Model-Reference Reinforcement Learning for Collision-Free Tracking Control of Autonomous Surface Vehicles

Zhang, Qingrui; Pan, Wei; Reppa, Vasso

DOI
10.1109/TITS.2021.3086033

Publication date
2021

Document Version
Final published version

Published in
IEEE Transactions on Intelligent Transportation Systems

Model-Reference Reinforcement Learning for Collision-Free Tracking Control of Autonomous Surface Vehicles

Qingrui Zhang, *Member, IEEE*, Wei Pan, *Member, IEEE*, and Vasso Reppa, *Member, IEEE*

***Abstract*— This paper presents a novel model-reference reinforcement learning algorithm for the intelligent tracking control of uncertain autonomous surface vehicles with collision avoidance. The proposed control algorithm combines a conventional control method with reinforcement learning to enhance control accuracy and intelligence. In the proposed control design, a nominal system is considered for the design of a baseline tracking controller using a conventional control approach. The nominal system also defines the desired behaviour of uncertain autonomous surface vehicles in an obstacle-free environment. Thanks to reinforcement learning, the overall tracking controller is capable of compensating for model uncertainties and achieving collision avoidance at the same time in environments with obstacles. In comparison to traditional deep reinforcement learning methods, our proposed learning-based control can provide stability guarantees and better sample efficiency. We demonstrate the performance of the new algorithm using an example of autonomous surface vehicles.**

***Index Terms*— Autonomous surface vehicles, reinforcement learning, collision avoidance, control architecture.**

I. Introduction

AUTONOMOUS surface vehicles (ASVs) have attracted extensive research attention, due to their advantages in applications, such as environmental monitoring [1], resource exploration [2], waterborne transportation [3], and many more [4]. Successful launch of ASVs in real life requires avoiding collisions with obstacles [5] and accurate tracking along a desired trajectory [6]. Both collision avoidance and tracking control are the major research topics for ASVs in the maritime engineering [7]–[10]. However, tracking control for ASVs in the presence of obstacles is challenging, as ASVs are subject to uncertain nonlinear hydrodynamics and unknown environmental disturbances [11]. Due to the complexity of the problem, collision avoidance and tracking control are mostly studied in a separate manner.

Manuscript received 14 August 2020; revised 19 February 2021 and 9 April 2021; accepted 28 May 2021. Date of publication 15 June 2021; date of current version 8 July 2022. This work was supported by the Cohesion Project by the Faculty of Mechanical, Maritime and Material Engineering, Delft University of Technology. The Associate Editor for this article was J. Alvarez. *(Corresponding authors: Vasso Reppa; Wei Pan.)*

Qingrui Zhang is with the School of Aeronautics and Astronautics, Sun Yat-sen University, Guangzhou 510275, China, and also with the Faculty of Mechanical, Maritime and Material Engineering (3mE), Delft University of Technology, 2628CD Delft, The Netherlands (e-mail: zhangqr9@mail.sysu.edu.cn).

Wei Pan and Vasso Reppa are with the Faculty of Mechanical, Maritime and Material Engineering (3mE), Delft University of Technology, 2628CD Delft, The Netherlands (e-mail: wei.pan@tudelft.nl; v.reppa@tudelft.nl).

Digital Object Identifier 10.1109/TITS.2021.3086033

Collision avoidance methods for ASVs are categorized into motion planning approaches [12]–[14] and optimization-based algorithms [9], [15]. In the path/motion planning approaches, a collision-free reference trajectory or motion is generated based on either off-line or on-line methods, e.g., A* [16], RRT* [12], potential field methods [17], and multi-objective optimization [14], etc. It is assumed that the generated collision-free reference trajectory can be tracked with high accuracy by the ASV based on a well-designed control module. Thus, collision avoidance following the motion planning approaches may fail for uncertain systems that lack valid tracking controllers. Due to the two-module design feature, there always exists a time delay for the ASV to apply collision avoidance actions, as the inner-loop controller needs time to react to changes in the reference trajectories. Such a time delay will also downgrade the performance of the motion planning approaches in environments with fast-moving obstacles.

The optimization-based algorithms can directly find a control law with collision avoidance by optimizing a certain objective function, e.g. model predictive control (MPC) [9] and reinforcement learning (RL) [18]. They potentially have a better performance than the motion planning approaches in dynamic environments. However, collision avoidance algorithms based on MPC suffer from high computational complexity and rely on accurate modeling of ASV systems [9]. They will, therefore, experience dramatic degradation in performances for uncertain ASVs. In comparison to MPC, RL can learn an intelligent collision avoidance law from data samples [18], [19], which can significantly reduce the dependence on modeling efforts and thus make RL very suitable for uncertain ASVs.

Tracking control algorithms for uncertain systems including ASVs mainly lie in four categories: 1) robust control that is the "worst-case" design for bounded uncertainties and disturbances [20]; 2) adaptive control that estimates uncertainty parameters [6], [21]; 3) disturbance observer (DO)-based control that compensates uncertainties and disturbances in terms of the observation technique [22]–[24]; and 4) reinforcement learning (RL) that learns a control law from data samples [25]. In robust control, uncertainties and disturbances are assumed to be bounded with known boundaries [26]. As a consequence, robust control may lead to conservative high-gain control laws that degrade the control performances (i.e., overshoot, settling time, and stability margins) [27]. Adaptive control can handle varying uncertainties with unknown boundaries,

TABLE I
REINFORCEMENT LEARNING-BASED CONTROL METHODS FOR ASVS

	Algorithms	Design considerations			
		Model uncertainties	Collision avoidance	Closed-loop stability	Combination with model-based control
DDPG	Woo et. al., 2019 [36]	✓	✗	✗	✗
	Shi et. al., 2019 [25]	✓	✗	✗	✗
PPO	Meyer et. al., 2020 [18]	✓	✓	✗	✗
DQN	Woo et. al., 2020 [40]	✓	✓	✗	✗
	Xu et. al., 2020 [41]	✓	✓	✗	✗
ADP	Cui et. al., 2017 [42]	✓	✗	✓	✗
	Wang et. al., 2020 [43]	✓	✗	✓	✗
Our method		✓	✓	✓	✓

but system uncertainties are assumed to be linearly parameterized with known structure and unknown parameters [28], [29]. DO-based control can adapt to both uncertainties and disturbances with unknown structures [30], [31]. However, the frequency information of uncertainty and disturbance signals are necessary in the DO-based control for choosing proper control gains, otherwise, it is highly possible to end up with a high-gain control law [30], [32]. In general, comprehensive modeling and analysis of systems are essential for all model-based methods.

In comparison to existing model-based methods, RL is capable of learning a complex tracking control law with collision avoidance from data samples using much less model information [19], [33]–[35]. It is, therefore, more promising in controlling systems subject to massive uncertainties and disturbances as ASVs [25], [36] and meanwhile achieving collision avoidance [18], given the sufficiency and good quality of collected data. Nevertheless, it is challenging for model-free RL to ensure closed-loop stability, though some research attempts have been made. Model-based RL with stability guarantee has been investigated by introducing a Lyapunov constraint into the objective function [37]. However, the model-based RL with stability guarantees requires an admissible control law — a control law that makes the original system asymptotically stable — for the initialization. Both the Lyapunov candidate function and system dynamics are assumed to be Lipschitz continuous with known Lipschitz constants for the construction of the Lyapunov constraint. It is challenging to find the Lipschitz constant of an uncertain system. Therefore, the introduced Lyapunov constraint function is restrictive, as it is established based on the worst-case consideration [37].

With the consideration of the merits and limitations of existing RL methods, we propose a novel learning-based control algorithm for uncertain ASVs with collision avoidance by combining a conventional control method with deep RL in this paper. The proposed learning-based control design, therefore, consists of two components: a baseline control law that stabilizes a nominal ASV system and a deep RL control law that compensates for system uncertainties and also achieves intelligent collision avoidance. Such a design structure has several advantages over both conventional model-based methods and pure deep RL methods. First of all, in relation to the "model-free" feature of deep RL, we can learn from data samples a control law that directly compensates for system uncertainties without exploiting their structures, boundaries, or frequencies [38]. Intelligent collision avoidance can also be learned by the deep RL. Second, closed-loop stability is guaranteed by the overall learned control law for the tracking control in obstacle-free environments, if the baseline control law can stabilize the ASV system at least locally. Our proposed design can avoid exploiting the Lipschitz constant of the overall system and potentially produce less conservative results. Lastly, the proposed design is potentially more sample efficient than an RL algorithm learning from scratch – that is, fewer data samples are needed for training. In RL, a system learns from mistakes, demanding a lot of trials and errors. In our design, the baseline control that can stabilize the overall system, can help to exclude unnecessary mistakes, thereby providing a good starting point for the RL training. The main contributions of this paper are summarized below.

1) A new design methodology is proposed for the learning-based control of ASV systems, leveraging the advantages of both model-based control methods and RL methods.
2) A model-reference RL algorithm is developed for the collision-free tracking control of uncertain ASVs. The proposed model-reference RL algorithm does not require any information aboutthe structures, boundaries, or frequencies of uncertainties. It is potentially more efficient than an RL algorithm that learns from scratch. Closed-loop stability is guaranteed.
3) The proposed model-reference RL algorithm is analyzed rigorously. Mathematical proofs are provided for the convergence analysis. Closed-loop stability is analyzed for the tracking control at obstacle-free environments.

Some research results can be found in [39]. In [39], the collision avoidance problem is not addressed. There are no mathematical proofs for the convergence analysis. In this paper, we present more details on the problem formulation and algorithm design, including the choices of the control policies in RL, discussions of reward functions, descriptions of the deep neural networks, and proof of closed-loop stability. Brief comparison between the existing methods and our method is summarized in Table I.

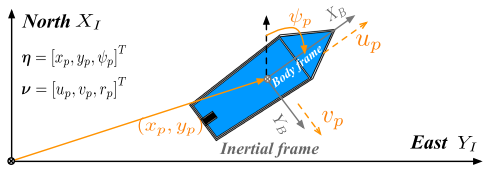


Fig. 1. Coordinate systems of an autonomous surface vehicle.

The rest of the paper is organized as follows. In Section II, we present the ASV dynamics. The model-reference reinforcement learning control is formulated at length in Section III, including the problem formulation, basic concepts of reinforcement learning, and choices of reward functions. In Section IV, the model-reference reinforcement Learning is developed based deep neural networks. Section V presents the details on the analysis of the proposed model-reference reinforcement learning algorithm, including the convergence analysis and stability analysis. Section VI provides the simulation results of the application of the algorithm to an example of ASVs. Conclusion remarks are given in Section VII.

Notations: Matrix transpose is denoted by the superscript "T". $\|\cdot\|_2$ denotes the Euclidean norm. $\mathbb{R}$ is the set of real numbers, and accordingly, $\mathbb{R}^n$ is an n-dimensional vector composed of real numbers. $\mathbb{E}[\cdot]$ is an expectation operator. "$\sum$" represents the summation operation.

II. Autonomous Surface Vehicle Dynamics

As shown in Fig. 1, $x_p \in \mathbb{R}$ and $y_p \in \mathbb{R}$ are the horizontal positions of an ASV in the inertial frame and $\psi_p \in \mathbb{R}$ the heading angle. In the body frame, $u_p \in \mathbb{R}$ and $v_p \in \mathbb{R}$ are the linear velocities in surge (x-axis) and sway (y-axis), respectively. The heading angular rate is denoted by $r_p \in \mathbb{R}$. The general 3-DOF nonlinear dynamics of an ASV are

$$\begin{cases} \dot{\boldsymbol{\eta}} = \boldsymbol{R}(\boldsymbol{\eta})\,\boldsymbol{v} \\ \boldsymbol{M}\dot{\boldsymbol{v}} + (\boldsymbol{C}(\boldsymbol{v}) + \boldsymbol{D}(\boldsymbol{v}))\,\boldsymbol{v} + \boldsymbol{G}(\boldsymbol{v}) = \boldsymbol{\tau} \end{cases} \tag{1}$$

where $\boldsymbol{\eta} = \left[x_p, y_p, \psi_p\right]^T \in \mathbb{R}^3$ is a generalized coordinate vector, $\boldsymbol{v} = \left[u_p, v_p, r_p\right]^T \in \mathbb{R}^3$ is the speed vector, $\boldsymbol{M}$ is the inertia matrix, $\boldsymbol{C}(\boldsymbol{v})$ denotes the matrix of Coriolis and centripetal terms, $\boldsymbol{D}(\boldsymbol{v})$ is the damping matrix, $\boldsymbol{\tau} = [\tau_u, \tau_v, \tau_r] \in \mathbb{R}^3$ represents the control forces and moments, $\boldsymbol{G}(\boldsymbol{v}) = \left[\boldsymbol{g}_1(\boldsymbol{v}), \boldsymbol{g}_2(\boldsymbol{v}), \boldsymbol{g}_3(\boldsymbol{v})\right]^T \in \mathbb{R}^3$ denotes unmodeled dynamics due to gravitational, buoyancy, and environmental forces and moments [11], and $\boldsymbol{R}$ is a rotation matrix given by

$$\boldsymbol{R}(\boldsymbol{\eta}) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The inertia matrix $\boldsymbol{M} = \boldsymbol{M}^T > 0$ is

$$\boldsymbol{M} = [M_{ij}] = \begin{bmatrix} M_{11} & 0 & 0 \\ 0 & M_{22} & M_{23} \\ 0 & M_{32} & M_{33} \end{bmatrix} \tag{2}$$

where $M_{11} = m - X_{\dot{u}}$, $M_{22} = m - Y_{\dot{v}}$, $M_{33} = I_z - N_{\dot{r}}$, $M_{32} = M_{23} = mx_g - Y_{\dot{r}}$, and $X_{(\cdot)}$, $Y_{(\cdot)}$, and $N_{(\cdot)}$ are hydrodynamic coefficients [11]. The matrix $\boldsymbol{C}(\boldsymbol{v}) = -\boldsymbol{C}^T(\boldsymbol{v})$ is

$$\boldsymbol{C} = [C_{ij}] = \begin{bmatrix} 0 & 0 & C_{13}(\boldsymbol{v}) \\ 0 & 0 & C_{23}(\boldsymbol{v}) \\ -C_{13}(\boldsymbol{v}) & -C_{23}(\boldsymbol{v}) & 0 \end{bmatrix} \tag{3}$$

where $C_{13}(\boldsymbol{v}) = -M_{22}v - M_{23}r$, $C_{23}(\boldsymbol{v}) = M_{11}u$. The damping matrix $\boldsymbol{D}(\boldsymbol{v})$ is

$$\boldsymbol{D}(\boldsymbol{v}) = [D_{ij}] = \begin{bmatrix} D_{11}(\boldsymbol{v}) & 0 & 0 \\ 0 & D_{22}(\boldsymbol{v}) & D_{23}(\boldsymbol{v}) \\ 0 & D_{32}(\boldsymbol{v}) & D_{33}(\boldsymbol{v}) \end{bmatrix} \tag{4}$$

where $D_{11}(\boldsymbol{v}) = -X_u - X_{|u|u}|u| - X_{uuu}u^2$, $D_{22}(\boldsymbol{v}) = -Y_v - Y_{|v|v}|v| - Y_{|r|v}|r|$, $D_{23}(\boldsymbol{v}) = -Y_r - Y_{|v|r}|v| - Y_{|r|r}|r|$, $D_{32}(\boldsymbol{v}) = -N_v - N_{|v|v}|v| - N_{|r|v}|r|$, $D_{33}(\boldsymbol{v}) = -N_r - N_{|v|r}|v| - N_{|r|r}|r|$.

III. Problem Formulation

A. Model-Reference Control Formulation

Let $\boldsymbol{x} = \left[\boldsymbol{\eta}^T, \boldsymbol{v}^T\right]^T$ and $\boldsymbol{u} = \boldsymbol{\tau}$, so (1) can be rewritten as

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & \boldsymbol{R}(\boldsymbol{\eta}) \\ 0 & \boldsymbol{A}(\boldsymbol{v}) \end{bmatrix}\boldsymbol{x} + \begin{bmatrix} 0 \\ \boldsymbol{B} \end{bmatrix}\boldsymbol{u} + \begin{bmatrix} 0 \\ -\boldsymbol{M}^{-1}\boldsymbol{G}(\boldsymbol{v}) \end{bmatrix} \tag{5}$$

where $\boldsymbol{A}(\boldsymbol{v}) = -\boldsymbol{M}^{-1}(\boldsymbol{C}(\boldsymbol{v}) + \boldsymbol{D}(\boldsymbol{v}))$, and $\boldsymbol{B} = \boldsymbol{M}^{-1}$. Assume an accurate model (5) is not available, but it is possible to get a nominal model expressed as

$$\dot{\boldsymbol{x}}_m = \begin{bmatrix} 0 & \boldsymbol{R}(\boldsymbol{\eta}) \\ 0 & \boldsymbol{A}_m \end{bmatrix}\boldsymbol{x}_m + \begin{bmatrix} 0 \\ \boldsymbol{B}_m \end{bmatrix}\boldsymbol{u}_m \tag{6}$$

where $\boldsymbol{A}_m$ and $\boldsymbol{B}_m$ are the known system matrices, and the unmodelled dynamics $\boldsymbol{G}(\boldsymbol{v})$ ignored. Note that $\boldsymbol{A}_m$ and $\boldsymbol{B}_m$ are different from $\boldsymbol{A}(\boldsymbol{v})$ and $\boldsymbol{B}$, respectively. By ignoring all unknown nonlinear terms, we have $\boldsymbol{D}_m = diag\{-X_v, -Y_v, -N_r\}$, and $\boldsymbol{M}_m = diag\{M_{11}, M_{22}, M_{33}\}$. Hence, $A_m = -\boldsymbol{M}_m^{-1}\boldsymbol{D}_m$ and $B_m = -\boldsymbol{M}_m^{-1}$ in (6).In $\boldsymbol{A}_m$ and $\boldsymbol{B}_m$, we will ignore all unknown nonlinear terms, and obtain a linear nominal model. Assume that there exists a control law $\boldsymbol{u}_m$ allowing the states of the nominal system (6) to converge to a reference signal $\boldsymbol{x}_r$, i.e., $\|\boldsymbol{x}_m - \boldsymbol{x}_r\|_2 \to 0$ as $t \to \infty$.

The objective of the work in this paper is to design a controller allowing the state $\boldsymbol{x}$ to track state trajectories of the nominal model (6) and avoid collisions with obstacles having known states $\boldsymbol{x}_{o_i}$, where $i \in \{1, \ldots, N_o\}$ indicates the i-th obstacle. As shown in Fig. 2, the overall control structure for the ASV system (5) is

$$\boldsymbol{u} = \boldsymbol{u}_b + \boldsymbol{u}_l \tag{7}$$

where $\boldsymbol{u}_b$ is a baseline control, and $\boldsymbol{u}_l$ is a control law from the deep RL module whose design is provided in Section IV-B.

Remark 1: The baseline control $\boldsymbol{u}_b$ is employed to ensure the basic tracking performance without obstacles, (i.e., local stability of the tracking control). It can be designed using any existing method based on the nominal model (6). One potential choice for the design of $\boldsymbol{u}_b$ is the nonlinear backstepping control [31]. Hence, we ignore the design process of $\boldsymbol{u}_b$, and focus on the development of $\boldsymbol{u}_l$ with RL.

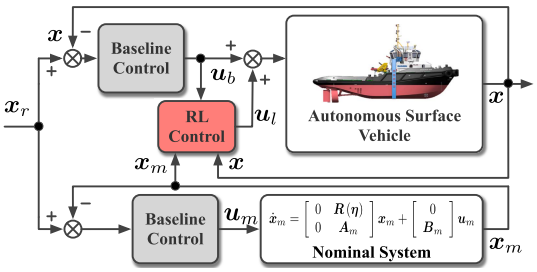


Fig. 2. Model-reference reinforcement learning control.

B. Markov Decision Process

For the formulation of RL, the ASV dynamics (5) and (6) are characterized using another mathematical model called Markov decision process that is denoted by a tuple $\mathcal{MDP} := \langle \mathcal{S},\ \mathcal{U},\ \mathcal{P},\ R,\ \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{U}$ specifies the action/input space, $\mathcal{P} : \mathcal{S} \times \mathcal{U} \times \mathcal{S} \to \mathbb{R}$ defines a transition probability, $R : \mathcal{S} \times \mathcal{U} \to \mathbb{R}$ is a reward function, and $\gamma \in [0,\ 1)$ is a discount factor. In this paper, the state vector $\boldsymbol{s} \in \mathcal{S}$ contains $\boldsymbol{x}$, $\boldsymbol{x}_m$, $\boldsymbol{u}_b$, and $\boldsymbol{x}_{o_i}$. Note that $\boldsymbol{x}_m$ is included as a state of the RL instead of $\boldsymbol{x}_r$. Hence, $\boldsymbol{s} = \left\{\boldsymbol{x}_m, \boldsymbol{x}, \boldsymbol{u}_b, \cup_i^{N_o} \boldsymbol{x}_{o_i}\right\}$, where $\cup_i^{N_o} \boldsymbol{x}_{o_i}$ are the states of N_o obstacles detected by the ASV. More details on obstacles will be given in Section III-D.

Since RL learns the control policies using data samples, it is assumed that we can sample input and state data from system (5) at discrete time steps. Let $\boldsymbol{x}_t$, $\boldsymbol{u}_{b,t}$, and $\boldsymbol{u}_{l,t}$ be the ASV state, the baseline control action, and the control action from RL at the time step t, respectively. The union of obstacles detected by the ASV is characterized by $\cup_i^{N_o} \boldsymbol{x}_{o_i,t}$. The state signal $\boldsymbol{s}$ at the time step t is $\boldsymbol{s}_t = \left\{\boldsymbol{x}_{m,t}, \boldsymbol{x}_t, \boldsymbol{u}_{b,t}, \cup_i^{N_o} \boldsymbol{x}_{o_i,t}\right\}$.

C. Reinforcement Learning

For standard RL, the objective is to maximize an expected accumulated return described by a value function $V_{\boldsymbol{\pi}}(\boldsymbol{s}_t)$ with

$$V_{\boldsymbol{\pi}}(\boldsymbol{s}_t) = \sum_{t}^{\infty} \sum_{\boldsymbol{u}_{l,t}} \boldsymbol{\pi}\left(\boldsymbol{u}_{l,t}|\boldsymbol{s}_t\right) \sum_{\boldsymbol{s}_{t+1}} \mathcal{P}_{t+1|t}\big(R_t + \gamma\, V_{\boldsymbol{\pi}}(\boldsymbol{s}_{t+1})\big)$$

where $\mathcal{P}_{t+1|t} = \mathcal{P}\left(\boldsymbol{s}_{t+1} \left| \boldsymbol{s}_t, \boldsymbol{u}_{l,t}\right.\right)$ is the transition probability of the ASV system, $R_t = R(\boldsymbol{s}_t, \boldsymbol{u}_{l,t})$ is the reward function, $\gamma \in [0,\ 1)$ is a constant discount factor, and $\boldsymbol{\pi}\left(\boldsymbol{u}_{l,t}|\boldsymbol{s}_t\right)$ is called control policy in RL. A policy in RL, denoted by $\boldsymbol{\pi}\left(\boldsymbol{u}_{l,t}|\boldsymbol{s}_t\right)$, is the probability of choosing an action $\boldsymbol{u}_{l,t} \in \mathcal{U}$ at a state $\boldsymbol{s}_t \in \mathcal{S}$ [19]. In this paper, a Gaussian policy is used, which is

$$\boldsymbol{\pi}\left(\boldsymbol{u}_l|\boldsymbol{s}\right) = \mathcal{N}\left(\boldsymbol{u}_l\left(\boldsymbol{s}\right), \boldsymbol{\sigma}\right) \tag{8}$$

where $\mathcal{N}(\cdot,\cdot)$ denotes a Gaussian distribution with $\boldsymbol{u}_l(\boldsymbol{s})$ as the mean value and $\boldsymbol{\sigma}$ as the covariance matrix. The covariance matrix $\boldsymbol{\sigma}$ controls the exploration performance at the learning stage. For the algorithm design, we also introduce an action-value function (a.k.a., Q-function) defined by

$$Q_{\boldsymbol{\pi}}\left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t}\right) = R_t + \gamma\, \mathbb{E}_{\boldsymbol{s}_{t+1}}\left[V_{\boldsymbol{\pi}}(\boldsymbol{s}_{t+1})\right] \tag{9}$$

where $\mathbb{E}_{\boldsymbol{s}_{t+1}}[\cdot] = \sum_{\boldsymbol{s}_{t+1}} \mathcal{P}_{t+1|t}[\cdot]$ is an expectation operator.

In this paper, the deep RL is resolved based on the soft actor-critic (SAC) algorithm that provides both sample efficient learning and convergence [44]. In SAC, an entropy term is added to regulate the exploration performance at the training stage, thus resulting in a modified Q-function in (10).

$$Q_{\boldsymbol{\pi}}\left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t}\right) = R_t + \gamma\, \mathbb{E}_{\boldsymbol{s}_{t+1}}\left[V_{\boldsymbol{\pi}}(\boldsymbol{s}_{t+1}) + \alpha \mathcal{H}\left(\boldsymbol{\pi}\left(\boldsymbol{u}_{l,t+1}|\boldsymbol{s}_{t+1}\right)\right)\right] \tag{10}$$

where $\mathcal{H}\left(\boldsymbol{\pi}\left(\boldsymbol{u}_{l,t}|\boldsymbol{s}_t\right)\right) = -\sum_{\boldsymbol{u}_{l,t}} \boldsymbol{\pi}\left(\boldsymbol{u}_{l,t}|\boldsymbol{s}_t\right) \ln\left(\boldsymbol{\pi}\left(\boldsymbol{u}_{l,t}|\boldsymbol{s}_t\right)\right) = -\mathbb{E}_{\boldsymbol{\pi}}\left[\ln\left(\boldsymbol{\pi}\left(\boldsymbol{u}_{l,t}|\boldsymbol{s}_t\right)\right)\right]$ is the entropy of the policy, and α is a temperature parameter [44].

Learning process of SAC will repeatedly execute policy evaluation and policy improvement. In the policy evaluation, the Q-value in (10) is computed by applying a Bellman operation $Q_{\boldsymbol{\pi}}\left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t}\right) = \mathcal{T}^{\boldsymbol{\pi}} Q_{\boldsymbol{\pi}}\left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t}\right)$ where

$$\mathcal{T}^{\boldsymbol{\pi}} Q_{\boldsymbol{\pi}}\left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t}\right) = R_t + \gamma\, \mathbb{E}_{\boldsymbol{s}_{t+1}}\left\{\mathbb{E}_{\boldsymbol{\pi}}\left[Q_{\boldsymbol{\pi}}\left(\boldsymbol{s}_{t+1}, \boldsymbol{u}_{l,t+1}\right) - \alpha \ln\left(\boldsymbol{\pi}\left(\boldsymbol{u}_{l,t+1}|\boldsymbol{s}_{t+1}\right)\right)\right]\right\}. \tag{11}$$

In the policy improvement, the policy is updated by

$$\boldsymbol{\pi}_{new} = \arg\min_{\boldsymbol{\pi}' \in \Pi} \mathscr{D}_{KL}\left(\boldsymbol{\pi}'\left(\cdot|\boldsymbol{s}_t\right) \middle\| Z^{\boldsymbol{\pi}_{old}} e^{\frac{1}{\alpha} Q^{\boldsymbol{\pi}_{old}}(\boldsymbol{s}_t, \cdot)}\right) \tag{12}$$

where Π denotes a policy set, $\boldsymbol{\pi}_{old}$ denotes the policy from the last update, $Q^{\boldsymbol{\pi}_{old}}$ is the Q-value of $\boldsymbol{\pi}_{old}$, $\mathscr{D}_{KL}$ denotes the Kullback-Leibler (KL) divergence, and $Z^{\boldsymbol{\pi}_{old}}$ is a normalization factor. Via mathematical derivations [44], (12) is rewritten as

$$\boldsymbol{\pi}^* = \arg\min_{\boldsymbol{\pi} \in \Pi} \mathbb{E}_{\boldsymbol{\pi}}\Big[\alpha \ln\left(\boldsymbol{\pi}\left(\boldsymbol{u}_{l,t}|\boldsymbol{s}_t\right)\right) - Q\left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t}\right)\Big]. \tag{13}$$

Remark 2: Once the optimization problem (13) is resolved, we will have $\boldsymbol{\pi}^*(\boldsymbol{u}_l|\boldsymbol{s}) = \mathcal{N}\left(\boldsymbol{u}_l^*(\boldsymbol{s}), \boldsymbol{\sigma}^*\right)$ according to (8). The variance $\boldsymbol{\sigma}^*$ will be close to 0. The mean value function $\boldsymbol{u}_l^*(\boldsymbol{s})$ is the learned optimal control to avoid collisions and compensate for system uncertainties. Notably, $\boldsymbol{u}_l^*(\boldsymbol{s})$ will be approximated using deep neural networks (DNNs) that will be discussed in Section IV-A. The learning process is to find the optimal parameters of the DNN that approximates $\boldsymbol{u}_l^*(\boldsymbol{s})$.

D. Reward Functions

In our design, two objectives are defined for the ASV: trajectory tracking and collision avoidance. For the trajectory tracking, system (5) needs to track the nominal system (6), so the tracking reward $R_{t,1}$ is defined as

$$R_{t,1} = -\left(\boldsymbol{x}_t - \boldsymbol{x}_{m,t}\right)^T \boldsymbol{H}_1 \left(\boldsymbol{x}_t - \boldsymbol{x}_{m,t}\right) - \boldsymbol{u}_{l,t}^T \boldsymbol{H}_2 \boldsymbol{u}_{l,t} \tag{14}$$

where $\boldsymbol{H}_1 > 0$ and $\boldsymbol{H}_2 > 0$ are positive definite matrices.

The second objective is to avoid obstacles along the trajectory of the ASV. Fig. 3 shows variables used for the definition of the reward function for collision avoidance. All obstacles are assumed to be inscribed in a circle. For the safe radius d_{s_i}, there exists $d_{s_i} > d_{o_i} + d_a$. If $d_{ao_i} \le d_d$, the i-th obstacle is visible to the ASV, where d_d is the radius of the detection region of the ASV. Note that the obstacles could be either static or moving, so the state vector the i-th obstacle is written as $\boldsymbol{x}_{o_i} = \left[\boldsymbol{p}_{o_i}^T, \boldsymbol{v}_{o_i}^T\right]^T$, where $\boldsymbol{p}_{o_i}$ is the position of the i-th obstacle, and $\boldsymbol{v}_{o_i}$ is the velocity of the i-th obstacle.

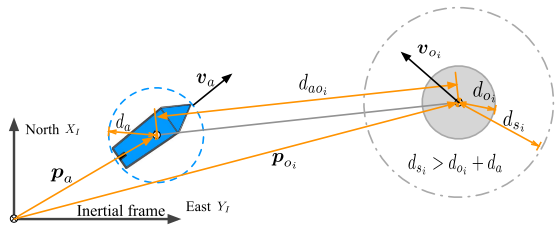


Fig. 3. Variables for collision avoidance (d_a: size of the ASV; d_{o_i}: size of the i-th obstacle; d_{s_i}: radius of the safe region; d_{ao_i}: relative distance between the ASV and the i-th obstacle).

Let $\boldsymbol{p}_a = \left[x_p, y_p\right]^T$ and $\boldsymbol{v}_a = \left[u_p, v_p\right]^T$ be the position and velocity of the ASV, respectively. For the i-th visible obstacle at the time step t, define the following variable shown in Fig. 4.

$$d_{i,t} = \begin{cases} \dfrac{\| \left(\boldsymbol{v}_a - \boldsymbol{v}_{o_i}\right)^{\times} \left(\boldsymbol{p}_{o_i} - \boldsymbol{p}_a\right) \|_2}{\|\boldsymbol{v}_a - \boldsymbol{v}_{o_i}\|_2}, & \boldsymbol{v}_a \neq \boldsymbol{v}_{o_i} \\ \|\boldsymbol{p}_{o_i} - \boldsymbol{p}_a\|_2, & \boldsymbol{v}_a = \boldsymbol{v}_{o_i} \end{cases} \tag{15}$$

where "$\times$" denotes the cross product operation, $d_{i,t}$ represents the closest possible distance between the ASV and the obstacle, if the ASV keeps its current moving direction relative to the obstacle. Note that $d_{i,t}$ is only meaningful, if $\left(\boldsymbol{v}_a - \boldsymbol{v}_{o_i}\right)^T \left(\boldsymbol{p}_{o_i} - \boldsymbol{p}_a\right) > 0$. If $\left(\boldsymbol{v}_a - \boldsymbol{v}_{o_i}\right)^T \left(\boldsymbol{p}_{o_i} - \boldsymbol{p}_a\right) > 0$, it implies that the ASV moves towards the obstacle, otherwise, the ASV moves away from the obstacle. Therefore, the reward function for collision avoidance is defined to be

$$R_{t,2} = \begin{cases} -\sum_{i=1}^{N_o} \dfrac{q_{c,i}\, \mathbb{1}_{o_i}\left(\boldsymbol{x}_{o_i}, \boldsymbol{p}_a, \boldsymbol{v}_a\right)}{1 + \exp\left(c_i \left(d_{i,t} - d_{s_i}\right)\right)}, & d_{ao_i} \leq d_d \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

where $q_{c,i} > 0$ is the maximum possible cost for collisions, $c_i > 0$ is a design parameter, and $\mathbb{1}_{o_i}\left(\boldsymbol{x}_{o_i}, \boldsymbol{p}_a, \boldsymbol{v}_a\right)$ is

$$\mathbb{1}_{o_i}\left(\boldsymbol{x}_{o_i}, \boldsymbol{p}_a, \boldsymbol{v}_a\right) = \begin{cases} 1, & \left(\boldsymbol{v}_a - \boldsymbol{v}_{o_i}\right)^T \left(\boldsymbol{p}_{o_i} - \boldsymbol{p}_a\right) > 0 \\ 0, & \text{otherwise,} \end{cases}$$

The overall reward function is, therefore, defined to be

$$R_t = R_{t,1} + R_{t,2}\ . \tag{17}$$

IV. Model-Reference Deep Reinforcement Learning Design and Implementation

A. Deep Neural Networks

In this paper, the DNNs that approximate both $Q_{\boldsymbol{\pi}}\left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t}\right)$ and $\boldsymbol{\pi}\left(\boldsymbol{u}_{l,t} | \boldsymbol{s}_t\right)$ are chosen to be fully connected multiple layer perceptrons (MLP) with rectified linear unit (ReLU) nonlinearities as the activation functions [45]. The ReLU nonlinearities are defined as $\rho\left(z\right) = \max\left\{z, 0\right\}$. For a vector $z = [z_1, \ldots, z_n]^T \in \mathbb{R}^n$, there exists $\rho\left(z\right) = [\rho\left(z_1\right), \ldots, \rho\left(z_n\right)]^T$. As an example, a MLP with two hidden layers is

$$\underline{MLP}^2_{\mathrm{w}}\left(z\right) = \mathrm{w}_2\left[\rho\left(\mathrm{w}_1\left[\rho\left(\mathrm{w}_0\begin{bmatrix} z \\ 1 \end{bmatrix}\right), 1\right]^T\right)^T, 1\right]^T \tag{18}$$

where $\left[z^T, 1\right]^T$ is a vector composed of z and a bias 1, the superscript "2" denotes the total number of hidden layers, the subscript "w" denotes the parameter set to be trained in a MLP with $\mathrm{w} = \{\mathrm{w}_0, \mathrm{w}_1, \mathrm{w}_2\}$, and w_0, w_1, and w_2 are weight matrices with appropriate dimensions.

If there is a set of inputs $z = \{z_1, \ldots, z_L\}$ for the MLP in (18) with $z_1, \ldots, z_L$ denoting vector signals, we have

$$\underline{MLP}^2_{\mathrm{w}}\left(z\right) = \underline{MLP}^2_{\mathrm{w}}\left(\left[z_1^T, \ldots, z_L^T\right]^T\right). \tag{19}$$

Besides, $\underline{MLP}^2_{\mathrm{w}}\left(z_1, z_2\right) = \underline{MLP}^2_{\mathrm{w}}\left(\left[z_1^T, z_2^T\right]^T\right)$ for two vector inputs z_1 and z_2. If $z_1 = \{z_{11}, \ldots, z_{1L}\}$ is a set of vectors, $\underline{MLP}^2_{\mathrm{w}}\left(z_1, z_2\right) = \underline{MLP}^2_{\mathrm{w}}\left(\left[z_{11}^T, \ldots, z_{1L}^T, z_2^T\right]^T\right)$.

Let $Q_\theta\left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t}\right)$ be the approximated Q-function using a MLP with a set of parameters denoted by θ. Following (18) and (19), the Q-function approximation $Q_\theta\left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t}\right)$ is

$$Q_\theta\left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t}\right) = \underline{MLP}^{K_1}_{\theta}\left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t}\right) \tag{20}$$

where $\theta = \left\{\theta_0, \ldots, \theta_{K_1}\right\}$ with θ_i for $0 \leq i \leq K_1$ denoting the weight matrices with proper dimensions. The deep neural network for Q_θ is illustrated in Fig. 5.

The control law $\boldsymbol{u}_l$ is also approximated using a MLP. The approximated control law of $\boldsymbol{u}_l$ with a parameter set ϕ is

$$\boldsymbol{u}_{l,\phi} = \underline{MLP}^{K_2}_{\phi}\left(\boldsymbol{s}_t\right). \tag{21}$$

The illustration of $\boldsymbol{u}_{l,\phi}$ is given in Fig. 5. In SAC, there are two outputs for the MLP in (21). One is the control law $\boldsymbol{u}_{l,\phi}$, the other one is σ_ϕ that is the standard deviation of the exploration noise [44]. According to (8), the parameterized policy $\boldsymbol{\pi}_\phi$ is

$$\boldsymbol{\pi}_\phi = \mathcal{N}\left(\boldsymbol{u}_{l,\phi}\left(\boldsymbol{s}_t\right), \boldsymbol{\sigma}^2_\phi\right). \tag{22}$$

The deep neural network for Q_θ is called "*critic*", while the one for $\boldsymbol{\pi}_\phi$ is called "*actor*".

B. Algorithm Design and Implementation

The algorithm training process is illustrated in Fig. 6. The whole training process will be offline. We repeatedly run the system (5) under a trajectory tracking task. At each time step $t+1$, we collect data samples, such as an input from the last time step $\boldsymbol{u}_{l,t}$, a state from the last time step $\boldsymbol{s}_t$, a reward R_t, and a current state $\boldsymbol{s}_{t+1}$. Those historical data will be stored as a tuple $\left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t}, R_t, \boldsymbol{s}_{t+1}\right)$ at a replay memory $\mathcal{D}$ [46]. At each policy evaluation or improvement step, we randomly sample a batch of historical data, $\mathcal{B}$, from the replay memory $\mathcal{D}$ for the training of the parameters θ and ϕ. Starting the training, we apply the baseline control policy $\boldsymbol{u}_b$ to an ASV system to collect the initial data $\mathcal{D}_0$ as shown in Algorithm 1. The initial data set $\mathcal{D}_0$ is used for the initial fitting of Q-value functions. When the initialization is over, we execute both $\boldsymbol{u}_b$ and the latest updated RL policy $\boldsymbol{\pi}_\phi\left(\boldsymbol{u}_{l,t} | \boldsymbol{s}_t\right)$ to run the ASV system.

At the policy evaluation step, the parameters θ are trained to minimize the following Bellman residual.

$$J_Q\left(\theta\right) = \mathbb{E}_{\left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t}\right) \sim \mathcal{D}}\left[\frac{1}{2}\left(Q_\theta\left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t}\right) - Y_{target}\right)^2\right] \tag{23}$$

where $\left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t}\right) \sim \mathcal{D}$ implies that we randomly pick data samples $\left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t}\right)$ from a replay memory $\mathcal{D}$. In the final implementation, two critics are introduced to reduce the over-estimation issue in the training of critic neural networks. Under the two-critic mechanism, the target value Y_{target} is

$$Y_{target} = R_t + \gamma \min\left\{Q_{\bar{\theta}_1}, Q_{\bar{\theta}_2}\right\} - \gamma\, \alpha \ln\left(\boldsymbol{\pi}_\phi\right). \tag{24}$$

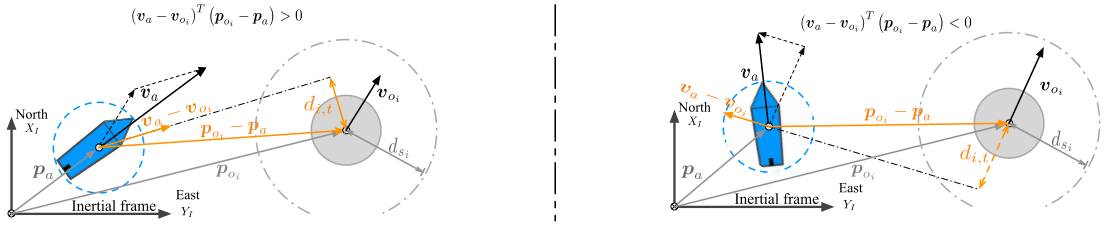


Fig. 4. Illustration of $d_{i,t}$ (Note that $d_{i,t}$ is only useful when $\left(\boldsymbol{v}_a - \boldsymbol{v}_{o_i}\right)^T \left(\boldsymbol{p}_{o_i} - \boldsymbol{p}_a\right) > 0$, otherwise collision is avoided).

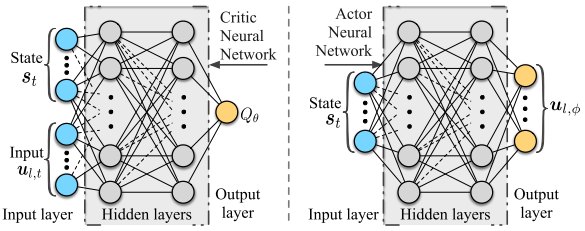


Fig. 5. Approximation of Q_θ and $\boldsymbol{u}_{l,\phi}$ using MLP's.

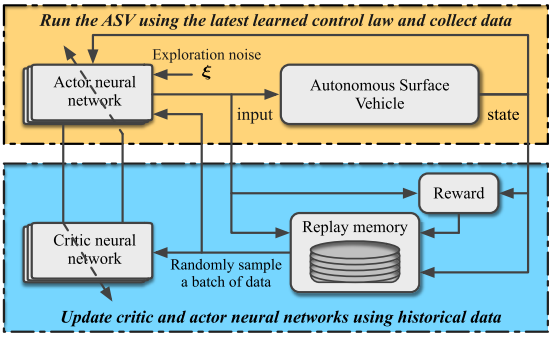


Fig. 6. Offline training process of deep RL.

where $\bar{\theta}_1$ and $\bar{\theta}_2$ are the target parameters updated slowly.

At the policy improvement step, the objective function defined in (13) is represented using data samples from the replay memory $\mathcal{D}$ as given in (25).

$$J_\pi(\phi) = \mathbb{E}_{(\boldsymbol{s}_t, \boldsymbol{u}_{l,t}) \sim \mathcal{D}} \Big(\alpha \ln(\boldsymbol{\pi}_\phi) - Q_\theta\left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t}\right) \Big). \quad (25)$$

Parameter ϕ is trained to minimize (25) using a stochastic gradient descent technique. The temperature parameters α are updated by minimizing

$$J_\alpha = \mathbb{E}_{\boldsymbol{\pi}} \left[-\alpha \ln \boldsymbol{\pi}\left(\boldsymbol{u}_{l,t} | \boldsymbol{s}_t\right) - \alpha \bar{\mathcal{H}} \right] \quad (26)$$

where $\bar{\mathcal{H}}$ is a target entropy. The entire process is summarized in Algorithm 1, in which ι_Q, ι_π, $\iota_\alpha > 0$ are learning rates, and $\kappa > 0$ is a constant scalar.

Once the training process is over, Algorithm 1 will output the optimal parameters for the DNNs in (20) and (21). Hence, the learned control law $\boldsymbol{u}_l$ is approximated by

$$\boldsymbol{u}_l \simeq \boldsymbol{u}_{l,\phi^*} \quad (27)$$

where ϕ^* is the optimal parameter set for the MLP in (21).

V. Performance Analysis

A. Convergence Analysis

As we mentioned in Section III-A, the baseline control $\boldsymbol{u}_b$ is assumed to stabilize the ASV without collision avoidance.

Algorithm 1 Model Reference Reinforcement Learning Control

Initialize parameters θ_1, θ_2, and ϕ for (20) and (21).
Assign values to the target parameters $\bar{\theta}_1 \leftarrow \theta_1$, $\bar{\theta}_2 \leftarrow \theta_2$, $\mathcal{D} \leftarrow \emptyset$, $\mathcal{D}_0 \leftarrow \emptyset$,
Get data set $\mathcal{D}_0$ by running $\boldsymbol{u}_b$ on (5) with $\boldsymbol{u}_l = \boldsymbol{0}$
Turn off the exploration and train initial critic parameters θ_1^0, θ_2^0 using $\mathcal{D}_0$ according to (23).
Initialize the replay memory $\mathcal{D} \leftarrow \mathcal{D}_0$
Assign initial values to critic parameters $\theta_1 \leftarrow \theta_1^0$, $\theta_2 \leftarrow \theta_2^0$ and their targets $\bar{\theta}_1 \leftarrow \theta_1^0$, $\bar{\theta}_2 \leftarrow \theta_2^0$
repeat
  for each data collection step **do**
    Choose an action $\boldsymbol{u}_{l,t}$ according to $\boldsymbol{\pi}_\phi\left(\boldsymbol{u}_{l,t} | \boldsymbol{s}_t\right)$
    Collect $\boldsymbol{s}_{t+1} = \left\{\boldsymbol{x}_{t+1}, \boldsymbol{x}_{m,t+1}, \boldsymbol{u}_{b,t+1}\right\}$
    $\mathcal{D} \leftarrow \mathcal{D} \bigcup \left\{\boldsymbol{s}_t, \boldsymbol{u}_{l,t}, R\left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t}\right), \boldsymbol{s}_{t+1}\right\}$
  end for
  for each gradient update step **do**
    Sample a batch of data $\mathcal{B}$ from $\mathcal{D}$
    $\theta_j \leftarrow \theta_j - \iota_Q \nabla_\theta J_Q\left(\theta_j\right)$, and $j = 1, 2$
    $\phi \leftarrow \phi - \iota_\pi \nabla_\phi J_{\boldsymbol{\pi}}(\phi)$,
    $\alpha \leftarrow \alpha - \iota_\alpha \nabla_\alpha J_\alpha(\alpha)$
    $\bar{\theta}_j \leftarrow \kappa \theta_j + (1-\kappa)\bar{\theta}_j$, and $j = 1, 2$
  end for
until convergence (i.e. $J_Q(\theta) <$ a small threshold)
Output the optimal parameters ϕ^* and θ_j^*, and $j = 1, 2$

Therefore, the following assumption is introduced for the convergence analysis.

Assumption 1: If there are no obstacles, the trajectory tracking errors of the ASV are bounded using the baseline control $\boldsymbol{u}_b$.

According to (14) and (16), both $R_{t,1}$ and $R_{t,2}$ are non-positive. With Assumption 1, the reward function $R_{t,1}$ is ensured to be bounded. Additionally, the reward function $R_{t,2}$ is bounded by design for a finite number of obstacles. Hence, the overall reward R_t is bounded, namely

$$R_t \in [R_{min},\ 0] \quad (28)$$

where R_{min} is the lowest bound for the reward function.

In terms of (28), we can present the following Lemma 1 and Lemma 2 for the convergence analysis of the entropy-regularized SAC algorithm [44], [47].

Lemma 1 (Policy Evaluation): Let $\mathcal{T}^\pi$ be the Bellman backup operator under a fixed policy $\boldsymbol{\pi}$ and $Q^{k+1}(\boldsymbol{s}, \boldsymbol{u}_l) = \mathcal{T}^\pi Q^k(\boldsymbol{s}, \boldsymbol{u}_l)$. The sequence $Q^{k+1}(\boldsymbol{s}, \boldsymbol{u}_l)$ will converge to the soft Q-function $Q^{\boldsymbol{\pi}}$ of the policy $\boldsymbol{\pi}$ as $k \to \infty$.

Proof: Proof details are given in Appendix A. □

Lemma 2 (Policy Improvement): Let $\boldsymbol{\pi}_{old}$ be an old policy and $\boldsymbol{\pi}_{new}$ be a new policy obtained according to (12). There exists $Q^{\boldsymbol{\pi}_{new}}(\boldsymbol{s}, \boldsymbol{u}_l) \geq Q^{\boldsymbol{\pi}_{old}}(\boldsymbol{s}, \boldsymbol{u}_l)$ $\forall \boldsymbol{s} \in \mathcal{S}$ and $\forall \boldsymbol{u} \in \mathcal{U}$.

Proof: Proof details are given in Appendix B. □

In terms of (1) and (2), we are ready to present Theorem 1 to show the convergence of the model-reference RL algorithm. In the sequel, the superscript i denotes the i-th iteration of the algorithm, where $i = 0, 1, \ldots, \infty$.

Theorem 1 (Convergence): Suppose $\boldsymbol{\pi}^i$ is the policy obtained at the i-th policy improvement with $\boldsymbol{\pi}^0$ denoting any initial policy in Π, and $i = 0, 1, \ldots, \infty$. If one repeatedly applies the policy evaluation and improvement steps, there exists $\boldsymbol{\pi}^i \to \boldsymbol{\pi}^*$ as $i \to \infty$ such that $Q^{\boldsymbol{\pi}^*}(\boldsymbol{s}, \boldsymbol{u}_l) \geq Q^{\boldsymbol{\pi}^i}(\boldsymbol{s}, \boldsymbol{u}_l)$ $\forall \boldsymbol{\pi}^i \in \Pi$, $\forall \boldsymbol{s} \in \mathcal{S}$, and $\forall \boldsymbol{u}_l \in \mathcal{U}$, where $\boldsymbol{\pi}^* \in \Pi$ is the optimal policy.

Proof: Proof details are given in Appendix C. □

B. Stability of the Tracking Control

The closed-loop stability is analyzed under the general tracking performance without the consideration of collision avoidance, as the tracking control is the fundamental task. Before the closed-loop stability is analyzed, Definition 1 is introduced, which is similar to the admissible control in adaptive dynamic programming [48].

Definition 1: A control law $\boldsymbol{u}_b$ is said admissible with respect to the system (5), if it can stabilize the system (5) and ensure that the state of (5) is uniformly ultimately bounded under system uncertainties.

Note that the admissible control in [34], [35] needs to provide the asymptotic stability for the system. However, the admissible control in Definition 1 doesn't necessarily ensure the system (5) to be asymptotically stable. Hence, the admissible control in this paper is less conservative than that in [34], [35].

Assume that the baseline control $\boldsymbol{u}_b$ developed using the nominal system (6) of the ASV (5) is an admissible control law for the uncertain system (5). Let $\Delta(t)$ be the overall uncertainties in (5). Without loss of generality, $\Delta(t)$ is assumed to be bounded, namely $\|\Delta(t)\|_{\mathcal{L}_\infty} \leq \bar{\Delta}$ where $\|\cdot\|_{\mathcal{L}_\infty}$ is the $\mathcal{L}_\infty$ norm. In this paper, the objective of the tracking control in obstacle-free environment is to ensure that an ASV (5) can track its desired behaviour defined by its nominal system (6), namely $\|\boldsymbol{x} - \boldsymbol{x}_m\|_2 \to 0$ as $t \to \infty$. Let $\boldsymbol{e}_t = \boldsymbol{x}_t - \boldsymbol{x}_{m,t}$ be the tracking error at the time instant t. The following assumption is made for a admissible baseline control $\boldsymbol{u}_b$ according to Definition 1 and Theorem 4.18 in [49] (Chapter 4, Page 172).

Assumption 2: The baseline control law $\boldsymbol{u}_b$ is admissible with respect to (5), and there exists a continuously differentiable function $\mathbb{V}(\boldsymbol{s}_t)$ associate with $\boldsymbol{u}_b$ such that

$$\begin{aligned} \mu_1(\|\boldsymbol{e}_t\|_2) \leq \mathbb{V}(\boldsymbol{e}_t) \leq \mu_2(\|\boldsymbol{e}_t\|_2) \\ \mathbb{V}(\boldsymbol{e}_{t+1}) - \mathbb{V}(\boldsymbol{e}_t) \leq -\mathbb{W}_1(\boldsymbol{e}_t) + \mu_3(\|\Delta(\boldsymbol{e}_t)\|_2) \\ \mathbb{W}_1(\boldsymbol{e}_t) > \mu_3(\|\Delta(t)\|_2), \ \forall \|\boldsymbol{e}_t\|_2 > c_\Delta \end{aligned} \quad (29)$$

where $\mu_1(\cdot)$, $\mu_2(\cdot)$, and $\mu_3(\cdot)$ are class $\mathcal{K}$ functions, $\mathbb{W}(\boldsymbol{e}_t)$ is a continuous positive definite function, and c_Δ is a constant related to the upper bound of system uncertainty

Assumption 2 is possible in real world. One can treat the nominal model (6) as a linearized model of the overall ASV system (5) around a certain equilibrium. Assumption 2 presents the basic design requirements for the baseline control law. With a baseline control law satisfying Assumption 1, we could obtain two advantages which makes the RL process more efficient. Firstly, it can ensure that the reward function R_t is bounded, implying that both $V_{\boldsymbol{\pi}}(\boldsymbol{s}_{t+1})$ and $Q(\boldsymbol{s}_t, \boldsymbol{u}_{l,t})$ are bounded. Secondly, it could provide a "warm" start for the RL process. In the stability analysis, we ignore the entropy term $\mathcal{H}(\boldsymbol{\pi})$, as it will converge to zero in the end and it is only introduced to regulate the exploration magnitude. Now, Theorem 2 is presented to demonstrate the closed-loop stability of the ASV system (5) under the composite control law (7).

Theorem 2 (Stability of Tracking Control): Suppose Assumption 2 holds. The overall control law $\boldsymbol{u}^i = \boldsymbol{u}_b + \boldsymbol{u}_l^i$ can always stabilize the ASV system (5), where $\boldsymbol{u}_l^i$ represents the RL control law from i-th iteration, and $i = 0, 1, 2, \ldots\infty$.

Proof: The details of proof can be found in Appendix D. □

Remark 3: The proposed algorithm can also obtain good results in the presence of environmental disturbances such as waves and sea winds. Disturbances due to waves and sea winds can be considered in the unknown term $\boldsymbol{G}(\boldsymbol{\nu}) = \left[\boldsymbol{g}_1(\boldsymbol{\nu}), \boldsymbol{g}_2(\boldsymbol{\nu}), \boldsymbol{g}_3(\boldsymbol{\nu})\right]^T$. To achieve good results in the face of disturbances, one only need to incorporate simulated disturbances by waves and sea winds at the training stage. Another way is to deploy ASVs in environments with waves or winds at the training stage. In this case, the proposed model reference reinforcement learning can learn to counteract the impact by environmental disturbances.

VI. Simulation Results

In this section, the proposed learning-based control algorithm is implemented to the trajectory tracking control of a supply ship model presented in [50], [51]. The ASV has two actuators that are a propeller and a rudder in the rear. Hence, we mainly consider two control inputs in the design, which are τ_u for the surge speed control and τ_r for the heading control, respectively. By default, the sway speed is not controlled, which implies $\tau_v = 0$ in the simulations. Model parameters are summarized in Table II in Appendix E. The unmodeled dynamic terms $\boldsymbol{g}_1(\boldsymbol{\nu})$, $\boldsymbol{g}_2(\boldsymbol{\nu})$, and $\boldsymbol{g}_3(\boldsymbol{\nu})$ are gravitational, buoyancy, and environmental forces and moments [11]. They depend on the shape, volume displacement, and motion of an ASV, and tends to affect the trajectory tracking performance of an ASV, e.g., course keeping and velocity holding. In the simulations, we choose $g_1 = 0.279uv^2 + 0.342\ v^2r$, $g_2 = 0.912u^2v$, and $g_3 = 0.156ur^2 + 0.278urv^3$, respectively [50], [51].

The based-line control law $\boldsymbol{u}_b$ is designed based on a nominal model with the following simplified linear dynamics in terms of the backstepping control method [31], [49].

$$\boldsymbol{M}_m \dot{\boldsymbol{\nu}}_m = \boldsymbol{\tau} - \boldsymbol{D}_m \boldsymbol{\nu}_m \quad (30)$$

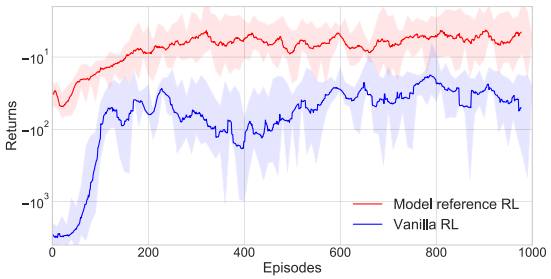


Fig. 7. Learning curves of two RL algorithms at training (One episode is a training trial, and 1000 time steps per episode).

where $\boldsymbol{D}_m = diag\{-X_v, -Y_v, -N_r\}$, and $\boldsymbol{M}_m = diag\{M_{11}, M_{22}, M_{33}\}$. Hence, $A_m = -\boldsymbol{M}_m^{-1}\boldsymbol{D}_m$ and $B_m = -\boldsymbol{M}_m^{-1}$ in (6).

In the simulation, a motion planner is employed to generate the reference trajectories. The motion planner is expressed as

$$\dot{\boldsymbol{\eta}}_r = \boldsymbol{R}\left(\boldsymbol{\eta}_r\right)\boldsymbol{\nu}_r, \quad \dot{\boldsymbol{\nu}}_r = \boldsymbol{a}_r \tag{31}$$

where $\boldsymbol{\eta}_r = [x_r, y_r, \psi_r]^T$, $\boldsymbol{\nu}_r = [u_r, 0, r_r]^T$, and $\boldsymbol{a}_r = [\dot{u}_r, 0, \dot{r}_r]^T$.

A. Trajectory Tracking Control Without Obstacles

In the first simulation, the initial position vector $\boldsymbol{\eta}_r(0)$ is chosen to be $\boldsymbol{\eta}_r(0) = \left[0, 0, \frac{\pi}{4}\right]^T$, and we set $u_r(0) = 0.4\ m/s$ and $r_r(0) = 0\ rad/s$. The reference acceleration $\dot{u}_r$ and angular rates are chosen to be

$$\dot{u}_r = \begin{cases} 0.005 \ \ m/s^2 & \text{if } t < 20\ s \\ 0 \ \ m/s^2 & \text{otherwise,} \end{cases} \tag{32}$$

$$\dot{r}_r = \begin{cases} \dfrac{\pi}{600} \ \ rad/s^2 & \text{if } 25\ s \le t < 50\ s \\ 0 \ \ rad/s^2 & \text{otherwise.} \end{cases} \tag{33}$$

The reference signals $\boldsymbol{\eta}_r$ and $\boldsymbol{\nu}_r$ are calculated using the reference motion planner (31) based on the aforementioned initial conditions and the reference acceleration and angular rates given in (32) and (33), respectively.

At the training stage, we uniformly randomly sample $x(0)$ and $y(0)$ from $(-1.5, 1.5)$, $\psi(0)$ from $(0.1\pi, 0.4\pi)$ and $u(0)$ from $(0.2, 0.4)$, and we choose $v(0) = 0$ and $r(0) = 0$. The proposed control algorithm is compared to two benchmark designs: the baseline control $\boldsymbol{u}_0$ and the RL control without $\boldsymbol{u}_0$. Configurations for the training and neural networks are found in Table III in Appendix E. The matrices $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ are chosen to be $\boldsymbol{H}_1 = diag\{0.025, 0.025, 0.0016, 0.005, 0.001, 0\}$ and $\boldsymbol{H}_2 = diag\left\{1.25e^{-3}, 1.25e^{-3}\right\}$, respectively. During the training process, we repeat the training processes for 1000 times (i.e., 1000 episodes). For each episode, the ASV system is run for 100 s. Fig. 7 shows the learning curves of the proposed algorithm (red) and the RL algorithm without baseline control (blue). The learning curves demonstrate that both of the two algorithms will converge in terms of the long term returns. However, our proposed algorithm results in a larger return (red) in comparison to the RL without baseline control (blue). Hence, the introduction of the baseline control helps to increase the sample efficiency significantly, as the proposed algorithm (blue) converges faster to a higher return value.

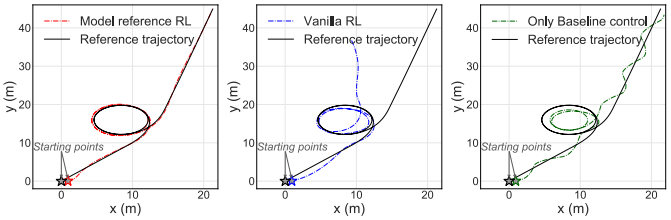


Fig. 8. Trajectory tracking results of the three algorithms.

At the evaluation stage, we run the ASV system for 200 s and change the reference trajectory to demonstrate whether the control law can ensure stable trajectory tracking. At the second evaluation, the reference angular acceleration is changed to

$$\dot{r}_r = \begin{cases} \dfrac{\pi}{600} \ \ rad/s^2 & \text{if } \ 25\ s \le t < 50\ s \\ -\dfrac{\pi}{600} \ \ rad/s^2 & \text{if } 125\ s \le t < 150\ s \\ 0 \ \ rad/s^2 & \text{otherwise.} \end{cases} \tag{34}$$

The tracking performance of the three algorithms (our proposed algorithm, the baseline control $\boldsymbol{u}_0$, and only RL control) is shown in Fig. 8. As observed in Fig. 8, an ASV using the control law learned using vanilla deep RL tends to drift away from the designed trajectory. Thus, vanilla deep RL could not ensure the closed-loop stability. In addition, the baseline control itself fails to achieve acceptable tracking performance due to the existence of system uncertainties. By combining the baseline control and deep RL, the tracking performance is improved dramatically, and the closed-loop stability is guaranteed. The tracking errors in the X- and Y-coordinates of the inertial frame are summarized in Figs. 9.(a) and 9.(b), respectively. The introduction of deep RL increases the tracking performance of the baseline control law substantially. The control inputs are shown in Figs. 9.(c) and 9.(d).

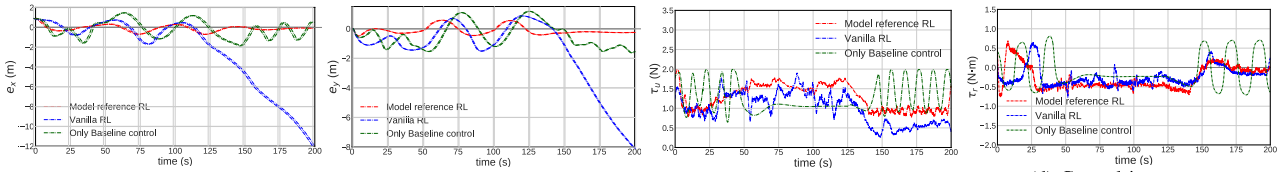
B. Tracking Control With Fixed Obstacles

In the second simulation, the initial position vector $\boldsymbol{\eta}_r(0)$ is chosen to be the same as the case in Section VI-A. We set $u_r(0) = 0.7\ m/s$ and $r_r(0) = 0\ rad/s$. The reference acceleration is set as $\dot{u}_r = 0\ m^2/s$. The angular rate is

$$\dot{r}_r = \begin{cases} \dfrac{\pi}{800} \ \ rad/s^2 & \text{if } 20\ s \le t < 50\ s \\ 0 \ \ rad/s^2 & \text{otherwise.} \end{cases} \tag{35}$$

Initial states of the ASV are randomly generated as summarized in Section VI-A. Three fixed obstacles are added to the simulation environment as shown in Fig. 10, which have a radius of 1.5 m, 1.8 m, and 2.0 m, respectively (from the lower to the upper). The detection radius for the ASV is $d_d = 7.5\ m$, and the radius of the ASV is $d_a = 1\ m$. The DNN configurations and training set-up for the collision avoidance scenario is the same as shown in Table III in Appendix E. In the simulation, $q_{c,i} = 1$ and $c_i = 25$ for all obstacles.

At the training stage, 1000 episodes of training are conducted. For each episode, the ASV system is run for 100 s.



(a) Tracking errors, e_x (b) Tracking errors, e_y (c) Control inputs, τ_u (d) Control inputs, τ_r

Fig. 9. Tracking errors and control inputs (without obstacles).

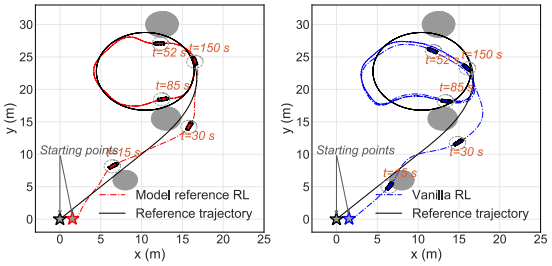


Fig. 10. Trajectory tracking with fixed obstacles.

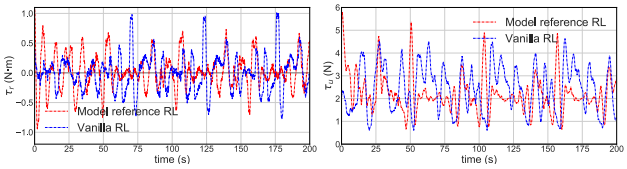


Fig. 11. Control inputs (with fixed obstacles).

At the evaluation stage, we run the ASV system for 200 s to demonstrate whether the control law can ensure stable trajectory tracking and collision avoidance. The proposed algorithm is compared with the RL algorithm without baseline control. The simulation results of both our algorithm and the RL without baseline control are shown in Fig. 10. The learned control law by the vanilla RL fails to avoid collision with some obstacle as demonstrated in Fig. 10. However, our algorithm can ensure both the trajectory tracking and the collision avoidance at the same time. The control inputs are shown in Fig. 11.

C. Tracking Control With Fixed Obstacles and Moving Obstacles

In the third simulation, we show the collision avoidance with moving obstacles. The reference trajectory is the same as that in the second simulation in Section VI-B. In the simulation, there are two fixed obstacles with radii of 1.5 and 2.0 m from the lower to the upper and one moving obstacle (e.g., another ASV). The moving obstacle has a safe radius of 1 m, and moving with a constant speed with $\boldsymbol{o}_{i,v} = [-0.4,\ 0.25]$ m/s in the simulation. The training setup is the same as the case in Section VI-B. At the evaluation, the ASV system is run for 200 s. The trajectory tracking performance of both our algorithm and the RL without baseline control is shown in Fig. 12. Although both of the two algorithms can learn a control law with collision avoidance, our algorithm apparently has better tracking performance than the vanilla RL.

D. The Impact of Different Choices of c

Three choices are considered for c, which are $c = 0.25$, $c = 2.5$, and $c = 25$, respectively. The trajectory tracking performance is illustrated in Fig. 13. A small c will make $R_{2,t}$ change slowly with respect to the distance between the ASV and an obstacle, thereby making the ASV take more conservative actions to avoid collisions as shown in Fig. 13.

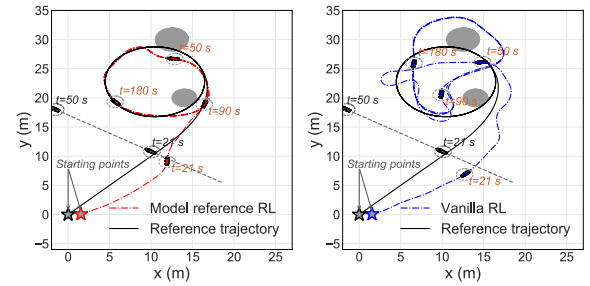


Fig. 12. Trajectory tacking with fixed and moving obstacles.

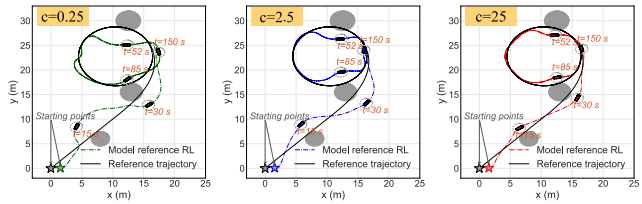


Fig. 13. Collision avoidance performance at different c's.

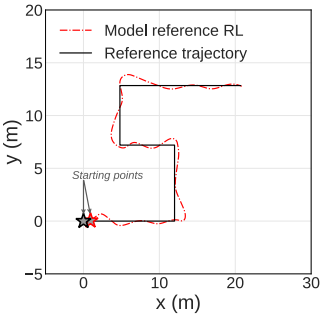


Fig. 14. Tracking a trajectory with right angles.

E. Tracking of a Non-Smooth Trajectory

In the last simulation, the proposed algorithm is applied to track a non-smooth trajectory to further demonstrate its potential. The trajectory tracking result is shown in Fig. 14. It is obvious that the proposed method has the potential to track a polyline with right angles or other complex curves.

VII. Conclusion

In this paper, we presented a novel learning-based control algorithm for ASV systems with collision avoidance. The proposed control algorithm combined a conventional control method with deep reinforcement learning to provide

closed-loop stability guarantee, uncertainty compensation, and collision avoidance. Convergence of the learning algorithm was analyzed. We also presented the stability analysis of the tracking control. The proposed control algorithm shows much better performance in both tracking control and collision avoidance than the RL without baseline control. In the future works, we will further analyze the sample efficiency of the proposed algorithm, and extend the design to the case with environmental disturbances.

APPENDIX

A. Proof of Lemma 1

Proof: Introduce an entropy-augmented reward function $\hat{R}_t$.

$$\hat{R}_t = R_t - \gamma\, \mathbb{E}_{\boldsymbol{s}_{t+1}} \left\{ \mathbb{E}_{\boldsymbol{\pi}} \left[\alpha \ln \left(\boldsymbol{\pi} \left(\boldsymbol{u}_{l,t+1} | \boldsymbol{s}_{t+1} \right) \right) \right] \right\} \quad \text{(A.1)}$$

Hence, the Bellman backup operation can be rewritten as

$$\mathcal{T}^{\boldsymbol{\pi}} Q_{\boldsymbol{\pi}} \left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t} \right) = \hat{R}_t + \gamma\, \mathbb{E}_{\boldsymbol{s}_{t+1}, \boldsymbol{\pi}} \left[Q_{\boldsymbol{\pi}} \left(\boldsymbol{s}_{t+1}, \boldsymbol{u}_{l,t+1} \right) \right] \quad \text{(A.2)}$$

For $\alpha < \infty$, the second term in (A.1) is bounded. According to (28), one has $\hat{R}_{min}$ and $\hat{R}_{max}$ such that $\hat{R}_t \in \left[\hat{R}_{min}, \hat{R}_{max} \right]$, and furthermore, $|\hat{R}_t| \leq \bar{R}$ with $\bar{R} = \max \left\{ |\hat{R}_{min}|, |\hat{R}_{max}| \right\}$. In terms of (9), $Q_{\boldsymbol{\pi}} \left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t} \right) = \hat{R}_t + \gamma \sum_{t+1}^{\infty} \sum_{\boldsymbol{u}_{l,t+1}} \boldsymbol{\pi} \left(\boldsymbol{u}_{l,t+1} | \boldsymbol{s}_{t+1} \right) \sum_{\boldsymbol{s}_{t+1}} \mathcal{P}_{t+1|t} \hat{R}_{t+1}$, thus

$$\| Q_{\boldsymbol{\pi}} \left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t} \right) \|_\infty \leq \bar{R} \big/ \left(1 - \gamma \right) \quad \text{(A.3)}$$

where $\| Q_{\boldsymbol{\pi}} \left(\boldsymbol{s}, \boldsymbol{u}_l \right) \|_\infty = \max_{s, u_l} | Q_{\boldsymbol{\pi}} \left(\boldsymbol{s}, \boldsymbol{u}_l \right) |$. Hence, the Q-value $Q_{\boldsymbol{\pi}}$ is bounded in ∞-norm based on the baseline control. For two distinct $Q_{\boldsymbol{\pi}}$ and $Q'_{\boldsymbol{\pi}}$, there exists

$$\begin{aligned} &\| \mathcal{T}^{\boldsymbol{\pi}} Q_{\boldsymbol{\pi}} - \mathcal{T}^{\boldsymbol{\pi}} Q'_{\boldsymbol{\pi}} \|_\infty \\ &= \| \hat{R}_t + \gamma\, \mathbb{E}_{\boldsymbol{s}_{t+1}, \boldsymbol{\pi}} \left[Q_{\boldsymbol{\pi}} \left(\boldsymbol{s}_{t+1}, \boldsymbol{u}_{l,t+1} \right) \right] \\ &\quad - \hat{R}_t - \gamma\, \mathbb{E}_{\boldsymbol{s}_{t+1}, \boldsymbol{\pi}} \left[Q'_{\boldsymbol{\pi}} \left(\boldsymbol{s}_{t+1}, \boldsymbol{u}_{l,t+1} \right) \right] \|_\infty \leq \gamma \| Q_{\boldsymbol{\pi}} - Q'_{\boldsymbol{\pi}} \|_\infty \end{aligned}$$

where $Q'_{\boldsymbol{\pi}}$ and $Q_{\boldsymbol{\pi}}$ are the Q values approximated at the last and current iterations, respectively. The Bellman backup operation (A.2) is γ-contraction with $0 \leq \gamma < 1$. According to Banach's fixed-point theorem, $\mathcal{T}^{\boldsymbol{\pi}}$ possesses a unique fixed point. Hence, $\lim_{k \to \infty} Q^{k+1} \left(\boldsymbol{s}, \boldsymbol{u}_l \right) \to Q^{\boldsymbol{\pi}}$. □

B. Proof of Lemma 2

Proof: Based on (12), we can obtain

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{\pi}_{new}} \Big[\alpha \ln \left(\boldsymbol{\pi}_{new} \left(\boldsymbol{u}_{l,t} | \boldsymbol{s}_t \right) \right) - Q^{\boldsymbol{\pi}_{old}} \left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t} \right) \Big] \\ &\quad \geq \mathbb{E}_{\boldsymbol{\pi}_{old}} \Big[\alpha \ln \left(\boldsymbol{\pi}_{old} \left(\boldsymbol{u}_{l,t} | \boldsymbol{s}_t \right) \right) - Q^{\boldsymbol{\pi}_{old}} \left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t} \right) \Big] \quad \text{(A.4)} \end{aligned}$$

According to (A.4) and (11), it yields

$$\begin{aligned} Q^{\boldsymbol{\pi}_{old}} \left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t} \right) \leq\ & R_t + \gamma\, \mathbb{E}_{\boldsymbol{s}_{t+1}} \Big[\mathbb{E}_{\boldsymbol{\pi}_{new}} \Big[R_{t+1} \\ & + \gamma\, \mathbb{E}_{\boldsymbol{s}_{t+2}} \left[\mathbb{E}_{\boldsymbol{\pi}_{new}} \left[Q^{\boldsymbol{\pi}_{old}} \left(\boldsymbol{s}_{t+2}, \boldsymbol{u}_{l,t+2} \right) \right. \right. \\ & \left. \left. - \alpha \ln \left(\boldsymbol{\pi}_{new} \left(\boldsymbol{u}_{l,t+2} | \boldsymbol{s}_{t+2} \right) \right) \right] \right] \\ & - \alpha \ln \left(\boldsymbol{\pi}_{new} \left(\boldsymbol{u}_{l,t+1} | \boldsymbol{s}_{t+1} \right) \right) \Big] \Big] \\ & \vdots \\ \leq\ & Q^{\boldsymbol{\pi}_{new}} \left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t} \right) \quad \text{(A.5)} \end{aligned}$$

□

C. Proof of Theorem 1

Proof: Lemma 2 implies that $Q^{\boldsymbol{\pi}^i} \left(\boldsymbol{s}, \boldsymbol{u}_l \right) \geq Q^{\boldsymbol{\pi}^{i-1}} \left(\boldsymbol{s}, \boldsymbol{u}_l \right)$, so $Q^{\boldsymbol{\pi}^i} \left(\boldsymbol{s}, \boldsymbol{u}_l \right)$ is monotonically non-decreasing with respect to the policy iteration step i. In addition, $Q^{\boldsymbol{\pi}^i} \left(\boldsymbol{s}, \boldsymbol{u}_l \right)$ is upper bounded according to (17), so $Q^{\boldsymbol{\pi}^i} \left(\boldsymbol{s}, \boldsymbol{u}_l \right)$ will converge to an upper limit $Q^{\boldsymbol{\pi}^*} \left(\boldsymbol{s}, \boldsymbol{u}_l \right)$ with $Q^{\boldsymbol{\pi}^*} \left(\boldsymbol{s}, \boldsymbol{u}_l \right) \geq Q^{\boldsymbol{\pi}^i} \left(\boldsymbol{s}, \boldsymbol{u}_l \right)$ $\forall \boldsymbol{\pi}_i \in \Pi$, $\forall \boldsymbol{s} \in \mathcal{S}$, and $\forall \boldsymbol{u}_l \in \mathcal{U}$. □

D. Proof of Theorem 2

Proof: In our proposed algorithm, we start the training/learning using the baseline control law $\boldsymbol{u}_b$. According to Lemma 1, we are able to obtain the corresponding Q value function for the baseline control law $\boldsymbol{u}_b$. Let the Q value function be $Q^0 \left(\boldsymbol{s}, \boldsymbol{u}_l^0 \right)$ at the beginning of the iteration where $\boldsymbol{u}_l^0$ is the initial RL-based control function. According to the definitions of the reward function in (17) and Q value function in (9), we can choose the Lyapunov function candidate as

$$\mathbb{V}^0 \left(\boldsymbol{e} \right) = -Q^0 \left(\boldsymbol{s}, \boldsymbol{u}_l^0 \right) \quad \text{(A.6)}$$

where $Q^0 \left(\boldsymbol{s}, \boldsymbol{u}_l^0 \right)$ is the action value function of the initial control law $\boldsymbol{u}_l^0$. Note that the baseline control $\boldsymbol{u}_b$ is implicitly included in the state vector $\boldsymbol{s}$, as $\boldsymbol{s}$ consists of $\boldsymbol{x}$, $\boldsymbol{x}_m$, and $\boldsymbol{u}_b$ in this paper as discussed in Section III. Hence, $\mathbb{V} \left(\boldsymbol{s}_t \right)$ in Assumption 2 is a Lyapunov function for the closed-loop system of (5) with the baseline control $\boldsymbol{u}_b$.

Since ASVs have deterministic dynamics and exploration noises are not considered, we have $Q^0 \left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t} \right) = V^0 \left(\boldsymbol{s}_t \right)$ and $Q^0 \left(\boldsymbol{s}_t, \boldsymbol{u}_{l,t} \right) = R_t^0 + \gamma\, Q^0 \left(\boldsymbol{s}_{t+1}, \boldsymbol{u}_{l,t+1} \right)$ where $R_t^0 = R(\boldsymbol{s}_t, \boldsymbol{u}_{l,t}^0)$. With the consideration of $\mathbb{V}^0 \left(\boldsymbol{e} \right) = -Q^0 \left(\boldsymbol{s}, \boldsymbol{u}_l \right)$, there exists $\mathbb{V}^0 \left(\boldsymbol{e}_t \right) = -R_t^0 + \gamma\, \mathbb{V}^0 \left(\boldsymbol{e}_{t+1} \right)$.

If Assumption 2 holds, there exists $\mathbb{V}^0 \left(\boldsymbol{e}_{t+1} \right) - \mathbb{V}^0 \left(\boldsymbol{e}_t \right) \leq -\mathbb{W} \left(\boldsymbol{e}_t \right) + \mu_3 \left(\| \Delta \left(\boldsymbol{s}_t \right) \|_2 \right)$ and $\mathbb{W} \left(\boldsymbol{e}_t \right) > \mu_3 \left(\| \Delta \left(t \right) \|_2 \right)$, $\forall \| \boldsymbol{e}_t \|_2 > c_\Delta$. Hence,

$$\left(1 - \gamma \right) \mathbb{V}^0 \left(\boldsymbol{e}_{t+1} \right) + R_t^0 \leq -\mathbb{W} \left(\boldsymbol{e}_t \right) + \mu_3 \left(\| \Delta \left(t \right) \|_2 \right) \quad \text{(A.7)}$$

In the policy improvement, the control law is updated by

$$\boldsymbol{u}_l^1 = arg \min_{\boldsymbol{u}_l} \left(-R_t^0 + \gamma\, \mathbb{V}^0 \left(\boldsymbol{e}_{t+1} \right) \right) \quad \text{(A.8)}$$

Note that $\boldsymbol{u}_l$ is implicitly contained in both R_t and $\mathbb{V}^0 \left(\boldsymbol{e}_{t+1} \right)$ according to (14) and (A.6). In the policy evaluation, the following Bellman backup operation is repeatedly conducted.

$$\mathbb{V}^1 \left(\boldsymbol{e}_t \right) = -R_t^1 + \gamma\, \mathbb{V}^0 \left(\boldsymbol{e}_{t+1} \right) \quad \text{(A.9)}$$

where $R_t^1 = R(\boldsymbol{s}_t, \boldsymbol{u}_{l,t}^1)$. The convergence of the Bellman backup operation is provided in Lemma 1. Once the update (A.9) converges, we have

$$\mathbb{V}^1 \left(\boldsymbol{e}_t \right) = -R_t^1 + \gamma\, \mathbb{V}^1 \left(\boldsymbol{e}_{t+1} \right) \quad \text{(A.10)}$$

According to (A.8), it is easy to conclude that $\mathbb{V}^1 \left(\boldsymbol{e}_t \right) \leq \mathbb{V}^0 \left(\boldsymbol{e}_t \right)$, $\forall\ \boldsymbol{e}_t$. In terms of (A.7), one has

$$\left(1 - \gamma \right) \mathbb{V}^1 \left(\boldsymbol{e}_{t+1} \right) \leq -\mathbb{W} \left(\boldsymbol{e}_t \right) + \mu_3 \left(\| \Delta \left(t \right) \|_2 \right) - R_t^0 \quad \text{(A.11)}$$

Hence, for $\boldsymbol{u}_l^1$, there exists

$$\begin{aligned} \mathbb{V}^1 \left(\boldsymbol{e}_{t+1} \right) - \mathbb{V}^1 \left(\boldsymbol{e}_t \right) &= \mathbb{V}^1 \left(\boldsymbol{e}_{t+1} \right) + R_t^1 - \gamma\, \mathbb{V}^1 \left(\boldsymbol{e}_{t+1} \right) \\ &\leq -\mathbb{W} \left(\boldsymbol{e}_t \right) + \mu_3 \left(\| \Delta \left(t \right) \|_2 \right) - R_t^0 + R_t^1 \end{aligned}$$

TABLE II
ASV MODEL PARAMETERS

Parameters	Values	Parameters	Values
m	23.8	$Y_{\dot{r}}$	-0.0
I_z	1.76	Y_r	0.1079
x_g	0.046	$Y_{\|v\|r}$	-0.845
$X_{\dot{u}}$	-2.0	$Y_{\|r\|r}$	-3.45
X_u	-0.7225	N_v	-0.1052
$X_{\|u\|u}$	-1.3274	$N_{\|v\|v}$	5.0437
X_{uuu}	-1.8664	$N_{\|r\|v}$	-0.13
$Y_{\dot{v}}$	-10.0	$N_{\dot{r}}$	-1.0
Y_v	-38.612	N_r	-1.9
$Y_{\|v\|v}$	-36.2823	$N_{\|v\|r}$	0.08
$Y_{\|r\|v}$	-0.805	$N_{\|r\|r}$	-0.75

TABLE III
REINFORCEMENT LEARNING CONFIGURATIONS

Parameters	Values
Learning rate ι_Q	0.001
Learning rate ι_π	0.0001
Learning rate ι_α	0.0001
κ	0.01
Actor neural network	Fully connected MLP with 2 hidden layers (128 neurons per hidden layer)
Critic neural networks	Fully connected MLP with 2 hidden layers (128 neurons per hidden layer)
Replay memory capacity	1×10^6
Sample batch size $\|\mathcal{B}\|$	128
γ	0.998
Training episodes	1000
Steps per episode	1000
Time step size δt	0.1

According to (A.8) and the property of optimality, one has $R_t^1 \geq R_t^0$. In terms of (28), both R_t^1 and R_t^0 are bounded, so $R_t^1 - R_t^0 \in [R_{min}, -R_{min}]$, where R_{min} is a negative constant. As $\mathbb{W}(\boldsymbol{e}_t) > \mu_3(\|\Delta(t)\|_2)$, $\forall \|\boldsymbol{e}_t\|_2 > c_\Delta$, there must exist a new constant $c_\Delta^1 > 0$ such that $\mathbb{W}(\boldsymbol{e}_t) > \mu_3(\|\Delta(t)\|_2) - R_t^0 + R_t^1$, $\forall \|\boldsymbol{e}_t\|_2 > c_\Delta^1$. Hence, the new control law $\boldsymbol{u}_l^1$ can also ensure the closed-loop ASV system to be uniformally ultimately bounded. In the worst case, $\mathbb{V}^1(\boldsymbol{e}_t) = \mathbb{V}^0(\boldsymbol{e}_t)$, which implies that $\boldsymbol{u}_l^1$ will have the same control performance with $\boldsymbol{u}_t^0$, namely guaranteeing the same ultimate boundaries for the tracking errors. If there exists $\mathbb{V}^1(\boldsymbol{e}_t) < \mathbb{V}^0(\boldsymbol{e}_t)$, it implies that $\boldsymbol{u}_l^1$ will result in smaller tracking errors than $\boldsymbol{u}_l^0$.

Following the same analysis, we can show that $\boldsymbol{u}_l^2$ also stabilizes the ASV system (5) in terms of $\mathbb{V}^1(\boldsymbol{s}_t)$ and replacing $\boldsymbol{u}_l^0$ in (A.8) and (A.9) with $\boldsymbol{u}_l^1$. Repeating (A.8) and (A.9) for all $i = 1, 2, \ldots$, we can prove that all $\boldsymbol{u}_l^i$ can stabilize the ASV system (5), if Assumption 2 holds. Thus, the ASV system (5) will be stabilized by the overall control law $\boldsymbol{u}^i = \boldsymbol{u}_b + \boldsymbol{u}_l^i$. □

E. Simulation Configurations

Tables II and III.

REFERENCES

[1] D. O. B. Jones, A. R. Gates, V. A. I. Huvenne, A. B. Phillips, and B. J. Bett, "Autonomous marine environmental monitoring: Application in decommissioned oil fields," *Sci. Total Environ.*, vol. 668, pp. 835–853, Jun. 2019.

[2] J. Majohr and T. Buch, *Advances in Unmanned Marine Vehicles*. London, U.K.: Institution of Engineering and Technology, 2006.

[3] E. Tu, G. Zhang, L. Rachmawati, E. Rajabally, and G.-B. Huang, "Exploiting AIS data for intelligent maritime navigation: A comprehensive survey from data to methodology," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1559–1582, May 2018.

[4] D. D. Bloisi, F. Previtali, A. Pennisi, D. Nardi, and M. Fiorini, "Enhancing automatic maritime surveillance systems with visual information," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 4, pp. 824–833, Apr. 2017.

[5] S. Campbell, W. Naeem, and G. W. Irwin, "A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres," *Annu. Rev. Control*, vol. 36, no. 2, pp. 267–283, Dec. 2012.

[6] K. D. Do and J. Pan, "Global robust adaptive path following of underactuated ships," *Automatica*, vol. 42, no. 10, pp. 1713–1722, Oct. 2006.

[7] A. R. J. Ruiz and F. S. Granja, "A short-range ship navigation system based on Ladar imaging and target tracking for improved safety and efficiency," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 186–197, Mar. 2009.

[8] L. P. Perera, P. Oliveira, and C. G. Soares, "Maritime traffic monitoring based on vessel detection, tracking, state estimation, and trajectory prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1188–1200, Sep. 2012.

[9] T. A. Johansen, T. Perez, and A. Cristofaro, "Ship collision avoidance and COLREGS compliance using simulation-based control behavior selection with predictive hazard assessment," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3407–3422, Dec. 2016.

[10] S. Li, J. Liu, and R. R. Negenborn, "Distributed coordination for collision avoidance of multiple ships considering ship maneuverability," *Ocean Eng.*, vol. 181, pp. 212–226, Jun. 2019.

[11] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Hoboken, NJ, USA: Wiley, 2011.

[12] F. S. Hover *et al.*, "Advanced perception, navigation and planning for autonomous in-water ship hull inspection," *Int. J. Robot. Res.*, vol. 31, no. 12, pp. 1445–1464, Oct. 2012.

[13] D. Gonzalez, J. Perez, V. Milanes, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.

[14] L. Hu *et al.*, "A multiobjective optimization approach for COLREGs-compliant path planning of autonomous surface vehicles verified on networked bridge simulators," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1167–1179, Mar. 2020.

[15] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 3, pp. 972–983, May 2021.

[16] M. Greytak and F. Hover, "Motion planning with an analytic risk cost for holonomic vehicles," in *Proc. 48h IEEE Conf. Decis. Control (CDC)*, Shanghai, China, Dec. 2009, pp. 5655–5660.

[17] D. Panagou, "A distributed feedback motion planning protocol for multiple unicycle agents of different classes," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1178–1193, Mar. 2017.

[18] E. Meyer, H. Robinson, A. Rasheed, and O. San, "Taming an autonomous surface vehicle for path following and collision avoidance using deep reinforcement learning," 2019, *arXiv:1912.08578*. [Online]. Available: https://arxiv.org/abs/1912.08578

[19] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introductions*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[20] R. Yu, Q. Zhu, G. Xia, and Z. Liu, "Sliding mode tracking control of an underactuated surface vessel," *IET Control Theory Appl.*, vol. 6, no. 3, pp. 461–466, Feb. 2012.

[21] J. A. Farrell and M. M. Polycarpou, *Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation Approaches*. New York, NY, USA: Wiley, 2006.

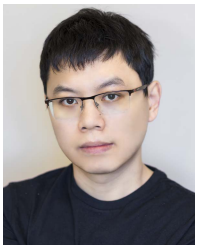
[22] H. Dahmani, O. Pages, A. El Hajjaji, and N. Daraoui, "Observer-based robust control of vehicle dynamics for rollover mitigation in critical situations," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 274–284, Feb. 2014.

[23] Y. Li, B. Yang, T. Zheng, Y. Li, M. Cui, and S. Peeta, "Extended-state-observer-based double-loop integral sliding-mode control of electronic throttle valve," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2501–2510, Oct. 2015.

[24] Q. Zhang and H. H. T. Liu, "Robust nonlinear close formation control of multiple fixed-wing aircraft," *J. Guid., Control, Dyn.*, vol. 44, no. 3, pp. 572–586, Mar. 2021.

[25] W. Shi, S. Song, C. Wu, and C. L. P. Chen, "Multi pseudo Q-learning-based deterministic policy gradient for tracking control of autonomous underwater vehicles," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 12, pp. 3534–3546, Dec. 2019.

[26] T. Shen and K. Tamura, "Robust H_∞ control of uncertain nonlinear system via state feedback," *IEEE Trans. Autom. Control*, vol. 40, no. 4, pp. 766–768, Apr. 1995.
[27] X. Liu, H. Su, B. Yao, and J. Chu, "Adaptive robust control of a class of uncertain nonlinear systems with unknown sinusoidal disturbances," in *Proc. 47th IEEE Conf. Decis. Control*, Cancun, Mexico, Dec. 2008, pp. 2594–2599.
[28] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.
[29] Q. Zhang and H. H. T. Liu, "Aerodynamic model-based robust adaptive control for close formation flight," *Aerosp. Sci. Technol.*, vol. 79, pp. 5–16, Aug. 2018.
[30] B. Zhu, Q. Zhang, and H. H. T. Liu, "Design and experimental evaluation of robust motion synchronization control for multivehicle system without velocity measurements," *Int. J. Robust Nonlinear Control*, vol. 28, no. 17, pp. 5437–5463, Nov. 2018.
[31] Q. Zhang and H. H. T. Liu, "UDE-based robust command filtered backstepping control for close formation flight," *IEEE Trans. Ind. Electron.*, vol. 65, no. 11, pp. 8818–8827, Nov. 2018.
[32] S. Mondal and C. Mahanta, "Chattering free adaptive multivariable sliding mode controller for systems with matched and mismatched uncertainty," *ISA Trans.*, vol. 52, no. 3, pp. 335–341, May 2013.
[33] Z. Cao, Q. Xiao, and M. Zhou, "Distributed fusion-based policy search for fast robot locomotion learning," *IEEE Comput. Intell. Mag.*, vol. 14, no. 3, pp. 19–28, Aug. 2019.
[34] T. Bian and Z.-P. Jiang, "Reinforcement learning for linear continuous-time systems: An incremental learning approach," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 2, pp. 433–440, Mar. 2019.
[35] T. Liu, B. Tian, Y. Ai, and F.-Y. Wang, "Parallel reinforcement learning-based energy efficiency improvement for a cyber-physical system," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 2, pp. 617–626, Mar. 2020.
[36] J. Woo, C. Yu, and N. Kim, "Deep reinforcement learning-based controller for path following of an unmanned surface vehicle," *Ocean Eng.*, vol. 183, pp. 155–166, Jul. 2019.
[37] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, "Safe model-based reinforcement learning with stability guarantees," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, Dec. 2017, pp. 1–20.
[38] R. S. Sutton, A. G. Barto, and R. J. Williams, "Reinforcement learning is direct adaptive optimal control," *IEEE Control Syst.*, vol. 12, no. 2, pp. 19–22, Apr. 1992.
[39] Q. Zhang, W. Pan, and V. Reppa, "Model-reference reinforcement learning control of autonomous surface vehicles with uncertainties," 2020, *arXiv:2003.13839*. [Online]. Available: https://arxiv.org/abs/2003.13839
[40] J. Woo and N. Kim, "Collision avoidance for an unmanned surface vehicle using deep reinforcement learning," *Ocean Eng.*, vol. 199, Mar. 2020, Art. no. 107001.
[41] X. Xu, Y. Lu, X. Liu, and W. Zhang, "Intelligent collision avoidance algorithms for USVs via deep reinforcement learning under COLREGs," *Ocean Eng.*, vol. 217, Dec. 2020, Art. no. 107704.
[42] R. Cui, C. Yang, Y. Li, and S. Sharma, "Adaptive neural network control of AUVs with control input nonlinearities using reinforcement learning," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 6, pp. 1019–1029, Jun. 2017.
[43] N. Wang, Y. Gao, H. Zhao, and C. K. Ahn, "Reinforcement learning-based optimal tracking control of an unknown unmanned surface vehicle," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Aug. 3, 2020, doi: 10.1109/TNNLS.2020.3009214.
[44] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. ICML*, Stockholmsmässan, Sweden, vol. 80, Jul. 2018, pp. 1861–1870.
[45] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 8609–8613.
[46] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.
[47] T. Haarnoja *et al.*, "Soft actor-critic algorithms and applications," 2018, *arXiv:1812.05905*. [Online]. Available: https://arxiv.org/abs/1812.05905
[48] Y. Jiang and Z.-P. Jiang, "Global adaptive dynamic programming for continuous-time nonlinear systems," *IEEE Trans. Autom. Control*, vol. 60, no. 11, pp. 2917–2929, Nov. 2015.
[49] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.
[50] R. Skjetne, T. I. Fossen, and P. V. Kokotović, "Adaptive maneuvering, with experiments, for a model ship in a marine control laboratory," *Automatica*, vol. 41, no. 2, pp. 289–298, Feb. 2005.
[51] Z. Peng, D. Wang, T. Li, and Z. Wu, "Leaderless and leader-follower cooperative control of multiple marine surface vehicles with unknown dynamics," *Nonlinear Dyn.*, vol. 74, nos. 1–2, pp. 95–106, Oct. 2013.



Qingrui Zhang (Member, IEEE) received the B.S. degree in automatic control from the Harbin Institute of Technology, Harbin, China, in 2013, and the Ph.D. degree in aerospace science and engineering from the University of Toronto, Toronto, ON, Canada, in 2019. He was a Post-Doctoral Research Fellow with the Delft University of Technology (TU Delft), Delft, The Netherlands. He is currently an Assistant Professor with Sun Yat-sen University, Guangzhou, China. His research interests include reinforcement learning, robust nonlinear control, adaptive control, multi-robot/agent systems, unmanned aerial vehicle systems, and aerial swarms. He was a recipient of the Gordon N. Patterson Student Award for the top Ph.D. graduate from the University of Toronto Institute for Aerospace Studies (UTIAS) in 2019.

Wei Pan (Member, IEEE) received the Ph.D. degree in bioengineering from Imperial College London in 2016. Until May 2018, he was a Project Leader with DJI, Shenzhen, China, responsible for machine learning research for DJI drones and AI accelerator. He is currently an Assistant Professor with the Department of Cognitive Robotics, Delft University of Technology. His research interests include machine learning and control theory with applications in robotics. He was a recipient of the Dorothy Hodgkin's postgraduate awards, the Microsoft Research Ph.D. Scholarship and Chinese Government Award for Outstanding Students Abroad, and the Shenzhen Peacock Plan Award. He is on the editorial board of CoRL, IROS, and *IET Cyber-Systems and Robotics*.

Vasso Reppa (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Patras, Greece, in 2010. In 2009, she joined the IBM Zurich Research Laboratory, Switzerland, as a Student Intern. From 2011 to 2017, she was a Research Associate (now a Research Affiliate) with the KIOS Research and Innovation Center of Excellence, Cyprus. In 2013, she was awarded the Marie Curie Intra European Fellowship and worked as a Research Fellow with the CentraleSupélec, University of Paris-Saclay, France, from 2014 to 2016. She was a Visiting Researcher with Imperial College London, U.K., and The University of Newcastle, Australia, in 2015 and 2016, respectively. She has been an Assistant Professor with the Department of Maritime and Transport Technology, Delft University of Technology, The Netherlands, since 2018. Her research interests include distributed fault diagnosis and fault tolerant control, adaptive learning, observer-based estimation, and applications of autonomous systems in transport, smart buildings, and robotics. She has led the implementation of work packages of several research and development projects (e.g., FP7, INTERREG, H2020, and NWO).