2537, Jul. 2022.

[11] X. Bu, Y. Xiao, and H. Lei, 'An Adaptive Critic Design-Based Fuzzy Neural Controller for Hypersonic Vehicles: Predefined Behavioral Nonaffine Control', *IEEEASME Trans. Mechatron.*, vol. 24, no. 4, pp. 1871–1881, Aug. 2019.

[12] X. Bu and Q. Qi, 'Fuzzy Optimal Tracking Control of Hypersonic Flight Vehicles via Single-Network Adaptive Critic Design', *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 1, pp. 270–278, Jan. 2022.

[13] P. J. Werbos, 'Advanced forecasting methods for global crisis warning and models of intelligence', *Gen. Syst.*, vol. 22, pp. 25–38, Jun. 1977.

[14] D. Liu, S. Xue, B. Zhao, B. Luo, and Q. Wei, 'Adaptive Dynamic Programming for Control: A Survey and Recent Advances', *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 51, no. 1, pp. 142–160, Dec. 2020.

[15] H. Dong, X. Zhao, and B. Luo, 'Optimal Tracking Control for Uncertain Nonlinear Systems with Prescribed Performance via Critic-Only ADP', *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 52, no. 1, pp. 561–573, Jul. 2020.

[16] S. Song, M. Zhu, X. Dai, and D. Gong, 'Model-Free Optimal Tracking Control of Nonlinear Input-Affine Discrete-Time Systems via an Iterative Deterministic Q-Learning Algorithm', *IEEE Trans. Neural Netw. Learn. Syst.*, Jun. 2022.

[17] J. Ye, Y. Bian, B. Luo, M. Hu, B. Xu, and R. Ding, 'Costate-Supplement ADP for Model-Free Optimal Control of Discrete-Time Nonlinear Systems', *IEEE Trans. Neural Netw. Learn. Syst.*, May 2022.

[18] C. Shen and Y. Shi, 'Distributed implementation of nonlinear model predictive control for AUV trajectory tracking', *Automatica*, vol. 115, p. 108863, May 2020.

[19] Z. Lin, J. Duan, S. E. Li, H. Ma, Y. Yin, and B. Cheng, 'Continuous-time finite-horizon ADP for automated vehicle controller design with high efficiency', in *2020 3rd International Conference on Unmanned Systems (ICUS)*, Nov. 2020, pp. 978–984.

[20] M. Brown, J. Funke, S. Erlien, and J. C. Gerdes, 'Safe driving envelopes for path tracking in autonomous vehicles', *Control Eng. Pract.*, vol. 61, pp. 307–316, Apr. 2017.

[21] H. Zhang, Y. Luo, and D. Liu, 'Neural-Network-Based Near-Optimal Control for a Class of Discrete-Time Affine Nonlinear Systems with Control Constraints', *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1490–1503, Sep. 2009.

[22] S. Xue, B. Luo, D. Liu, and Y. Gao, 'Event-Triggered ADP for Tracking Control of Partially Unknown Constrained Uncertain Systems', *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 9001–9012, Mar. 2021.

[23] S. Lyashevskiy, 'Constrained optimization and control of nonlinear systems: new results in optimal control', in *Proceedings of 35th IEEE Conference on Decision and Control*, Dec. 1996, pp. 541–546.

[24] S. Zhao, J. Wang, H. Wang, and H. Xu, 'Goal representation adaptive critic design for discrete-time uncertain systems subjected to input constraints: The event-triggered case', *Neurocomputing*, vol. 492, pp. 676–688, Jul. 2022.

[25] Z. Marvi and B. Kiumarsi, 'Safe reinforcement learning: A control barrier function optimization approach', *Int. J. Robust Nonlinear Control*, vol. 31, no. 6, pp. 1923–1940, Jun. 2021.

[26] M. Mazouchi, S. Nageshrao, and H. Modares, 'Conflict-Aware Safe Reinforcement Learning: A Meta-Cognitive Learning Framework', *IEEECAA J. Autom. Sin.*, vol. 9, no. 3, pp. 466–481, Mar. 2021.

[27] J. Na, B. Wang, G. Li, S. Zhan, and W. He, 'Nonlinear Constrained Optimal Control of Wave Energy Converters with Adaptive Dynamic Programming', *IEEE Trans. Ind. Electron.*, vol. 66, no. 10, pp. 7904–7915, Oct. 2019.

[28] J. Garcıa and F. Fernandez, 'A Comprehensive Survey on Safe Reinforcement Learning', *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 1437–1480, Aug. 2015.

[29] J. Achiam, D. Held, A. Tamar, and P. Abbeel, 'Constrained Policy Optimization'. arXiv, May 2017.

[30] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, 'Trust Region Policy Optimization', in *Proceedings of the 32nd International Conference on Machine Learning*, PMLR, Jun. 2015, pp. 1889–1897.

[31] 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, May 2022.

[32] H. Wang, C. Hu, J. Zhou, L. Feng, B. Ye, and Y. Lu, 'Path tracking control of an autonomous vehicle with model-free adaptive dynamic programming and RBF neural network disturbance compensation', *Proc. Inst. Mech. Eng. Part J. Automob. Eng.*, vol. 236, no. 5, pp. 825–841, Jul. 2021.

[33] G. Che and Z. Yu, 'Neural-network estimators based fault-tolerant tracking control for AUV via ADP with rudders faults and ocean current disturbance', *Neurocomputing*, vol. 411, pp. 442–454, Oct. 2020.

[34] K. Duan, S. Fong, and C. L. P. Chen, 'Reinforcement learning based model-free optimized trajectory tracking strategy design for an AUV', *Neurocomputing*, vol. 469, pp. 289–297, Jan. 2022.

[35] I. Carlucho, M. De Paula, S. Wang, Y. Petillot, and G. G. Acosta, 'Adaptive low-level control of autonomous underwater vehicles using deep reinforcement learning', *Robot. Auton. Syst.*, vol. 107, pp. 71–86, Sep. 2018.

[36] B. Luo, Y. Yang, and D. Liu, 'Policy Iteration Q-Learning for Data-Based Two-Player Zero-Sum Game of Linear Discrete-Time Systems', *IEEE Trans. Cybern.*, vol. 51, no. 7, pp. 3630–3640, Feb. 2020.

[37] X. Shan, 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, Jun. 2020.

[38] B. Kiumarsi, F. L. Lewis, and Z.-P. Jiang, 'H∞ control of linear discrete-time systems: Off-policy reinforcement learning', *Automatica*, vol. 78, pp. 144–152, Apr. 2017.

[39] X. Yang and H. He, 'Event-Driven H∞-Constrained Control Using Adaptive Critic Learning', *IEEE Trans. Cybern.*, vol. 51, no. 10, pp. 4860–4872, Oct. 2021.
[40] D. Liu and Q. Wei, 'Policy Iteration Adaptive Dynamic Programming Algorithm for Discrete-Time Nonlinear Systems', *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 621–634, Mar. 2014.
[41] D. Liu, Q. Wei, and P. Yan, 'Generalized Policy Iteration Adaptive Dynamic Programming for Discrete-Time Nonlinear Systems', *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 45, no. 12, pp. 1577–1591, Dec. 2015.
[42] M. Hu *et al.*, 'Coordinated collision avoidance for connected vehicles using relative kinetic energy density', *Int. J. Automot. Technol.*, vol. 18, no. 5, pp. 923–932, Oct. 2017.
[43] H. Dong, Q. Hu, and M. R. Akella, 'Safety Control for Spacecraft Autonomous Rendezvous and Docking Under Motion Constraints', *J. Guid. Control Dyn.*, vol. 40, no. 7, pp. 1680–1692, Apr. 2017.
[44] L. Brunke *et al.*, 'Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning', *Annu. Rev. Control Robot. Auton. Syst.*, vol. 5, no. 1, pp. 411–444, 2022.
[45] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, 'Control Barrier Function Based Quadratic Programs for Safety Critical Systems', *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017.
[46] Y. Han and H. Modares, 'A Satisficing Control Design Framework with Safety and Performance Guarantees for Constrained Systems under Disturbances'. arXiv, Sep. 2020.
[47] H. Ma *et al.*, 'Model-based Constrained Reinforcement Learning using Generalized Control Barrier Function', in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2021, pp. 4552–4559.
[48] N. M. Yazdani, R. K. Moghaddam, B. Kiumarsi, and H. Modares, 'A Safety-Certified Policy Iteration Algorithm for Control of Constrained Nonlinear Systems', *IEEE Control Syst. Lett.*, vol. 4, no. 3, pp. 686–691, Jul. 2020.
[49] Asma Al-Tamimi, Frank L. Lewis, and Murad Abu-Khalaf, 'Discrete-Time Nonlinear HJB Solution Using Approximate Dynamic Programming: Convergence Proof', *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 38, no. 4, pp. 943–949, Aug. 2008.
[50] Y. Liu, H. Zhang, R. Yu, and Z. Xing, 'H∞ Tracking Control of Discrete-Time System with Delays via Data-Based Adaptive Dynamic Programming', *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 50, no. 11, pp. 4078–4085, Nov. 2020.
[51] Q. Lin, Q. Wei, and D. Liu, 'A novel optimal tracking control scheme for a class of discrete-time nonlinear systems using generalised policy iteration adaptive dynamic programming algorithm', *Int. J. Syst. Sci.*, vol. 48, no. 3, pp. 525–534, May 2016.
[52] S. N. Balakrishnan, J. Ding, and F. L. Lewis, 'Issues on Stability of ADP Feedback Controllers for Dynamical Systems', *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 38, no. 4, pp. 913–917, Aug. 2008.
[53] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.
[54] T. I. Fossen, *Marine control systems: guidance, navigation and control of ships, rigs and underwater vehicles*. Marine Cybernetics, 2002.
[55] P. J. Werbos, 'Approximate dynamic programming for real-time control and neural modeling', *Handb. Intell. Control Neural Fuzzy Adapt. Approaches*, pp. 493–525, Jan. 1992.

Jun Ye received the B.S. degree in school of automotive engineering from Wuhan University of Technology, Wuhan, China, in 2019. He is currently pursuing the Ph.D. degree in vehicle engineering with Hunan University, Changsha, China. His research interest includes intelligent control and decision, reinforcement learning, and adaptive dynamic programming.

Hongyang Dong received the Ph.D. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2018.,He is currently an Assistant Professor with the School of Engineering, University of Warwick, Coventry, U.K.. He was a Research Fellow in Machine Learning and Intelligent Control with the University of Warwick from 2019 to 2022, before he became an Assistant Professor in November 2022. His current research interest is control theories and machine learning methods with their applications in complex systems, including offshore renewable energy systems and autonomous systems.

Yougang Bian (Member, IEEE) received the B.E. and Ph.D. degrees from Tsinghua University, Beijing, China, in 2014 and 2019, respectively. He was a visiting scholar with the Department of Electrical and Computer Engineering, University of California at Riverside, from 2017 to 2018. He is currently an Associate Professor with the College of Mechanical and Vehicle Engineering and the State Key Laboratory of Advanced Design and Manufacturing Technology for Vehicle, Hunan University, Changsha, China. His research interests include distributed control, cooperative control, and their applications to connected and automated vehicles. He is a recipient of the Best Paper Award at the 2017 IEEE Intelligent Vehicles Symposium.

Hongmao Qin received the Ph.D. degree in vehicle operation engineering from Jiangsu University, Zhenjiang, China, in 2014.,From 2014 to 2016, he was a Postdoctoral Researcher with the Department of Automotive Engineering, Tsinghua University. From 2016 to 2019, he was an Assistant Professor with the School of Transportation Science and Engineering, Beihang University. He is currently a Research Scientist with the College of Mechanical and Vehicle Engineering, Hunan University, Changsha, China. His research interests include driving behavior models and vehicle cyber security.

Xiaowei Zhao (Member, IEEE) received the Ph.D. degree in control theory from Imperial College London, London, U.K., in 2010. He was a Postdoctoral Researcher with the University of Oxford, Oxford, U.K., for three years before joining the University of Warwick, Coventry, U.K., in 2013. He is Professor of control engineering and an EPSRC Fellow with the School of Engineering, University of Warwick. His main research interests include control theory and machine learning with applications in offshore renewable energy systems, smart grids, and autonomous systems.