






Letter

Policy Gradient Adaptive Dynamic Programming for Model-Free Multi-Objective Optimal Control

Hao Zhang , Yan Li , Zhuping Wang , Yi Ding , and Huaicheng Yan 

Dear Editor,

In this letter, the multi-objective optimal control problem of nonlinear discrete-time systems is investigated. A data-driven policy gradient algorithm is proposed in which the action-state value function is used to evaluate the policy. In the policy improvement process, the policy gradient based method is employed, which can improve the performance of the system and finally derive the optimal policy in the Pareto sense. The actor-critic structure is established to implement the algorithm. In order to improve the efficiency of data usage and enhance the learning effect, the experience replay technology is used during the training process, with both offline data and online data. Finally, simulation is given to illustrate the effectiveness of the method.

Introduction: The multi-objective optimal control problems have become a growing research field in recent years, due to its wide application in autonomous driving [1], smart grid [2] and other autonomous intelligent systems [3]. In some cases, the multi-objective optimal control problems can be converted to solve the Hamilton-Jacobi-Bellman equation (HJBE), which need accurate system model parameters. However, it is hard to find optimal controllers for systems without accurate models. At present, reinforcement learning as a model-free multi-objective optimal control method, which is widely used to learn policy from the process that interacts with the unknown environment.

In recent years, ADP is introduced in order to solve the problem that HJBE cannot solve directly. The generalized policy iteration algorithm was proposed by combining the policy iteration algorithm with the value iteration [4]. Under the framework of policy iterative algorithm, the policy gradient adaptive dynamic programming (PGADP) is an important policy-based method. It used the gradient descent in step of policy improvement [5]. In [6], the experience replay was used in combination with ADP, using the past and current data concurrently. In [7]–[9], the adaptive optimal controller was designed by the online actor-critic learning, in order to solve the robust optimal control problem for a class of nonlinear systems. In [10], a model-free λ -policy iteration (λ -PI) was presented for the discrete-time linear quadratic regulation (LQR) problem. However, the above results only consider the solution under a single goal. In engineering practice and scientific research, many problems need more performance indices to describe the goals of the system. In [11], the policy iteration algorithm was extended to solve dynamic multi-

objective optimal control problem for continuous-time systems. There are few results using policy gradient based methods with experience replay mechanism to solve multi-objective optimal control problems. It inspires the motivation to extend the related methods from single objective optimal control to multi-objective optimal control.

Thus, the objective of this letter is try to find the optimal controller in the sense of Pareto for a discrete-time system with multiple control objectives. The contributions of this letter can be summarized as follows. Firstly, the action-state value function Q instead of the state value function is used in multi-objective optimal control. The potential dynamic constraints can be separated from the actual controller parameters by using the action state value function. Secondly, the dependency on the model can be removed completely. The experience replay technique is incorporated into multi-objective optimal control problem, which improves data usage efficiency with fixed-size offline dataset and single-frame real-time data received from the environment. Third, the policy gradient method is extended from single-objective optimal control to multi-objective optimal control. This method can make the learning process smoother and reduce the amount of computation. The convergence of PGADP with multiple objectives is guaranteed in this letter.

Problem statement: First, the related concept of Pareto optimal is introduced.

Definition 1 (Pareto optimal): A Solution u^* is said to be a Pareto optimal solution if $J(u^*) \leq J(u)$ for all $u \in \mathcal{U}$. $J(u^*)$ is said to be Pareto optimal.

Consider the autonomous intelligent systems with following discrete-time general nonlinear form:

$$x_{k+1} = F(x_k, u_k) \quad (1)$$

where $m, n, k \in \mathbb{Z}_+$, \mathbb{R} and \mathbb{Z}_+ denote the set of real numbers and non-negative integers, respectively. $x_k \in \mathbb{R}^n$ and $u_k \in \mathbb{R}^m$ are the state and control input of the system, respectively. Assume that $F(x, u)$ is Lipschitz continuous. In a multi-objective optimization problem, infinite horizon performance indices $J_j(x_0, u) = \sum_{l=0}^{\infty} R_j(x_l, u_l)$, $j = 1, \dots, N$, are used to evaluate the performance of system (1), where $R_j(x_l, u_l)$ is a utility function that satisfies $R_j(x, u) \triangleq W_j(u) + S_j(x)$. $W_j(u)$ and $S_j(x)$ are positive definite functions of x and u , respectively. Define the vector $J = [J_1, \dots, J_N]^T$, with J_j represents the j -th performance index, and the state value functions for an admissible control policy $u(x)$ are defined as $V_{(j,u)}(x_k) \triangleq \sum_{l=k}^{\infty} R_j(x_l, u(x_l))$, $j = 1, \dots, N$, and $V_u = [V_{(1,u)}, \dots, V_{(N,u)}]^T$. Then, the above optimization problem is $\min_u V_u(x_0)$.

Main results:

Policy gradient adaptive dynamic programming for multi-objective nonlinear systems: The value function $V_u(x)$ can be expressed as

$$\begin{aligned} V_{(j,u)}(x_k) &= R_j(x_k, u(x_k)) + \sum_{l=k+1}^{\infty} R_j(x_l, u(x_l)) \\ &= R_j(x_k, u(x_k)) + V_{(j,u)}(x_{k+1}) \\ &= R_j(x_k, u(x_k)) + V_{(j,u)}(F(x_k, u_k)). \end{aligned} \quad (2)$$

In order to describe the reward of each action more directly, the Q -function, also called state-action value function is defined as

$$Q_u(x_k, \mu) \triangleq R(x_k, \mu) + \sum_{l=k+1}^{\infty} R(x_l, u(x_l)) \quad (3)$$

where $u(x) \in \mathcal{U}(X)$ and $Q_u(0, 0) = 0$. In (3), $Q_u(x_k, \mu)$ represents the value of the performance index of system (1) by using control policy u after taking action μ at state x_k . Then the j -th Q -function can be expressed as $Q_{(j,u)}(x_k, \mu) = R_j(x_k, \mu) + \sum_{l=k+1}^{\infty} R_j(x_l, u(x_l)) = R_j(x_k, \mu) + Q_{(j,u)}(x_{k+1}, u) = R_j(x_k, \mu) + V_{(j,u)}(x_{k+1})$. For

$$Q_{(j,u)}^{(i)}(x_k, \mu) = R_j(x_k, \mu) + Q_{(j,u)}^{(i)}(x_{k+1}, u^{(i)}) \quad (4)$$

Corresponding author: Zhuping Wang.

Citation: H. Zhang, Y. Li, Z. Wang, Y. Ding, and H. Yan, "Policy gradient adaptive dynamic programming for model-free multi-objective optimal control," *IEEE/CAA J. Autom. Sinica*, vol. 11, no. 4, pp. 1060–1062, Apr. 2024.

H. Zhang, Y. Li, Z. Wang, and Y. Ding are with the Department of Control Science and Engineering, Tongji University, Shanghai 200092, China (e-mail: zhang_hao@tongji.edu.cn; 1810366@tongji.edu.cn; elewzp@tongji.edu.cn; 1930757@tongji.edu.cn).

H. Yan is with the Key Laboratory of Advanced Control and Optimization for Chemical Processes of Ministry of Education, School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China (e-mail: hcyan@ecust.edu.cn).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JAS.2023.123381

where $j = 1, \dots, N$ and i refers to the iteration index. Update the control policy $u(x)$ with gradient descent as

$$u^{(i+1)}(x) = u^{(i)}(x) - \alpha \sum_{j=1}^N w_j \nabla_{\mu} Q_{(j,u)}^{(i)}(x, \mu) \Big|_{\mu=u^{(i)}(x)} \quad (5)$$

where α and w_j are constants, α represents the learning rate. w_j is the weight of the j -th performance index, which can generally be set according to the importance of each target. Thus, the weight vector $\omega = [\omega_1, \omega_2, \dots, \omega_N]$ can be constructed, which satisfies $\sum_{j=1}^N \omega_j = 1$.

If $u^*(x)$ reaches the optimal control policy, the j -th Q-function is given by

$$Q_j^*(x_k, \mu) = R_j(x_k, \mu) + V_j^*(x_{k+1}) \quad (6)$$

where $Q_j^*(x, \mu) \triangleq Q_{(j,u^*)}(x, \mu)$, and $V_j^*(x) \triangleq V_{(j,u^*)}(x)$.

Theorem 1: Let the initial control policy $u^{(0)}(x)$ be admissible, that is, $u^{(0)}(x) \in \mathcal{U}(X)$, and the learning rate α satisfies (4). The Q-functions for $\forall j \in \{1, 2, \dots, N\}$ and control policy are given in (4) and (5). Then, $u^{(l)}(x)$ is admissible and the system is asymptotically stable.

Proof: The following proof can be carried out by mathematical induction. At first, there is an initial policy $u^{(0)}(x) \in \mathcal{U}(X)$ when $i = 0$. Assuming that $u^{(l)}(x) \in \mathcal{U}(X)$ holds when $i = l$, then the situation when $i = l+1$ needs to be discussed. With the control policy $u^{(l+1)}(x)$, the system is given by

$$x_{k+1} = F(x_k, u^{(l+1)}(x_k)). \quad (7)$$

The state-value function $V_j^{(l)}(x_k)$ can be selected as the Lyapunov function. Then the difference of $V_j^{(l)}(x_k)$ along the state trajectory of system (7) can be obtained as

$$\begin{aligned} \Delta V_j^{(l)}(x_k) &= V_j^{(l)}(F(x_k, u^{(l+1)}(x_k))) - V_j^{(l)}(x_k) \\ &= \mathcal{H}_j(x_k, u^{(l+1)}, V_j^{(l)}) - R_j(x_k, u^{(l+1)}(x_k)). \end{aligned}$$

Then, $\mathcal{H}_j(x_k, u^{(l+1)}, V_j^{(l)}) \leq 0$ holds. Thus, $\Delta V_j^{(l)}(x_k) \leq -R_j(x_k, u^{(l+1)}(x_k)) < 0$ with $x, u \neq 0$, which shows the asymptotic stability of system (7). ■

Neural network implementation of PGADP algorithm with actor-critic structure: The online data set is $z_k = \{x_{k-1}, u_{k-1}, x_k\}$ and the offline data set is defined as $Z_M = \{x_l, \mu_l, x'_l | x_l, x'_l \in X, \mu_l \in \mathcal{U}, l = 1, 2, \dots, M\}$, where M is the data set size. Denote $\Psi_j(x, \mu) \triangleq \{\psi_{js}(x, \mu)\}_{s=1}^\infty$, and $\Phi(x) \triangleq \{\phi_s(x)\}_{s=1}^\infty$ as complete sets of linearly independent basis functions. According to (4) and (5), $Q_j(x, \mu)$ and $u(x)$ can be expressed as

$$\begin{aligned} \hat{Q}_j^{(i)}(x, \mu) &\triangleq \sum_{s=1}^{L_1} \hat{\theta}_{js}^{(i)} \psi_{js}(x, \mu) = \Psi_{jL}^T(x, \mu) \hat{\theta}_j^{(i)} \\ \hat{u}^{(i)}(x) &\triangleq \sum_{s=1}^{L_2} \hat{v}_s^{(i)} \phi_s(x, \mu) = \Phi_L^T(x) \hat{v}^{(i)} \end{aligned} \quad (8)$$

where $\Psi_{jL}(x, \mu) \triangleq [\psi_{j1}(x, \mu), \psi_{j2}(x, \mu), \dots, \psi_{jL_1}(x, \mu)]^T$ is the activation function of j -th critic network, and $\Phi_L^T(x) \triangleq [\phi_1(x, \mu), \phi_2(x, \mu), \dots, \phi_{L_2}(x, \mu)]^T$ is the activation function of the actor network. The vectors $\hat{\theta}_j^{(i)} \triangleq [\hat{\theta}_{j1}^{(i)}, \hat{\theta}_{j2}^{(i)}, \dots, \hat{\theta}_{jL_1}^{(i)}]^T$ and $\hat{v}^{(i)} \triangleq [\hat{v}_1^{(i)}, \hat{v}_2^{(i)}, \dots, \hat{v}_{L_2}^{(i)}]^T$ are estimated weights of the actor and critic networks, respectively.

There are errors in the approximate estimation process using neural networks. Based on (4), the error function of critic network can be calculated as

$$\begin{aligned} e_{Q_j}^{(k)}(x, \mu, x') &\triangleq \hat{Q}_j^{(k)}(x, \mu) - \hat{Q}_j^{(k)}(x', \hat{u}^{(k)}) - \mathcal{R}_j(x, \mu) \\ &= \Psi_{jL}^T(x, \mu) \hat{\theta}_j^{(k)} - \Psi_{jL}^T(x', \Phi_L^T(x') \hat{v}^{(k)}) \hat{\theta}_j^{(k)} - \mathcal{R}_j(x, \mu). \end{aligned} \quad (9)$$

The main basis for residual weight method to solve $\hat{\theta}_j^{(k)}$ is that the weighted integral is zero in the sense of weighted average, which can be expressed as

$$\int_{\mathcal{K}} C_{Q_j}(x, \mu) e_{Q_j}^{(k)}(x, \mu, x') d(x, \mu) = 0 \quad (10)$$

where $C_{Q_j}(x, \mu) \triangleq [c_{Q_j,1}(x, \mu), c_{Q_j,2}(x, \mu), \dots, c_{Q_j,L_1}(x, \mu)]^T$ represents a weight function vector with respect to (x, μ) . For simplicity, denote $\psi_{jp}(x_k, \mu_k) = \psi_{jp}^k$, $C_{Q_j}(x_k, \mu_k) = C_{Q_j}^k$ and

$$\mathcal{A}_j^0 \triangleq \sum_{\tau=1}^M C_{Q_j}^{\tau} (\Psi_{jL}^{\tau})^T, \quad \mathcal{B}_j^0 \triangleq \sum_{\tau=1}^M C_{Q_j}^{\tau} \Psi_{jL}^T(x_{\tau'}, \Phi_L^T(x_{\tau'}) \eta_0)$$

$$\mathcal{A}_j^k \triangleq \mathcal{A}_j^0 + C_{Q_j}^{k-1} (\Psi_{jL}^{k-1})^T, \quad \zeta_j^0 \triangleq \sum_{\tau=1}^M C_{Q_j}^{\tau} \mathcal{R}_j(x_{\tau}, \mu_{\tau})$$

$$\mathcal{B}_j^k \triangleq \sum_{\tau=1}^M C_{Q_j}^{\tau} \Psi_{jL}^T(x_{\tau'}, \Phi_L^T(x_{\tau'}) \eta_k) + C_{Q_j}^{k-1} (\Psi_{jL}^k)^T$$

$$\zeta_j^k \triangleq \zeta_j^0 + C_{Q_j}^{k-1} \mathcal{R}_j(x_{k-1}, \mu_{k-1}), \quad \Xi \triangleq \frac{\Gamma_{\mathcal{K}}}{M+1}$$

where $\Gamma_{\mathcal{K}} \triangleq \int_{\mathcal{K}} d(x, \mu)$. Then, $\hat{\theta}_j^{(0)}$ and $\hat{\theta}_j^{(k)}$ are written as

$$\hat{\theta}_j^{(0)} = \left[\frac{\Gamma_{\mathcal{K}}}{M} \mathcal{A}_j^0 - \frac{\Gamma_{\mathcal{K}}}{M} \mathcal{B}_j^0 \right]^{-1} \times \frac{\Gamma_{\mathcal{K}}}{M} \zeta_j^0, \quad \hat{\theta}_j^{(k)} = [\Xi \mathcal{A}_j^k - \Xi \mathcal{B}_j^k]^{-1} \times \Xi \zeta_j^k. \quad (11)$$

The weight vector ρ_j^0 and ρ_j^k of j -th critic network are computed with $\rho_j^0 = (\mathcal{A}_j^0 - \mathcal{B}_j^0)^{-1} \zeta_j^0$, $\rho_j^k = (\mathcal{A}_j^k - \mathcal{B}_j^k)^{-1} \zeta_j^k$.

Next, the update rule for η_k is exported, which represents the weight of the actor network. Denote $\mathcal{S}(x) = C_u^T(x) \sum_{j=1}^N w_j \nabla_{\mu} \times \Psi_{jL}^T(x, \Phi_L^T(x) \hat{v}^{(k)})$. Similar to the weight vector of critic network, the weight vector of actor network $\hat{v}^{(k+1)}$ is written as

$$\hat{v}^{(k+1)} = \hat{v}^{(k)} - \alpha \left[\int_X C_u^T(x) \Phi_L^T(x) dx \right]^{-1} \times \int_X \mathcal{S}(x) \hat{\theta}_j^{(k)} dx. \quad (12)$$

The Monte Carlo integration method is also used to calculate the integral, that is

$$\begin{aligned} \int_X C_u^T(x) \Phi_L^T(x) dx &= \frac{\Gamma_{\mathcal{K}}}{M+1} \mathcal{P}_k \\ \int_X \mathcal{S}(x) \hat{\theta}_j^{(k)} dx &= \int_X \mathcal{S}(x) \rho_j^k dx = \frac{\Gamma_{\mathcal{K}}}{M+1} \mathcal{D}_k(\rho_j^k) \end{aligned}$$

where

$$\begin{aligned} \mathcal{P}_k &\triangleq \mathcal{P}_0 + C_u^T(x_k) \Phi_L^T(x_k) = \sum_{\tau=1}^M C_u^T(x_{\tau}) \Phi_L^T(x_{\tau}) + C_u^T(x_k) \Phi_L^T(x_k) \\ \mathcal{D}_k(\rho_j^k) &\triangleq \sum_{\tau=1}^M \mathcal{S}(x_{\tau}) \rho_j^k + \mathcal{S}(x_k) \rho_j^k. \end{aligned} \quad (13)$$

Summarized from (12) to (13), the update rule for η_k can be obtained

$$\eta_{k+1} = \eta_k - \alpha \mathcal{P}_k^{-1} \mathcal{D}_k(\rho_j^k). \quad (14)$$

Theorem 2: For $\forall (x, \mu) \in \mathcal{K}$, $\varepsilon_{1j} > 0$, $\varepsilon_{2j} > 0$, there exists a positive integer \mathcal{K} that for $\forall k \geq \mathcal{K}$, there are

$$|\hat{Q}_j^{(k)}(x, \mu) - Q_j^{(k)}(x, \mu)| \leq \varepsilon_{1j}$$

and

$$|\hat{Q}_j^{(k)}(x, \mu) - Q_j^{(*)}(x, \mu)| \leq \varepsilon_{2j}.$$

Proof: The detailed proof process is similar to the proof of Theorem 2 in [5]. For brevity, it is not described in detail here. ■

Simulation: In this section, the effectiveness of the PGADP-based multi-objective control method will be tested through the simulation of a discrete-time nonlinear system.

Consider the multi-objective optimal control problem for the following system:

$$x_{k+1} = \begin{bmatrix} (x_{k,1} + x_{k,2}^2 + u_k) \cos(x_{k,2}) \\ 0.5(x_{k,1}^2 + x_{k,2} + u_k) \sin(x_{k,2}) \end{bmatrix}$$

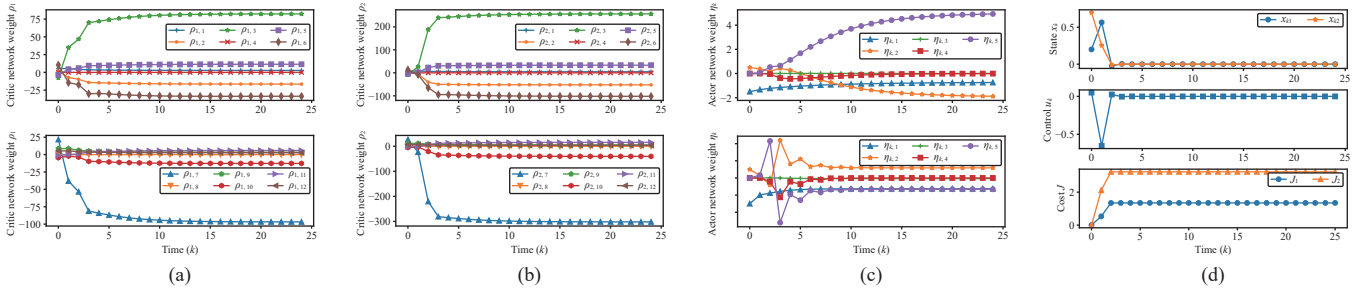


Fig. 1. The weight vectors of the critic network when $\alpha = 0.02$. (a) The weight vectors for the first critic network $\rho_{1,1} - \rho_{1,12}$; (b) The weight vectors for the second critic network $\rho_{2,1} - \rho_{2,12}$; (c) The weight vectors for actor network η_k ; (d) The state trajectories $x_{k,1}$, $x_{k,2}$, the control policy trajectories u_k , and the performance index $J_{k,1}$, $J_{k,2}$.

where the state $x_k = [x_{k,1}, x_{k,2}]^T$, and the initial state $x_0 = [0.2, 0.7]^T$. Here, the case of two goals is considered. The simulations are carried out separately from the perspective of different learning rates and different value functions. The value functions can be given by $V_{(1,u)}(x_0) = \sum_{l=0}^{\infty} x_l^T x_l + u_l^T u_l$ and $V_{(2,u)}(x_0) = \sum_{l=0}^{\infty} 2x_l^T x_l + u_l^T u_l$. Collect 40 frames of offline data and select $\alpha = 0.02$, $\omega = [0.2, 0.8]$, and $\eta_0 = [-1.5, 0.5, 0, 0]^T$. Then the simulation results are shown in Figs. 1(a) and 1(b) represent the trajectories of the critic network weight vector $\rho_{k,1}$ and $\rho_{k,2}$. Fig. 1(c) demonstrates the convergence of the critic network weight vectors η_k . Select $\alpha = 0.06$ as a comparative experiment, and the results are shown in Figs. 2(a)–2(c). It can be seen from the figure that although the convergence is finally reached, the process is more oscillating compared with the case of $\alpha = 0.02$.

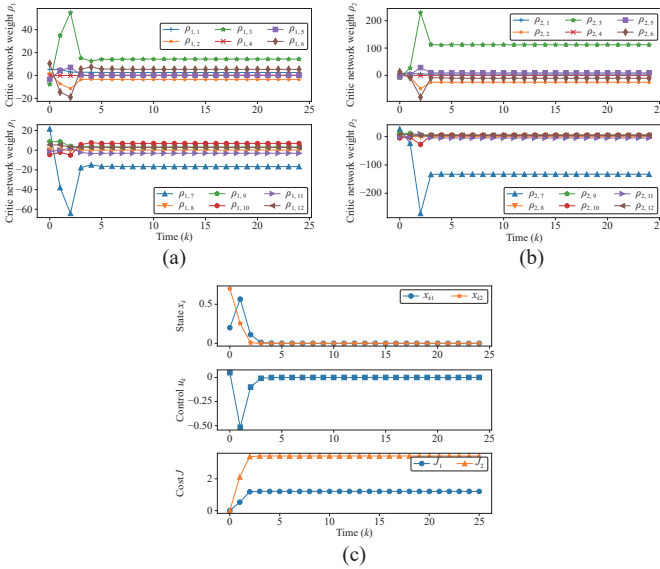


Fig. 2. The weight vectors of the critic network when $\alpha = 0.06$. (a) The weight vectors for the first critic network $\rho_{1,1} - \rho_{1,12}$; (b) The weight vectors for the second critic network $\rho_{2,1} - \rho_{2,12}$; (c) The state trajectories $x_{k,1}$, $x_{k,2}$, the control policy trajectories u_k , and the performance index $J_{k,1}$, $J_{k,2}$.

Conclusion: In this letter, by using data from real system instead of calculating by the mathematical system models, a PGADP-based control algorithm is proposed to solve the multi-objective optimal control problem. The optimal control policy is designed to ensure the multiple objective functions converge to the optimal vectors in the Pareto sense, and the stability and convergence of the algorithm is proved. The policy gradient method is used to reduce unnecessary calculations. In addition, the experience replay technique is used to derive the rules for updating actor-critic network parameters. Finally,

a simulation example is given to verify the performance of the method. The future works can be extended to the multi-objective optimization problem with conflicts and more actual situations.

Acknowledgments: This work was supported in part by the National Natural Science Foundation of China (61922063, 62273255, 62150026), in part by the Shanghai International Science and Technology Cooperation Project (21550760900, 22510712000), the Shanghai Municipal Science and Technology Major Project (2021SHZDZX0100), and the Fundamental Research Funds for the Central Universities.

References

- [1] S. Chinchali, S. C. Livingston, M. Chen, and M. Pavone, "Multi-objective optimal control for proactive decision making with temporal logic models," *Int. J. Robotics Research*, vol. 38, no. 12–13, pp. 1490–1512, 1490.
- [2] J. A. Villegas-Florez, B. S. Hernandez-Orsorio, and E. Giraldo, "Multi-objective optimal control of resources applied to an electric power distribution system," *Engineering Letters*, vol. 28, no. 3, pp. 756–761, 2020.
- [3] J. Chen, J. Sun, and G. Wang, "From unmanned systems to autonomous intelligent systems," *Engineering*, vol. 12, no. 5, pp. 16–19, 2022.
- [4] Q. Wei, F.-Y. Wang, D. Liu, and X. Yang, "Finite-approximation-error-based discrete-time iterative adaptive dynamic programming," *IEEE Trans. Cyber.*, vol. 44, no. 12, pp. 2820–2833, 2014.
- [5] B. Luo, D. Liu, H.-N. Wu, D. Wang, and F. L. Lewis, "Policy gradient adaptive dynamic programming for data-based optimal control," *IEEE Trans. Cyber.*, vol. 47, no. 10, pp. 3341–3354, 2016.
- [6] D. Zhao, Q. Zhang, D. Wang, and Y. Zhu, "Experience replay for optimal control of nonzero-sum game systems with unknown dynamics," *IEEE Trans. Cyber.*, vol. 46, no. 3, pp. 854–865, 2015.
- [7] Y. Yang, W. Gao, H. Modares, and C.-Z. Xu, "Robust actor-critic learning for continuous-time nonlinear systems with unmodeled dynamics," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 6, pp. 2101–2112, 2021.
- [8] J. Xu, L. Wang, Y. Liu, and H. Xue, "Event-triggered optimal containment control for multi-agent systems subject to state constraints via reinforcement learning," *Nonlinear Dynamics*, vol. 109, pp. 1651–1670, 2022.
- [9] J. Xu, L. Wang, Y. Liu, J. Sun, and Y. Pan, "Finite-time adaptive optimal consensus control for multi-agent systems subject to time-varying output constraints," *Applied Math. and Computation*, vol. 427, p. 127176, 2022.
- [10] Y. Yang, B. Kiumarsi, H. Modares, and C. Xu, "Model-free λ -policy iteration for discrete-time linear quadratic regulation," *IEEE Trans. Neural Networks and Learning Systems*, vol. 34, no. 2, pp. 635–649, 2023.
- [11] V. G. Lopez and F. L. Lewis, "Dynamic multi objective control for continuous-time systems using reinforcement learning," *IEEE Trans. Autom. Control*, vol. 64, no. 7, pp. 2869–2874, 2019.