Optimal trajectory tracking control for USVs under dynamic uncertainties and time-varying disturbances via PI and IRL algorithms

Tran Thanh Tuan[1], Vu Quoc Huy[1*], Nguyen Quang Hung[2]

[1]Institute of Automation, Academy of Military Science and Technology, 89 Ly Nam De, Hoan Kiem, Hanoi, Vietnam;
[2]East Asia University of Technology, Phan Tay Nhac, Xuan Phuong, Ha Noi, Vietnam.
[*]Corresponding author: maihuyvu@gmail.com
Received 27 Sep. 2025; Revised 17 Nov. 2025; Accepted 10 Dec. 2025; Published 25 Dec. 2025.
DOI: https://doi.org/10.54939/1859-1043.j.mst.108.2025.11-20

ABSTRACT

This paper presents a model-free optimal control framework for trajectory tracking of Unmanned Surface Vehicles operating under unknown dynamics and time-varying disturbances via Policy Iteration (PI) and Integral Reinforcement Learning (IRL) algorithms. The IRL-PI controller is developed based on an order reduction technique and an off-policy Actor-Critic neural network structure, allowing real-time approximation of the Hamilton-Jacobi-Bellman solution without requiring model knowledge. Simulation results on a three three-degree-of-freedom (3-DOF) USV model demonstrate that the proposed method outperforms conventional controllers in both tracking accuracy and robustness. These results highlight the potential of the IRL-PI controller to develop robust control solutions for complex marine systems operating in uncertain and dynamic environments.

Keywords: Integral reinforcement learning; PI; Optimal control; HJB; USVs.

1. INTRODUCTION

In real-world marine environments, the dynamics of unmanned surface vehicles (USVs) are inherently nonlinear and affected by unknown disturbances such as wind, waves, and ocean currents [1-7]. As a result, deriving an accurate mathematical model of USVs is difficult, and traditional optimal control approaches often fail to compute feasible solutions due to modeling uncertainties and nonlinearities. To overcome the aforementioned challenges, extensive research has been devoted to the development of advanced nonlinear control strategies capable of adapting to system variations in real time, including control frameworks grounded in Lyapunov stability theory, adaptive optimized backstepping (AOBC) [4], disturbance-observer-based sliding mode control [3], and model predictive control (MPC) [7]. To further enhance robustness, hybrid intelligent control approaches, including adaptive neural network (NN) control, disturbance observer-based control, and neural sliding backstepping, have been proposed [2]. Although these techniques exhibit strong capability in handling nonlinearity and parametric uncertainties, they often require model information or depend on restrictive assumptions and usually suffer from practical limitations, such as neglecting input constraints, sensitivity to parameter tuning, or increased computational complexity.

To achieve the trajectory tracking control task for USV, finding the optimal solution for such nonlinear systems requires solving a nonlinear partial differential Hamilton–Jacobi–Bellman (HJB) equation, which is generally very difficult when model uncertainties are present. In recent years, Reinforcement Learning (RL) has emerged as a promising paradigm for model-free optimal control, where the control policy is learned directly from data-driven interactions rather than explicit model information [8, 10]. In particular, Actor–Critic neural network (ACNN) architectures, developed within the framework of Adaptive Dynamic Programming (ADP), have demonstrated significant potential in approximating solutions to the HJB equation and enabling online optimal control design [5, 6, 11-13]. With advances in sensor fusion and data-driven modeling, integrating RL with intelligent nonlinear control offers a promising pathway toward robust, adaptive, and real-time control of USVs in uncertain marine environments.

This paper proposes an algorithm based on the Integral Reinforcement Learning (IRL) [9, 14] and Policy Iteration (PI) [5, 11] framework for optimal trajectory tracking of USVs with completely unknown dynamics and external disturbances. The approach is entirely data-driven, and employs an actor–critic architecture with NNs to approximate the Bellman error and iteratively update the control policy. By leveraging off-policy learning, the method improves data efficiency and robustness, enabling real-time implementation without prior system identification. The proposed controller ensures prescribed tracking performance, adaptability to dynamic changes, and time-varying disturbances, making it highly suitable for USV operations in uncertain marine environments.

2. METHODOLOGY

2.1. Problem statements

According to [1], the overall motion of the 3-DOF USV (illustrated in figure 1) is:

$$\begin{cases} \boldsymbol{M}\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{v})\boldsymbol{v} + \boldsymbol{D}(\boldsymbol{v})\boldsymbol{v} + \boldsymbol{d}(t) = \boldsymbol{\tau}_m \\ \dot{\boldsymbol{\eta}} = \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{v} \end{cases} \tag{1}$$

where $\boldsymbol{\eta} = [\eta_x\ \eta_y\ \eta_\psi]^T \in \mathbb{R}^3$ are the planar position and heading, respectively; $\boldsymbol{v}(t) = [v_x(t)\ v_y(t)\ v_\psi(t)]^T \in \mathbb{R}^3$ are velocities in the body-fixed frame, respectively; $\boldsymbol{M} \in \mathbb{R}^{3\times3}$ is the inertia matrix. $\boldsymbol{C}(\boldsymbol{v}) \in \mathbb{R}^{3\times3}$ is the Coriolis and centripetal acceleration matrix. $\boldsymbol{D}(\boldsymbol{v}) \in \mathbb{R}^{3\times3}$ is the damping matrix, $\boldsymbol{\tau}_m \in \mathbb{R}^3$ is the control input, $\boldsymbol{d}(t)$ is the disturbance. $\boldsymbol{J}(\boldsymbol{\eta}) \in SO(3)$ is a Jacobian transformation matrix relating in body-fixed and NED reference frames:

$$\boldsymbol{J}(\boldsymbol{\eta}) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix};\ \boldsymbol{J}^{-1}(\boldsymbol{\eta}) = \boldsymbol{J}^T(\boldsymbol{\eta})$$

For the convenience of controller design, the dynamic model in Eq.(1) is reformulated into the following form:

$$\bar{\boldsymbol{M}}(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + \bar{\boldsymbol{C}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}} + \bar{\boldsymbol{D}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}} = \boldsymbol{\tau}_\eta + \boldsymbol{\tau}_{ed} \tag{2}$$

where: $\bar{\boldsymbol{M}} = \boldsymbol{J}^{-T}\boldsymbol{M}\boldsymbol{J}^{-1}$; $\bar{\boldsymbol{C}} = \boldsymbol{J}^{-T}(\boldsymbol{C} - \boldsymbol{M}\boldsymbol{J}^{-1}\dot{\boldsymbol{J}})\boldsymbol{J}^{-1}$; $\bar{\boldsymbol{D}} = \boldsymbol{J}^{-T}\boldsymbol{D}\boldsymbol{J}^{-1}$; $\boldsymbol{\tau}_\eta = \boldsymbol{J}^{-T}\boldsymbol{\tau}_m$; $\boldsymbol{\tau}_{ed} = \boldsymbol{J}^{-T}\boldsymbol{d}$.

To develop the proposed controller, the following assumptions are introduced:

Assumption 1. Vector $\boldsymbol{\eta}$ and $\dot{\boldsymbol{\eta}}$ are bounded by $\bar{\eta}_1, \bar{\eta}_2 \in \mathbb{R}^+$ such that $\|\boldsymbol{\eta}\| \le \bar{\eta}_1, \|\dot{\boldsymbol{\eta}}\| \le \bar{\eta}_2$. Given that $\boldsymbol{\eta}(t), \dot{\boldsymbol{\eta}}(t) \in L_\infty$, it is ensured that all functions $\boldsymbol{C}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}}), \boldsymbol{D}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})$, as well as the first, second partial derivatives with respect to $\boldsymbol{\eta}(t), \dot{\boldsymbol{\eta}}(t)$, are bounded.

Assumption 2. The reference trajectory $\boldsymbol{\eta}_{ref}$ and the 1st, 2nd derivatives exist and are bounded. There exist transfer functions $\boldsymbol{h}_1(\boldsymbol{\eta}_{ref}), \boldsymbol{h}_2(\boldsymbol{\eta}_{ref})$ such that $\dot{\boldsymbol{\eta}}_{ref} = \boldsymbol{h}_1(\boldsymbol{\eta}_{ref}), \ddot{\boldsymbol{\eta}}_{ref} = \boldsymbol{h}_2(\dot{\boldsymbol{\eta}}_{ref})$.

2.2. Controller design

2.2.1. General control structure

Assignment of the coordinates for the USV is presented in figure 1. The structure of the proposed controller is depicted in figure 2.

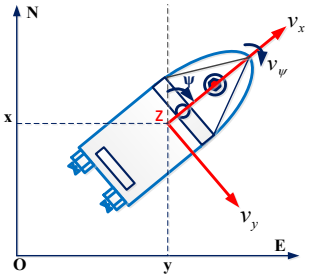

Figure 1. *Coordinates of a USV.*

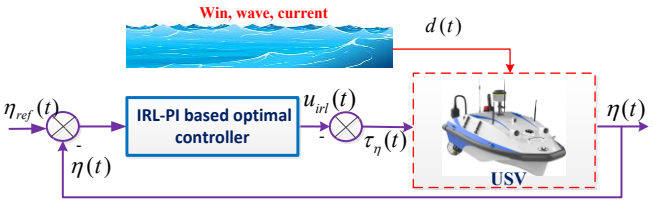

Figure 2. *The control scheme of the USV system.*

To design a controller that ensures $\boldsymbol{\eta}(t)$ tracks a reference trajectory $\boldsymbol{\eta}_{ref}(t)$, while minimizing the measurement performance index. To quantify this target, a tracking error, denoted by $\boldsymbol{e} \in \mathbb{R}^n$:

$$\boldsymbol{e} = \boldsymbol{\eta}_{ref} - \boldsymbol{\eta} \Rightarrow \dot{\boldsymbol{e}} = \dot{\boldsymbol{\eta}}_{ref} - \dot{\boldsymbol{\eta}} \tag{3}$$

For reducing second-order uncertainty and disturbed USV, we define a sliding surface:

$$\boldsymbol{s}(t) = \dot{\boldsymbol{e}} + \boldsymbol{\lambda}\boldsymbol{e};\ \boldsymbol{\lambda} = \text{diag}(\lambda_1, \lambda_2, \lambda_3), \lambda_i > 0; i = \overline{1,3} \tag{4}$$

Take the time derivative of (4), then multiply both sides by $\bar{\boldsymbol{M}}(\boldsymbol{\eta})$ and substitute (2) into that, the system for $\boldsymbol{s}(t)$ can be obtained as (5):

$$\bar{\boldsymbol{M}}(\boldsymbol{\eta})\dot{\boldsymbol{s}} = -(\bar{\boldsymbol{C}} + \bar{\boldsymbol{D}})\boldsymbol{s} - \boldsymbol{\tau}_\eta + \boldsymbol{f} - \boldsymbol{\tau}_{ed} \tag{5}$$

Where $\boldsymbol{f}$ is a vector of multi-variable nonlinear functions:

$$\boldsymbol{f} = \bar{\boldsymbol{M}}(\boldsymbol{\eta})\left(\ddot{\boldsymbol{\eta}}_{ref} + \boldsymbol{\lambda}\dot{\boldsymbol{e}}\right) + (\bar{\boldsymbol{C}} + \bar{\boldsymbol{D}})\left(\dot{\boldsymbol{\eta}}_{ref} + \boldsymbol{\lambda}\boldsymbol{e}\right) \tag{6}$$

The proposed controller relies on figure 2, which is an IRL-PI:

$$\boldsymbol{\tau}_\eta(t) = -\boldsymbol{u}_{irl}(t) \tag{7}$$

From (5) and (7), we have the following model after eliminating the estimation error $\boldsymbol{f} - \boldsymbol{\tau}_{ed}$:

$$\bar{\boldsymbol{M}}(\boldsymbol{\eta})\dot{\boldsymbol{s}} = -(\bar{\boldsymbol{C}} + \bar{\boldsymbol{D}})\boldsymbol{s} + \boldsymbol{u}_{irl}(t) \tag{8}$$

From (3), (4), and (8), we have the following time-varying system:

$$\dot{\boldsymbol{z}} = \begin{bmatrix} -\boldsymbol{\lambda}\boldsymbol{e} + \boldsymbol{s} \\ -\bar{\boldsymbol{M}}^{-1}\left(\boldsymbol{\eta}_{ref} - \boldsymbol{e}\right) - (\bar{\boldsymbol{C}} + \bar{\boldsymbol{D}})\left(\boldsymbol{\eta}_{ref} - \boldsymbol{e}, \dot{\boldsymbol{\eta}}_{ref} + \boldsymbol{\lambda}\boldsymbol{e} - \boldsymbol{s}\right)\boldsymbol{s} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_{n\times n} \\ \bar{\boldsymbol{M}}^{-1} \end{bmatrix}\boldsymbol{u}_{irl}(t) \tag{9}$$

Where $\boldsymbol{z} = [\boldsymbol{e}^T\ \boldsymbol{s}^T]^T$, the infinite horizon exponential cost function to be minimized is [2, 5]:

$$V(\boldsymbol{z}, \boldsymbol{u}_{irl}) = \int_t^\infty e^{-\gamma(\tau - t)}\left(\frac{1}{2}\boldsymbol{z}^T\boldsymbol{Q}\boldsymbol{z} + \frac{1}{2}\boldsymbol{u}_{irl}^T\boldsymbol{R}\boldsymbol{u}_{irl}\right)d\tau \tag{10}$$

$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{Q}_e & \boldsymbol{0}_{n\times 3n} \\ \boldsymbol{0}_{3n\times n} & \boldsymbol{0}_{3n\times 3n} \end{bmatrix} \in \mathbb{R}^{12}, \boldsymbol{Q}_e = \text{diag}(q_{e1}\ q_{e2}\ q_{e3}); \boldsymbol{R} \in \mathbb{R}^{n\times n}$ are positive definite symmetric matrices, γ is the discount factor. For the satisfaction of finite value, it needs to guarantee that:

$$\lim_{t\to\infty} e^{-\gamma(\tau - t)}\left(\frac{1}{2}\boldsymbol{z}^T\boldsymbol{Q}\boldsymbol{z} + \frac{1}{2}\boldsymbol{u}_{irl}^T\boldsymbol{R}\boldsymbol{u}_{irl}\right) = 0 \tag{11}$$

The discount factor γ is introduced to regulate $\boldsymbol{u}_{irl}(\tau)$ and maintain control performance. Consequently, off-policy learning techniques with function approximation are employed to estimate the remaining term $\boldsymbol{u}_{irl}(\tau)$, ensuring that the desired tracking performance is still achieved.

Define the state vector $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{z}^T & \boldsymbol{\eta}_{ref}^T & \dot{\boldsymbol{\eta}}_{ref}^T \end{bmatrix}^T$, the dynamics equation of $\boldsymbol{X}$ is:

$$\dot{\boldsymbol{X}}(t) = \boldsymbol{F}(\boldsymbol{X}) + \boldsymbol{G}(\boldsymbol{X})\boldsymbol{u}_{irl} \tag{12}$$

$$\boldsymbol{F}(\boldsymbol{X}) = \begin{bmatrix} -\boldsymbol{\lambda}\boldsymbol{e} + \boldsymbol{s} \\ \boldsymbol{M}^{-1}\left(\boldsymbol{\eta}_{ref}\right)\left(\boldsymbol{\eta}_{ref} - \boldsymbol{e}\right) - (\boldsymbol{C} + \boldsymbol{D})\left(\boldsymbol{\eta}_{ref} - \boldsymbol{e}, \dot{\boldsymbol{\eta}}_{ref} + \boldsymbol{\lambda}\boldsymbol{e} - \boldsymbol{s}\right)\boldsymbol{s} \\ \boldsymbol{h}_1\left(\boldsymbol{\eta}_{ref}\right) \\ \boldsymbol{h}_2\left(\dot{\boldsymbol{\eta}}_{ref}\right) \end{bmatrix};\ \boldsymbol{G}(\boldsymbol{X}) = \begin{bmatrix} \boldsymbol{0}_{n\times n} \\ \boldsymbol{M}^{-1} \\ \boldsymbol{0}_{2n\times n} \end{bmatrix} \tag{13}$$

Let $\pi(\boldsymbol{X}, \boldsymbol{u}_{irl}) = \frac{1}{2}\boldsymbol{X}^T\boldsymbol{Q}\boldsymbol{X} + \frac{1}{2}\boldsymbol{u}_{irl}^T\boldsymbol{R}\boldsymbol{u}_{irl}$, the cost function of (11) can be rewritten as:

$$V(\boldsymbol{X}, \boldsymbol{u}_{irl}) = \int_t^\infty e^{-\gamma(\tau - t)}\pi(\boldsymbol{X}, \boldsymbol{u}_{irl})d\tau \tag{14}$$

Remark 1: To guarantee stability in the design of optimal control, it is necessary to consider a specific class of control signals referred to as "*Admissible Policies*" (see [13]).

2.2.2. Design the optimal controller

Based on equation (12), the optimal control inputs can be regarded as static state-feedback laws $\boldsymbol{u}_{irl}^{*}(\boldsymbol{X})$ and the corresponding time-invariant Bellman function $V^{*}(\boldsymbol{X}(t))$ can be directly derived as:

$$V^{*}(\boldsymbol{X}(t)) = \min_{\boldsymbol{u}_{irl}(\boldsymbol{X})\in\Upsilon(\Omega)} V(\boldsymbol{X}(t), \boldsymbol{u}_{irl}(\boldsymbol{X}(t)) \tag{15}$$

Taking the time derivative of the function $V^{*}(\boldsymbol{X}(t))$:

$$\dot{V}^{*}(\boldsymbol{X}(t)) = \frac{\partial V^{*}}{\partial \boldsymbol{X}}(\boldsymbol{F}(\boldsymbol{X}) + \boldsymbol{G}(\boldsymbol{X})\boldsymbol{u}_{irl}^{*}) \tag{16}$$

According to the Dynamic Programming (DP) principle and the cost function (14), the following results are implied:

$$\frac{V^{*}\big(\boldsymbol{X}(t)\big) - V^{*}\big(\boldsymbol{X}(t+\delta)\big)}{\delta} = \frac{1}{\delta}\int_{t}^{t+\delta} e^{-\gamma(\tau-t)}\pi\big(\boldsymbol{X}(\tau), \boldsymbol{u}_{irl}^{*}(\tau)\big)d\tau + \frac{(e^{-\gamma\delta}-1)}{\delta}V^{*}\big(\boldsymbol{X}(t+\delta)\big) \tag{17}$$

As $\delta \to 0$, and by combining with (15), (17), the function $V^{*}(\boldsymbol{X}(t))$ from $\boldsymbol{u}_{irl}(\boldsymbol{X})$ can be derived by solving the following partial differential equation:

$$\pi\big(\boldsymbol{X}(t), \boldsymbol{u}_{irl}^{*}(t)\big) - \gamma V^{*}\big(\boldsymbol{X}(t)\big) + \frac{\partial V^{*}}{\partial \boldsymbol{X}}[\boldsymbol{F}(\boldsymbol{X}) + \boldsymbol{G}(\boldsymbol{X})\boldsymbol{u}_{irl}^{*}] = 0 \tag{18}$$

Equation (18) represents the Hamilton-Jacobi-Bellman (HJB) equation reformulated for the USV tracking problem, where the value function $V^{*}(X)$ encodes both stability and optimality.

According to the principle of Dynamic Programming (DP) [5, 11, 13], the optimal value function $V^{*}(X)$ can be equivalently represented as an integral form of the cost function under the optimal policy over the entire infinite time horizon. This formulation serves as the direct foundation for applying the Policy Iteration (PI) algorithm.

In addition, $\boldsymbol{u}_{irl}(\boldsymbol{X})$ is obtainable from:

$$V^{*}(\boldsymbol{X}(t)) = \min_{\boldsymbol{u}_{irl}(\boldsymbol{X})\in\Upsilon(\Omega)} \int_{t}^{\infty} \pi(\boldsymbol{X}(t), \boldsymbol{u}_{irl}(\boldsymbol{X}(t)))\, d\tau \tag{19}$$

According to the DP principle, it implies that:

$$V^{*}\big(\boldsymbol{X}(t)\big) = \min_{\boldsymbol{u}_{irl}(\boldsymbol{X})\in\Upsilon(\Omega)} \int_{t}^{t+\delta} \pi(\boldsymbol{X}(t), \boldsymbol{u}_{irl}(\boldsymbol{X}(t)))d\tau + \min_{\boldsymbol{u}_{irl}(\boldsymbol{X})\in\Upsilon(\Omega)} e^{-\gamma\delta}V^{*}(\boldsymbol{X}(t+\delta) \tag{20}$$

As $\delta \to 0^{+}$, we can achieve $\boldsymbol{u}_{irl}(\tau)$ from $V^{*}\big(\boldsymbol{X}(t)\big)$ in the following optimization problem:

$$\min_{\boldsymbol{u}_{irl}^{*}(\boldsymbol{X})\in\Upsilon(\Omega)} \pi\big(\boldsymbol{X}(t), \boldsymbol{u}_{irl}^{*}(t)\big) - \gamma V^{*}\big(\boldsymbol{X}(t)\big) + \frac{\partial V^{*}}{\partial \boldsymbol{X}}[\boldsymbol{F}(\boldsymbol{X}) + \boldsymbol{G}(\boldsymbol{X})\boldsymbol{u}_{irl}^{*}] = 0 \tag{21}$$

By using the Policy Iteration (PI) algorithm [5, 11], the solution for equations (18) and (21) can be solved with algorithm 1.

Algorithm 1. Policy Iteration (PI) algorithm

Step 1. Initializing: *Initializing admissible policy* $\boldsymbol{u}_{RL}^{(0)}(\tau) \in \Upsilon(\Omega)$ *for system and let* $k \to 0$.

- Set iteration counter with $k = 0$.

Step 2. Policy Evaluation: *For* $k = 0, 1, 2, \ldots$*given current policy* $\boldsymbol{u}_{irl}(\tau)$*;*

- Find the approximate Bellman function by solving the partial derivative (18) *with* $V^{(k+1)}\big(\boldsymbol{X}(\tau)\big)$*:*

$$\pi\left(\boldsymbol{X}(t), \boldsymbol{u}_{irl}^{(k)}(t)\right) - \gamma V^{(k+1)}\big(\boldsymbol{X}(t)\big) + \frac{\partial V^{(k+1)}}{\partial \boldsymbol{X}}\Big[\boldsymbol{F}(\boldsymbol{X}) + \boldsymbol{G}(\boldsymbol{X})\boldsymbol{u}_{irl}^{(k)}\Big] = 0 \tag{22}$$

- On convergence, set $k \rightarrow (k+1)$.

Step 3. Policy Improvement*: Update the policy signal* $\boldsymbol{u}_{irl}^{(k+1)}(\tau)$ *using the gradient of* $V^{(k+1)}$*:*

$$\boldsymbol{u}_{irl}^{(k+1)}(\tau) = -\frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{G}^{T}(\boldsymbol{X})\frac{\partial V^{(k+1)}}{\partial \boldsymbol{X}} \tag{23}$$

Step 4. Iteration: *Return to Step 2 until* $\boldsymbol{u}_{irl}^{(k+1)}(\tau) \approx \boldsymbol{u}_{irl}^{(k)}(\tau)$.

However, this approach requires knowledge of matrices $\boldsymbol{F}(\boldsymbol{X})$ and $\boldsymbol{G}(\boldsymbol{X})$. So, we propose an off-policy IRL approach that only assumes uncertainty in $\boldsymbol{F}(\boldsymbol{X})$, which is sufficient to address the implication $\boldsymbol{u}_{irl}^{*}(\boldsymbol{X}) \rightarrow V^{*}(\boldsymbol{X}(t))$. Nevertheless, knowledge of $\boldsymbol{G}(\boldsymbol{X})$ is still necessary for solving the inverse implication $V^{*}\big(\boldsymbol{X}(t)\big) \rightarrow \boldsymbol{u}_{irl}^{*}(\boldsymbol{X})$. To comprehensively handle the model uncertainties in (12), the off-policy framework introduced in [9, 14] is adopted and further extended to the USV control problem within the PI scheme, where control actions are selected iteratively during the learning process as follows:

$$\overline{\boldsymbol{u}}_{irl}^{(k)}(\boldsymbol{X}(\tau)) = \boldsymbol{u}_{irl}^{(k)}\big(\boldsymbol{X}(\tau)\big) + \boldsymbol{\epsilon}(\tau) \text{ with } \boldsymbol{\epsilon}(\tau) = [c_j]\ \forall \tau \in \left[t_j, t_{j+1}\right], j = \overline{1, L} \tag{24}$$

Where $\boldsymbol{\epsilon}(\tau) = \boldsymbol{\epsilon}(\tau + T)$ is considered piece-wise constant and $t_0 = t \leq t_1 \leq \cdots \leq t_L = t + T$, are constant vectors, k is the index of policy in algorithm 2, $T > 0$ is the period.

Algorithm 2. The off-policy IRL algorithm

Step 1. Initialization: *admissible policy* $\boldsymbol{u}_{irl}^{(0)}(\tau) \in \Upsilon(\Omega)$. *Let* $k \rightarrow 0$. *Set iteration counter* $k = 0$.

Step 2. Iterative Update *(for* $k = 0,1,2,\ldots$*)*

- Given the current policy $\boldsymbol{u}_{irl}^{(k)}(\tau)$*, calculate the control* $\overline{\boldsymbol{u}}_{irl}^{(k)}(\tau)$ *according to* (24)*.*

- Solve the approximate Bellman function $V^{(k+1)}\big(\boldsymbol{X}(\tau)\big)$ *and* $\boldsymbol{u}_{irl}^{(k+1)}(\tau)$ *at the same time, using the off-policy Bellman equation:*

$$\begin{aligned} &V^{(k+1)}\big(\boldsymbol{X}(t+T)\big) - e^{-\gamma T}V^{(k+1)}\big(\boldsymbol{X}(t+T)\big) \\ &\qquad = -\int_{t}^{t+T} e^{-\gamma[\tau-(t+T)]}\Big[\boldsymbol{X}^{T}\boldsymbol{Q}\boldsymbol{X} + \boldsymbol{u}_{irl}^{(k)^{T}}\boldsymbol{R}\boldsymbol{u}_{irl}^{(k)} + 2\boldsymbol{u}_{irl}^{(k+1)^{T}}\boldsymbol{R}\boldsymbol{\epsilon}(\tau)\Big]d\tau \end{aligned} \tag{25}$$

- On convergence, set $k \rightarrow (k+1)$.

Step 3. *Go to 2 until* $\boldsymbol{V}$ *and* $\boldsymbol{u}_{irl}$ *converge.*

Remark 2: The behavior policy $\overline{\boldsymbol{u}}_{irl}^{(k)}$ is used to collect data trajectories and is intentionally perturbed by an exploration noise term $\boldsymbol{\epsilon}(\tau)$. This policy differs from the control policy $\boldsymbol{u}_{irl}^{(k)}$, which is iteratively refined to approximate the optimal policy $\boldsymbol{u}_{irl}^{*}$. While $\overline{\boldsymbol{u}}_{irl}^{(k)}$ ensures sufficient state-action space exploration, $\boldsymbol{u}_{irl}^{(k)}$ is updated through value function evaluation and policy improvement. The solution to Eq. (22) is obtained via proper data collection. Since the NNs approximation requires a unique solution, Eq. (25) can be reformulated for a sufficiently small sampling interval T as:

$$\begin{aligned} V^{(k+1)}\big(\boldsymbol{X}(t+T)\big) - V^{(k+1)}\big(\boldsymbol{X}(t)\big) &= -\int_{t}^{t+T}\Big(\boldsymbol{X}^{T}\boldsymbol{Q}\boldsymbol{X} + \boldsymbol{u}_{irl}^{(k)^{T}}\boldsymbol{R}\boldsymbol{u}_{irl}^{(k)}\Big)d\tau \\ + \int_{t}^{t+T}\gamma V^{(k+1)}\big(\boldsymbol{X}(t)\big)d\tau + \int_{t}^{t+T} 2\boldsymbol{u}_{irl}^{(k+1)^{T}}\boldsymbol{R}\Big[\boldsymbol{u}_{irl}^{(k)}(\boldsymbol{X}) - \overline{\boldsymbol{u}}_{irl}^{(k)}(\boldsymbol{X})\Big]d\tau & \end{aligned} \tag{26}$$

The value function and the optimal control signal can typically be approximated by NNs with arbitrary accuracy, provided a sufficient number of neurons are used, as expressed below [9, 12]:

$$V\big(\boldsymbol{X}(t)\big) = \boldsymbol{W}_c\boldsymbol{\phi}\big(\boldsymbol{X}(t)\big) + \boldsymbol{\varepsilon}_c\big(\boldsymbol{X}(t)\big) \text{ and } \boldsymbol{u}_{irl}\big(\boldsymbol{X}(t)\big) = \boldsymbol{W}_a^T\boldsymbol{\theta}\big(\boldsymbol{X}(t)\big) + \boldsymbol{\varepsilon}_a(\boldsymbol{X}(t)) \tag{27}$$

Where, $\boldsymbol{W}_c^T \in \mathbb{R}^{l_c\times 1}; \boldsymbol{W}_a = [W_{a1}, W_{a2}, \dots, W_{am}] \in \mathbb{R}^{l_a\times m}$ are the ideal weight vectors of the critic and actor NNs, respectively. $\boldsymbol{\phi}\big(\boldsymbol{X}(t)\big) \in \mathbb{R}^{l_c}$ and $\boldsymbol{\theta}\big(\boldsymbol{X}(t)\big) \in \mathbb{R}^{l_a}$ are column vectors of linearly independent activation functions. The terms $\boldsymbol{\varepsilon}_c\big(\boldsymbol{X}(t)\big)$, $\boldsymbol{\varepsilon}_a(\boldsymbol{X}(t))$ denote the NNs approximation errors, satisfying $\lim\limits_{l_a\to\infty}\boldsymbol{\varepsilon}_a\big(\boldsymbol{X}(t)\big) = 0$, $\lim\limits_{l_c\to\infty}\boldsymbol{\varepsilon}_c\big(\boldsymbol{X}(t)\big) = 0$. Following the approach in [9], the value function $V^{(k)}\big(\boldsymbol{X}(t)\big)$ and the optimal control input $\boldsymbol{u}_{irl}^{(k)}(\tau)$ can be expressed as:

$$V^{(k)}(\boldsymbol{X}(t)) = \widehat{\boldsymbol{W}}_c^{(k)^T}\boldsymbol{\phi}\big(\boldsymbol{X}(t)\big) \text{ and } \boldsymbol{u}_{irl}^{(k)}(\tau) = \widehat{\boldsymbol{W}}_a^{(k)^T}\boldsymbol{\theta}\big(\boldsymbol{X}(t)\big) \tag{28}$$

Where $\widehat{\boldsymbol{W}}_c, \widehat{\boldsymbol{W}}_a$ are the estimated values of the ideal NN weights $\widehat{\boldsymbol{W}}_c^{(k)}$ and $\widehat{\boldsymbol{W}}_a^{(k)}$, respectively. Implementing the data collection at time instants $t = t_0 < \cdots < t_N$ the updated laws of $\widehat{\boldsymbol{W}}_c, \widehat{\boldsymbol{W}}_a$ can be given using equation (26) with $t = t_{m-1}; t + T = t_m; m = \overline{1,N}$. N is the number of neurons.

The equation (26) can be modified as:

$$\begin{aligned}
&\boldsymbol{\vartheta}^{(k)}\left(\boldsymbol{X}(t_{l-1}), \boldsymbol{u}_{irl}^{(k)}, \boldsymbol{X}(t_l)\right) \\
&\quad = \big[\boldsymbol{\phi}\big(\boldsymbol{X}(t_{l-1})\big) - \boldsymbol{\phi}\big(\boldsymbol{X}(t_l)\big)\big]^T\widehat{\boldsymbol{W}}_c^{(k+1)} + \int_{t_{l-1}}^{t_l}\gamma\big[\boldsymbol{\phi}\big(\boldsymbol{X}(\tau)\big)\big]^T\widehat{\boldsymbol{W}}_c^{(k+1)}d\tau \\
&+2\int_{t_{l-1}}^{t_l}\Big[\boldsymbol{u}_{irl}^{(k)^T}\otimes\boldsymbol{\theta}^T\big(\boldsymbol{X}(\tau)\big)\Big][\boldsymbol{R}\otimes\boldsymbol{I}^{l_a\times l_a}]\text{vec}\left(\widehat{\boldsymbol{W}}_a^{(k+1)}\right)d\tau - \int_{t_{l-1}}^{t_l}\Big[\boldsymbol{X}^T\boldsymbol{Q}\boldsymbol{X} + \boldsymbol{u}_{irl}^{(k)^T}\boldsymbol{R}\boldsymbol{u}_{irl}^k\Big]d\tau
\end{aligned} \tag{29}$$

To obtain the updated laws of $\widehat{\boldsymbol{W}}_c^{(k)}$, $\widehat{\boldsymbol{W}}_a^{(k)}$, we give an optimization method, and the formula (29) can be rewritten as:

$$\vartheta^{(m,k)}\left(\boldsymbol{X}(t_{l-1}), \overline{\boldsymbol{u}}_{irl}^{(k)}, \boldsymbol{X}(t_l)\right) = \boldsymbol{v}_1^{(m,k)}\widehat{\boldsymbol{W}}^{(k+1)} - v_2^{(m,k)} \tag{30}$$

Where: $\widehat{\boldsymbol{W}}^{(k+1)} = \left[\widehat{\boldsymbol{W}}_c^{(k+1)^T}, \widehat{\boldsymbol{W}}_{a1}^{(k+1)^T}, \dots, \widehat{\boldsymbol{W}}_{am}^{(k+1)^T}\right]^T$

$\boldsymbol{v}_1^{(m,k)} = [\boldsymbol{\rho}_1 + \boldsymbol{\rho}_2, \boldsymbol{\rho}_3 - \boldsymbol{\rho}_4]; v_2^{(m,k)} = \int_{t_{l-1}}^{t_l}[\boldsymbol{X}^T\boldsymbol{Q}\boldsymbol{X} + \boldsymbol{u}_{irl}^{(k)^T}\boldsymbol{R}\boldsymbol{u}_{irl}^k]d\tau;$

$\boldsymbol{\rho}_1 = \boldsymbol{\phi}\big(\boldsymbol{X}(t_{l-1})\big) - \boldsymbol{\phi}\big(\boldsymbol{X}(t_l)\big)\,; \boldsymbol{\rho}_2 = \int_{t_{l-1}}^{t_l}\gamma\big[\boldsymbol{\phi}\big(\boldsymbol{X}(\tau)\big)\big]^T\widehat{\boldsymbol{W}}_c^{(k+1)}d\tau$

$$\boldsymbol{\rho}_3 = 2\int_{t_{l-1}}^{t_l}\Big[\boldsymbol{u}_{irl}^{(k)^T}\otimes\boldsymbol{\theta}^T\big(\boldsymbol{X}(\tau)\big)\Big][\boldsymbol{R}\otimes\boldsymbol{I}^{l_a\times l_a}]\text{vec}\left(\widehat{\boldsymbol{W}}_a^{(k+1)}\right)d\tau$$

$$\boldsymbol{\rho}_4 = 2\int_{t_{l-1}}^{t_l}\Big[\overline{\boldsymbol{u}}_{irl}^{(k)^T}\otimes\boldsymbol{\theta}^T\big(\boldsymbol{X}(\tau)\big)\Big][\boldsymbol{R}\otimes\boldsymbol{I}^{l_a\times l_a}]\text{vec}(\widehat{\boldsymbol{W}}_a^{(k+1)})d\tau$$

In equation (30), m denotes the data sample within the collected trajectory set, while k refers to the current policy iteration. $\vartheta^{(m,k)}$ represents the temporal-difference (TD) error corresponding to the m^{th} state transition under the k^{th} policy estimate. The matrices $\boldsymbol{v}_1^{(m,k)}$ are constructed from the basis functions $\boldsymbol{\theta}(\boldsymbol{X})$ and the instantaneous cost function $\pi(\boldsymbol{X}, \boldsymbol{u}_{irl})$ associated with this transition. By stacking all *N* samples, the regression problem described in (31), (32) is formulated to update the critic weights $\widehat{\boldsymbol{W}}^{(k+1)}$, ensuring that the TD error is minimized over the entire dataset.

Implementing Eq. (30) with $m = \overline{1,N}$ we obtain the following equation:

$$\boldsymbol{\vartheta}^{(k)} = \boldsymbol{v}_1^{(k)}\widehat{\boldsymbol{W}}^{(k+1)} - \boldsymbol{v}_2^{(k)} \tag{31}$$

Where:

$$\boldsymbol{v}_1^{(k)} = \left[\boldsymbol{v}_1^{(1,k)^T}, \boldsymbol{v}_1^{(2,k)^T}, \dots, \boldsymbol{v}_1^{(N,k)^T}\right]^T;\ \boldsymbol{v}_2^{(k)} = \left[v_2^{(1,k)}, v_2^{(2,k)}, \dots, v_2^{(N,k)}\right]^T \tag{32}$$

Based on the approximation in (26), the Least-Squares method can be employed to estimate the matrix $\widehat{\boldsymbol{W}}^{(k+1)}$, thereby deriving the updated law that minimizes the residual error as:

$$\widehat{\boldsymbol{W}}^{(k+1)} = \boldsymbol{v}_1^{(k)^T}\boldsymbol{v}_2^{(k)}\left(\boldsymbol{v}_1^{(k)^T}\boldsymbol{v}_1^{(k)}\right)^{-1} \tag{33}$$

Remark 3: The optimal control for USV trajectory tracking was derived under specific conditions with an explicit relation between the weighting matrices in the cost function. However, when dynamic uncertainties and a general exponential performance index in (11) are considered, off-policy algorithms are introduced to establish a model-free optimal control framework. The stability of the proposed control scheme is verified using Lyapunov stability theory, where the Lyapunov candidate is constructed based on the Bellman function.

3. RESULTS AND DISCUSSION

Simulations of a USV system are performed by using MATLAB Simulink 24b. The parameters of the USV system are adopted in [4]. Simulation parameters are presented in table 1.

The mass of the USV is $m = 21kg$, its length and width are $1.2m$ and $0.3m$, respectively. The certainties of inertia, Coriolis centripetal, and damping matrices are given respectively by:

$$\boldsymbol{M} = \begin{bmatrix} 20 & 0 & 0 \\ 0 & 19 & 0.72 \\ 0 & 0.72 & 2.7 \end{bmatrix};\ \boldsymbol{C} = \begin{bmatrix} 0 & 0 & -19v_y - 0.72v_\psi \\ 0 & 0 & 20v_x \\ 19v_y + 0.72v_\psi & -20v_x & 0 \end{bmatrix}$$

$$\boldsymbol{D} = \begin{bmatrix} 0.72 + 1.3|v_x| + 5.8v_x^2 & 0 & 0 \\ 0 & 0.86 + 36|v_y| + 3|v_\psi| & -0.1 - 2|v_y| + 2|v_\psi| \\ 0 & -0.1 - 5|v_y| + 3|v_\psi| & 6 + 4|v_y| + 4|v_\psi| \end{bmatrix}$$

Table 1. *Initial conditions and various parameters for the control algorithm.*

Initial conditions	$\boldsymbol{\eta}^{(0)}, \boldsymbol{v}^{(0)}, l_c, l_a, \boldsymbol{W}_c^{(0)}, \boldsymbol{W}_a^{(0)}$	$[0.5;\ 0.1;\ 0.2]^T, [0.1;\ 0.2;\ 0.3]^T,$ $35,\ 12,\ \text{zeros}(35,1),\ \text{zeros}(12,3)$
Parameters for (10)	$\gamma, \boldsymbol{Q}_e, \boldsymbol{R}$	$0.5,\ 20 \times \boldsymbol{I}_{3\times3},\ 0.25 \times \boldsymbol{I}_{3\times3}$
Gains sliding surface	$\boldsymbol{\lambda}$	$\text{diag}(5\ 4\ 25)$
Design trajectory	$\boldsymbol{\eta}_{ref}$	$[10\sin 0.2t\,;\ -10\cos 0.2t\,;\ 0.2t]^T$
The active functions	$\boldsymbol{\theta}$, $\boldsymbol{\phi}$	$[x_1\ x_2 \dots x_{12}]^T \in \mathbb{R}^{12}$, $[x_1^2\ x_2^2 \dots x_{12}^2\ x_1x_2\ x_2x_3 \dots x_{11}x_{12}\ x_1^4\ x_2^4 \dots x_{12}^4]^T \in \mathbb{R}^{35}$
Disturbance	$\boldsymbol{d}(t)$	$\begin{bmatrix} 2.5 + 0.6\sin 0.5t + 0.5\cos\left(0.2t + \frac{\pi}{4}\right) \\ 3 + 0.4\sin\left(0.2t + \frac{\pi}{6}\right) + 0.2\cos 0.4t \\ 2 + \sin\left(0.6t + \frac{\pi}{6}\right) \end{bmatrix}^T$

Admissible control signal input: $\boldsymbol{u}(t) = -10(\boldsymbol{C} + \boldsymbol{D})\left[\boldsymbol{v}(t);\ \boldsymbol{J}^T\left(\eta(t)\right);\ \boldsymbol{s}(t)\right]$

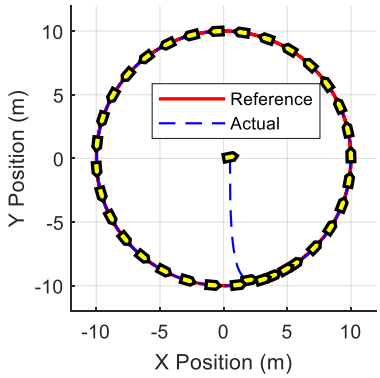

Figure 3. *The trajectory of a USV under the IRL-PI controller.*

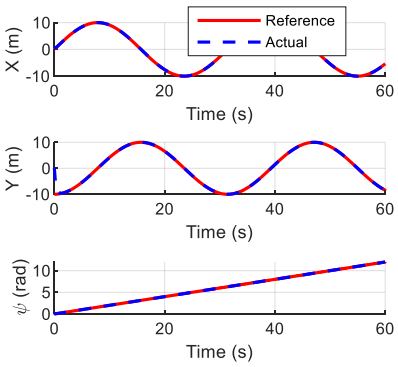

Figure 4. *Tracking trajectory in the (x, y, ψ) axes of USV under IRL-PI.*

The tracking control results are depicted in figure 3, figure 4, which show the trajectory tracking performance of the USV compared to the reference trajectory.

Table 2. *Quantitative comparison between IRL-PI and AOBC.*

Controller	RMSE		Convergence times (s)
	$E_{xy}(m)$	$E_{\psi}(rad)$	
AOBC [4]	0.2618	0.1927	N.A
IRL-PI (proposed)	0.06753	0.00181	< 2

As shown in table 2, the proposed IRL-PI controller achieves significantly lower RMSE in all three motion states compared to AOBC [4]. These improvements prove the effectiveness of robustness with IRL-based optimality.

Figure 5, figure 6 show the tracking errors and their distribution on the (x, y, ψ) axes. Figure 7 shows the response of control inputs using the proposed controller, and figure 8 shows the disturbances (with strong gusts of wind at 12 - 13 s and 27 - 28 s), which operate throughout the entire simulation. Simulation results demonstrate that the proposed IRL-PI controller achieves superior disturbance rejection.

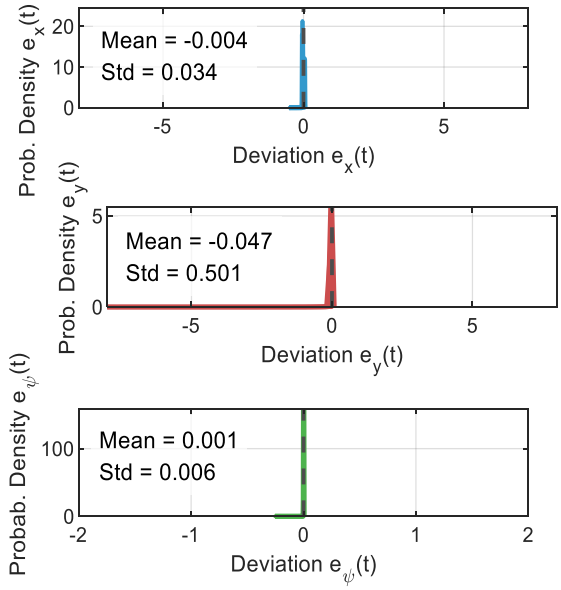

Figure 5. *The tracking error distribution.*

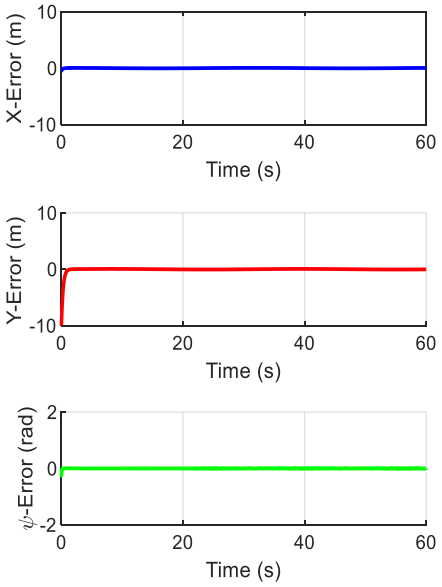

Figure 6. *The tracking errors on the axis.*

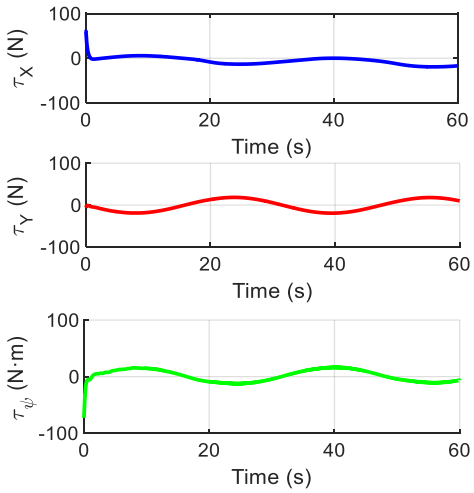

Figure 7. *The control inputs of IRL-PI.*

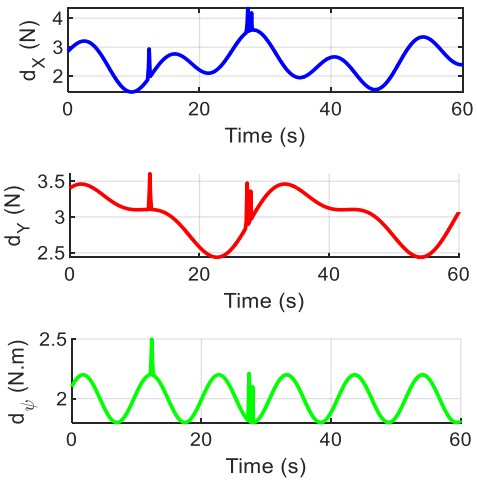

Figure 8. *Disturbance with gusts of wind.*

Figure 9 and figure 10 show the convergence behavior of the actor and critic neural network weights under the IRL method. After the 2nd iteration, the weights of the NNs converge.

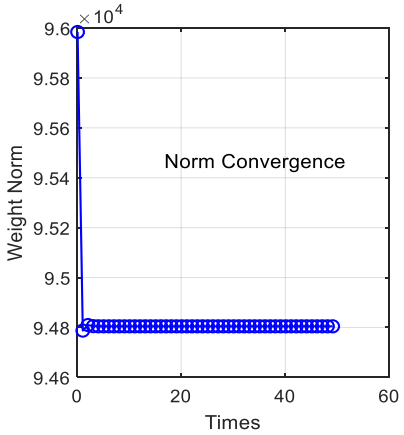

Figure 9. *Norm convergence of the ACNN weights.*

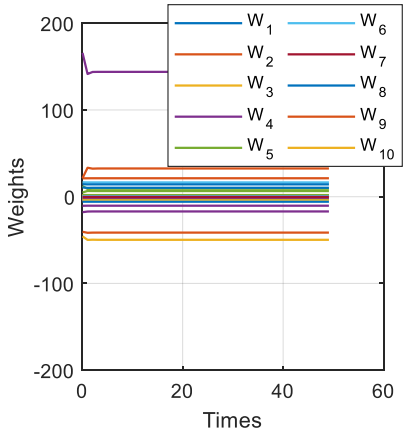

Figure 10. *Weights convergence with the IRL technique.*

The analysis of the IRL-PI controller highlights their learning efficiency, robustness, and overall tracking performance. In addition, the controller demonstrates satisfactory trajectory tracking accuracy under simulation conditions, suggesting that it is capable of learning an effective policy for the USV trajectory tracking task, consistently outperforming the other controllers in terms of convergence speed and robustness in affected external disturbances and dynamic uncertainties.

4. CONCLUSIONS

This study presents a model-free IRL-PI framework for optimal and coordinated trajectory tracking of USVs under unknown dynamics and time-varying disturbances. The method removes the need for model knowledge and ensures robustness and tracking precision. By applying an order-reduction technique with a sliding surface, the algorithm efficiently solves the infinite-horizon optimal control problem. Simulation results verify its superior robustness, accuracy, and stability, demonstrating that the proposed approach effectively unifies optimal and coordination control for USVs. Future work will focus on experimental validation and extend the practical deployment of the proposed controller for USVs.

REFERENCES

[1]. T. I. Fossen, *"Handbook of marine craft hydrodynamics and motion control"*, John Wiley & Sons Ltd., (2011).

[2]. X. Lin, H. Jiang, et al., *"Adaptive sliding-mode trajectory tracking control for underactuated surface vessels based on NDO"*, Proceedings of IEEE International Conference on Mechatronics and Automation (ICMA), pp. 1043–1049, (2018).

[3]. C. Liu et al., *"Trajectory tracking of underactuated surface vessels based on neural network and hierarchical sliding mode"*, Journal of Marine Science and Technology, vol. 20, pp. 322–330, (2015).

[4]. G. Wen et al., *"Adaptive tracking control of surface vessel using optimized backstepping technique"*, IEEE Transactions on Cybernetics, Vol. 49, No. 9, pp. 3420–3431, (2018).

[5]. G. Xiao, H. Zhang, Y. Luo, H. Jiang, *"Data-driven optimal tracking control for a class of affine nonlinear continuous-time systems"*, (2016).

[6]. V. T. Vu, T. L. Pham, Q. H. Tran, P. N. Dao, *"Optimal control for fully-actuated surface vessel systems"*, iRobotics, Vol. 4, No. 1, (2021).

[7]. C. Liu, et al., *"Trajectory tracking control for underactuated surface vessels based on nonlinear model predictive control"*, Lecture Notes in Computer Science (ICCL), Vol. 9335, pp. 166–180, (2015).

[8]. K. Kamalapurkar, W. E. Dixon, et al., *"Model-based reinforcement learning for infinite-horizon approximate optimal tracking"*, IEEE Transactions on Neural Networks and Learning Systems, vol. 28, pp. 753–758, (2016).

[9]. X. Guo, W. Yan, R. Cui, *"Integral reinforcement learning-based adaptive neural network control for continuous-time nonlinear MIMO systems with unknown control directions"*, IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 50, No. 11, pp. 4068–4077, (2019).

[10]. Z. Zheng, et al., *"Reinforcement learning control for underactuated surface vessel with output error constraints and uncertainties"*, Neurocomputing, Vol. 399, pp. 479–490, (2020).

[11]. X. Yang, et al., *"Adaptive dynamic programming for robust neural control of unknown continuous-time nonlinear systems"*, IET Control Theory & Applications, Vol. 11, pp. 2307–2316, (2017).

[12]. Y. Zhu, D. Zhao, X. Li, *"Using reinforcement learning techniques to solve continuous-time nonlinear optimal tracking problem without system dynamics"*, IET Control Theory & Applications, Vol. 10, pp. 1339–1347, (2016).

[13]. K. Dupree, P. M. Patre, Z. D. Wilcox, W. E. Dixon, *"Asymptotic optimal control of uncertain nonlinear Euler–Lagrange systems"*, Automatica, vol. 47, pp. 99–107, (2011).

[14]. J. Y. Lee, et al., *"Integral reinforcement learning for continuous-time input-affine nonlinear systems with simultaneous invariant explorations"*, IEEE Transactions on Neural Networks and Learning Systems, Vol. 26, pp. 916–932, (2014).

TÓM TẮT

Điều khiển tối ưu bám quỹ đạo cho USV có động lực học bất định và nhiễu biến thiên theo thời gian bằng thuật toán PI và IRL

Bài báo trình bày một khung điều khiển tối ưu phi mô hình cho bài toán bám quỹ đạo của tàu mặt nước không người lái (USVs) hoạt động trong điều kiện động lực học chưa biết và nhiễu biến thiên theo thời gian, được phát triển thông qua thuật toán Học tăng cường tích phân (IRL) và lặp chính sách (PI). Bộ điều khiển IRL-PI được thiết kế dựa trên kỹ thuật giảm bậc và cấu trúc mạng nơ-ron Actor-Critic chính sách ngoại tuyến, cho phép xấp xỉ nghiệm phương trình Hamilton-Jacobi-Bellman (HJB) trong thời gian thực mà không cần biết trước mô hình hệ thống. Kết quả mô phỏng trên mô hình USV ba bậc tự do (3-DOF) cho thấy phương pháp được đề xuất vượt trội hơn các bộ điều khiển truyền thống về cả độ chính xác bám quỹ đạo và tính bền vững. Những kết quả này khẳng định tiềm năng của bộ điều khiển IRL-PI trong việc phát triển các giải pháp điều khiển bền vững cho các hệ thống hàng hải phức tạp hoạt động trong môi trường bất định và biến động.

Từ khoá: Học tăng cường tích phân; Lặp chính sách; Điều khiển tối ưu; HJB; USVs.