BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

CÔNG TRÌNH THAM DỰ
GIẢI THƯỞNG "SINH VIÊN NGHIÊN CỨU KHOA HỌC" CẤP TRƯỜNG
NĂM 2020-2021

Tên công trình:

MULTIAGENT CONTROL WITH ADAPTIVE REINFORCEMENT LEARNING STRATEGY FOR SURFACE VESSELS

Mã số:

Họ và tên sinh viên	Lớp, khóa	Giới tính	Khoa/Viện	Tel
Phạm Đình Dương	CTTN - ĐKTĐ K62	Nam	Điện	0989466318
Nguyễn Xuân Khải	CTTN - ĐKTĐ K62	Nam	Điện	0394491201

Giáo viên hướng dẫn: TS. Đào Phương Nam

SUMMARY

This article proposes a complete control structure including formation control and adaptive reinforcement learning (ARL) algorithm for multiagent system of surface vessels (SVs). The ARL strategy is established for each SV to process non-autonomous system without solving Hamilton-Jacobi-Bellman (HJB) equation. The additional formation controller is implemented to complete control structure of multiple SV systems and to guarantee the formation tracking problem. Simulation studies are developed to show the performance of proposed control structure..

Keywords: adaptive reinforcement learning, surface vessel, actor-critic algorithm, formation control, multiagent systems.

I. INTRODUCTION

Surface Vessel (SV) systems are the special case of robotic systems and we can absolutely utilize the control structures in general robotics in considering control design of SVs. Recent years have witnessed the control design development for surface vessel (SV) systems with dynamic uncertainties and external disturbances [1-11]. Among numerous approaches to enhance robust adaptive controllers, the model of SV systems can be considered in two cases involving the under-actuated systems [1-4] and fully-actuated systems [5-10]. There is a similarity between under-actuated SV systems and a typical class of nonholonomic system, such as a wheeled mobile robotic (WMR) system [12]. Thus, model separation technique in control law of [12] can be utilized for a class of underactuated SV systems. However, it can be seen that, in contrast to the under-actuator WMRs, the kinematic subsystem and dynamic sub-system are fully-actuated and under-actuated, respectively. Therefore, this means that the control design in [1-3] does not utilize the transformation matrix to be introduced in [12]. Authors in [1] developed an output feedback control scheme with a neural network based adaptive observer. Additionally, due to the model separation technique, the control scheme was investigated by backstepping method based on considering the additional dynamic term in handling the actuator saturation and observer [1]. In [2], because of the description of under-actuated systems, the coordinate transformations was considered to separate the surface vessel into two subsystems including rotational and translational subsystems. The control law of Yaw and Surge are dealt with rotational and translational subsystems, respectively. Moreover, the authors in [2][developed the stability analysis for cascade system with finite-time uncertainty observer. This method is extended in control structure [3] with the situation of full state regulation control. The backstepping technique for under-actuated model was implemented sequentially from kinematic model to dynamic subsystem with transformation being mentioned [4,11]. Furthermore, the proposed controller is able to handle input saturation constraint using the smooth bounded function [4].

Regarding fully-actuated SV systems, the control design of uncertain systems has become considerably challenging in relation to input, full-state constraints, finite time [5-10]. To tackle these challenges, the backstepping based robust adaptive control scheme and barrier Lyapunov function (BLF) have been cleverly combined to obtain the appropriate controllers [5-9]. In [5], although the cascade control system is also considered in the situation of fully-actuated SVs, but it is obviously different from the existing methods in [1], the tan-BLF technique addressed the error constraint and finite time control problem. In [10], the control design can be investigated with non-singular fast terminal SMC technique for whole systems without using Backstepping technique for subsystems. This is in contrast to classical Backstepping technique for SVs to be introduced in [5-9]. Additionally, the finite time disturbance observer is also inserted to guarantee the finite-time reachability of the sliding surface [10]. On the other hand, when the state trajectories of closed system is located on the sliding surface, the finite-time tracking problem can be satisfied under the description of non-singular fast terminal of this sliding surface [10]. In [8] and [9], the integral sliding surface is mentioned to obtain the sliding mode control (SMC) strategy combining with NN being employed to approximate the uncertainty term as well as the backstepping method is also handled for designing the controller. For fixed-time tracking control scheme, many efforts have been made in the recent time, such as the controllers in [6,7] are designed by using the framework of exponential function based controller and a fixed-time extended state observer. Moreover, the fixed-time convergence can be achieved under the estimation of the appropriate Lyapunov function and the lemma of Polyakov [7-13]. In comparison with [6], the proposed SMC scheme in [7] is extended for sensor fault diagnosis. However, the fixed-time controllers presented in 6] and [7] are disadvantageous for underactuated systems and uncertain matrix

M. A new structural reliability based matrix for fully-actuated systems is presented in [14] to lead the computational complexity of controller. Additionally, similar to [15], the controller is added more the term Nussbaum-type function to deal with the unknown coefficient sign. The backstepping method is absolutely extended with the disturbance observer (DO) using signum function for control design of multiple dynamic positioning vessels with the auxiliary term to cope with input constraint [16]. It is similar to the work in [15,16], the challenges of input saturation and state constraint have been tackled by the framework of BLF, additional dynamic term and Nussbaum-type function [17]. The control designs in [13,18,19] are implemented by the same control structure but the work in [13] is extended with fixed-time control based on the theorem by Polyakov about fixed-time stability for DO design of external disturbances and the trajectory tracking control is established by fixed-time command filter for backstepping method. Authors in [19] handle more the description of the unknown parameters with exponential convergence.

Different from the classical solution dealing with the full state constraints, input saturation via additional dynamic term, barrier Lyapunov function [15-17], the optimal control scheme can handle by solving the constrained optimal problem. However, under the mathematical viewpoint, finding an optimal controller is equivalent to solving the nonlinear partial differential equation Hamilton–Jacobi–Bellman (HJB) equation, which is difficult to obtain a global analytic solution. Therefore, many adaptive/approximate Reinforcement Learning (ARL) based recent investigations have focused on the approximate optimal methods instead of the accurate optimal ones. The main technique of ARL can be known as the iterative method, which has been extended to many approaches, such as Actor/Critic technique [20-22], off-policy based integral reinforcement learning (IRL) [23-25], Q learning method [26,27}, etc. The optimal control input can be computed by simultaneously training of both Actor NN and Critic NN. Moreover, the disadvantages of actuator saturation and external disturbances can be solved by modified performance index [21]. The Q learning method is established using Q function being a function of both the state variable and control input. Based on the relation between Bellman function and Q function, the optimal controller can be achieved in the case of dynamic uncertainties. The IRL method in [23, 25] is implemented by the consideration of integration on an interval combining with data collection to compute optimal control. This method is able to cope with dynamic uncertainties via off-policy technique [23-25]. In [24], thanks to the performance index being not classical quadratic form, the critic NN, which only depend on temporal difference error, can be found without using Halminton function. Additionally, in the step that computing Actor NN from Critic NN, this method can eliminate the influence of control input by Nussabaum function [24]. A different approach of ARL technique in designing for unknown dynamic can be regarded using identifier [22,28-31]. It can be seen that the control methods can be generally categorized into two main groups, including classical nonlinear control techniques and optimal control approaches.

Today, the problem of controlling the multi-object cooperative system is studied widely in the field of control and automation. In many applications, it is interested in the synchronization of objects to achieve combined goals [34-37]. The problem of many cooperative objects takes the form of objects following the leading audience (Leader-Follower), the distributed herd (Distributed Swarms) or synchronizing squad. In those studies, graph theory was used as background knowledge to design the communication diagram of a cooperated multi-object system, which includes multiple nodes and branch links [38-40]. The system at each node in the graph is a linear system [41,42] or nonlinear system [30,40]. Based on the approximate probability, NN is used to design adaptive collaborative controllers [40,43-45]. In [40], Distributed NN is used to design a cooperative input feedback control

algorithm output for many nonlinear systems with a dynamic component that lacks specific information. In [43,45] algorithm for sustainable adaptive control based on NN is designed for many nonlinear systems that follow the trajectory of the leading subject with blocked grip error. In [44], the NN based adaptive control algorithm is designed for many collaborative nonlinear systems using linearization techniques. Most of the aforementioned algorithms do not minimum any quality index function so they are not considered to be the optimal control algorithms. Combining the properties of the thing Adaptive and optimal control for the cooperative control problem is essential. However, here is a complex and challenging problem. On the other hand, multi-agent systems have been mentioned by many approaches [46-50]. In [48], the distributed control for multiagent systems was presented with the consideration of Kronecker product, Neural Networks, Linear Matrix Inequalities (LMIs). Moreover, LQR optimal control was developed for multi-agent systems [47]. However, almost previous control designs for multi-agent systems have considered each agent in easy cases of linear systems, sub-systems [46-50]. Additionally, optimal solutions for multiagent systems have not mentioned the obstacle in solving HJB equation [46-50]. The ARL algorithm shows the efficiency in computational costs and storage resources to speed convergence. Extending the ARL to the problem of controlling multi-object collaboration is necessary.

Motivated by the above works and consideration from traditional nonlinear control scheme to optimal control strategy for multi-agent systems, the work focuses on the combination of these two control direction with main contribution being listed in the following:

1. The proposed ARL based control scheme can achieve simplicity in calculation in compare with the existing papers [20,32,33]. The fact is that the proposed cascade control design is only realized ARL algorithm in dynamic sub-system control loop. Furthermore, the proposed method is able to carry out for autonomous systems with a smaller number of state variables despite of the nonautonomous property of closed systems under time-varying reference trajectory.
2. This paper proposed the control structure being the frame of formation control for generating the reference and ARL based optimal control for each SV. The proposed algorithm is able to obtain the tracking of formation, trajectory tracking control and the unification of optimality problem and stability.

The remaining parts are summarized as follows. The mathematical model of SV systems and control objective, the proposed ARL control scheme and theoretical analysis are shown in Section 2.1 and Section 2.2 respectively. Graph theory and formation control are presented in Section 2.3. Overall control structure of SV formation is summarized in Section 2.4. Section 2.5 verify the proposed control approach by numerical simulation.

II. RESEARCH OUTPUT

2.1 SURFACE VESSELS MODEL

This section presents the model and the mathematical definition of the tracking control problem for SV systems. Ignoring the motion in heaven, roll and pitch axes, one can only consider the mathematical model of SVs in the case of three degree-of-freedom (3DOF). This allows us to model the SV with the dynamic equation given in a matrix form.

$$\begin{aligned} &\dot{\eta} = J(\eta)v(t) \\ &M(\eta)\dot{v} + C(v)v + D(v)v + g(\eta) = \tau + \Delta(\eta, v) \end{aligned} \tag{1}$$

where the vector $\eta = [x, y, \psi]^T \in \mathbb{R}^3$ includes the planar position (x, y) and heading angle ψ in the earth-fixed frame. $v = [u, \vartheta, r]^T \in \mathbb{R}^3$ represents the corresponding linear velocities with surge, sway velocities and yaw in the body-fixed frame of SV systems; and $\tau \in \mathbb{R}^3$ is the control input vector. The rotation matrix $J(\eta)$ is represented in the following matrix

$$J(\eta) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

the inertial matrix of system

$$M = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ 0 & mx_g - N_{\dot{v}} & I_z - N_{\dot{r}} \end{bmatrix} \tag{3}$$

the Coriolis matrix

$$C(v) = \begin{bmatrix} 0 & 0 & -(m - Y_{\dot{v}})v - (mx_g - Y_{\dot{r}})r \\ 0 & 0 & (m - X_{\dot{u}})u \\ (m - Y_{\dot{v}})v + (mX_g - Y_{\dot{r}})r & -(m - X_{\dot{u}})u & 0 \end{bmatrix} \tag{4}$$

The hydrodynamic reduction matrix:

$$\begin{aligned} &D(v) = D + D_n(v) \\ &D = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix} \\ &D_n(v) = \begin{bmatrix} -X_{|u|u}|u| & 0 & 0 \\ 0 & -Y_{|v|v}|v| - Y_{|r|v}|r| & -Y_{|v|r}|v| - Y_{|r|r}|r| \\ 0 & -N_{|v|v}|v| - N_{|r|v}|r| & -N_{|v|r}|v| - N_{|r|r}|r| \end{bmatrix} \end{aligned} \tag{5}$$

$g(\eta)$ is a vector of thrust and gravity, a ship with three degrees of freedom can be considered as $g(\eta) = 0$. However, noise from the environment can be affected to tilt the ship, then force and thrust will appear to bring the ship back into position balance. Therefore, there is no loss of generality while in formula (1) remains with component $g(\eta)$.

The system (1) has the following characteristics:

1. $M - M^T > 0$
2. $C(v) = -C^T(v)$ (6)
3. $D(v) > 0$
4. $J(\eta)$ is the matrix that revolves around axis Z and $J^{-1}(\eta) = J^T(\eta)$

Assumption 1: The term of uncertainties and disturbances $\Delta(\eta, v)$ in (1) is bounded as

$$\|\Delta(\eta, v)\| \le \bar{\Delta} \quad (7)$$

where $\bar{\Delta}$ is a known constant.

In this context, the control objective is to implement an ARL-based trajectory tracking cascade control scheme of surface vessel (1) suffering from the term of uncertainties and disturbances $\Delta(\eta, v)$ and then formation trajectory generator.

It can be seen that Assumption 1 is reasonable as analyzed in [8]. However, compared with the corresponding assumption in [8], this work has an advantage in that it is able to eliminate the condition of bound of its derivative $\frac{d}{dt}\|\Delta(\eta, v)\|$. Unlike previous approaches in [1] - [19], the control objective not only requires trajectory tracking but also guarantees optimal control problem. Due to the challenge of directly implementing the optimal control design, this paper develops the ARL based trajectory tracking cascade controller in next sections. Furthermore, in contrast to the control objective in [20], this work allows us to deal with uncertainties and disturbances $\Delta(\eta, v)$ in SV systems (1).

2.2 ADAPTIVE REINFORCEMENT LEARNING CONTROL STRATEGY

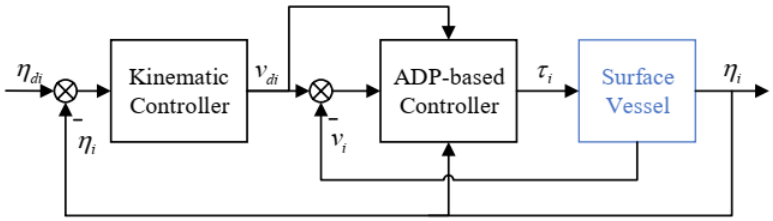

Figure 1 Cascade control structure for each Surface Vessel

Since the optimal controller will be utilized according to the dynamic subsystem, it thus is reasonable to establish the cascade control system as represented in Figure 1. First, the kinematic control law computes the desired velocities vector for inner control loop. Second, the dynamic controller is the frame of optimal controller and feed forward term. Additionally, due to the difficulties in handling the optimal control algorithm, the ARL strategy is considered with actor-critic structure. After that, the stability of whole surface vessels is determined by using Lyapunov stability theory without traditional backstepping technique.

2.2.1 Control Law and Feed Forward Design

For the kinematic sub-system $\dot{\eta} = J(\eta)v$ in a SV (1) ,the kinematic controller can be designed as

$$v_d(z_\eta, \eta_d) = J^{-1}(\eta)(\dot{\eta}_d - \beta_\eta z_\eta) \quad (8)$$

where β_η is a positive definite matrix. Due to the purpose of utilizing the ARL algorithm, it is necessary to obtain the autonomous systems via feed-forward controller. The proposed ADP-based controller for dynamic subsystem in Figure 1 is structured by not only the optimal control input but also the feed-forward term as follows

$$\tau = u + \tau_d \tag{9}$$

where the feed-forward term is chosen for the purpose of obtaining the autonomous systems Figure 1 in the development of ARL algorithm.

$$\tau_d = M(v)\dot{v}_d + C(v_d)v_d + D(v_d)v_d + g(\eta) \tag{10}$$

One can replace the control law (9) and (10) into (1) to achieve the dynamic equation with $u(t)$ being satisfied the following differential equation:

$$\begin{aligned}\dot{X} &= \begin{bmatrix} -M^{-1}l(z_v + v_d(z_{\eta,} n_d)) + M^{-1}l(v_d(z_{\eta,} n_d)) \\ J(z_\eta + \eta_d)z_v - \beta_\eta z_\eta \\ h_1(\eta_d) \end{bmatrix} + \begin{bmatrix} M^{-1} \\ 0 \\ 0 \end{bmatrix} u + \begin{bmatrix} M^{-1} \\ 0 \\ 0 \end{bmatrix} \Delta(\eta, v) \\ &= F(X) + G(X)(u + \Delta(\eta, v))\end{aligned} \tag{11}$$

where

$$\begin{aligned} l_1 &= l\left(z_v + v_d(z_\eta, \eta_d)\right) \\ X &= [z_v^T, z_\eta^T, \eta_d^T]^T \\ l(x) &= C(x)x + D(x)x \end{aligned} \tag{12}$$

Remark 1: Thanks to the squared rotation matrix (2) in model (1), the kinematic controller is designed without using the transformation matrix in [12]. Additionally, it should be emphasized that the combination of kinematic controller (8), feedforward control law (10) and considering a new state variable (12) is able to obtain the autonomous systems (11). Hence, this proposed method plays an important role in developing optimal control design in next sections. In addition, though the control method of using a Moore-Penrose pseudo inverse matrix proposed in [33] also implement for autonomous systems, an important difference between this paper and [33] is that the number of state variables (12) in equivalent autonomous model (11) is 9, while the completed model in [33] used the state variables vector with 12 elements. Therefore, it leads to the distinction in handling ARL based optimal control algorithm in next sections.

2.2.2 Actor-Critic Architecture based Control Scheme

In this section, based on Actor-Critic structure, we aim to establish ARL control scheme for the dynamic subsystem of the surface vessel.

a) Optimal Control Problem

In this paper, one consider the finite horizon integral performance index associated with (1) as

$$J(X, u) = \int_0^\infty r(X(\tau), u(\tau))d\tau \tag{13}$$

where $r(X(\tau), u(\tau)) = X^T QX + u^T Ru$ and the user defined weighting matrices $Q = Q^T \geq 0$ and $R = R^T > 0$ with appropriate dimensions. It should be noted that the optimal

problem for surface vessel (11) is to design an admissible control policy [34] obtaining the minimal cost. The fact is that the state feedback control policy $\hat{u}(X)$ guarantees the existence of solution of estimated Bellman function $\hat{V}(X)$ in HJB equation, which will be analyzed in next sections.

Definition 1: A control scheme $u(X)$ is defined to be admissible in term of the performance index (13) on a compact set Ω, known as continuous signal $u(X) \in \mathrm{Y}(\Omega)$ if it satisfies not only the stabilization of (11) but also the limitation of $J(X,u)$ for every $X \in Q$.

It should be noted that, in general case of nonautonomous systems, the optimal state feedback control law needs to be given as a time-varying function $u^*(X,t)$. However, thanks to the advantage of the framework of kinematic controller, feed-forward term and state variables selection, one obtain an autonomous system (11). Hence, the optimal control input is also determined as a time-invariant function $u^*(X)$ and the Bellman function with respect to arbitrary time $V^*(X(t))$ can be known as $V^*(X(t)) = \min_{u(X)\in \mathrm{Y}(\Omega)} J(X(t),u(t))$. Taking the time derivative of $V^*(X(t))$ by using two different methods. First, one can compute directly as

$$\frac{d}{dt}V^*(X(t)) = \frac{\partial V^*}{\partial X}\frac{dX}{dt} = \frac{\partial V^*}{\partial X}(F(X)+G(X)u^*) \tag{14}$$

Second, thanks to the Dynamic Programming Law, one also obtain the derivative of $V^*(X(t))$ as follows

$$\frac{d}{dt}V^*(X(t)) = -r(X(\tau),u^*(\tau)) \tag{15}$$

According to (14) and (15) one obtain that

$$r(X(\tau),u^*(\tau)) + \frac{\partial V^*}{\partial X}(F(X)+G(X)u^*) = 0 \tag{16}$$

Additionally, one have the cost function formulated as

$$\begin{aligned} V^*(X(t)) &= \min_{u(X)\in \mathrm{Y}(\Omega)} \int_t^{\infty} r(X(\tau),u^*(\tau))d\tau \\ &= \min_{u(X)\in \mathrm{Y}(\Omega)} \int_t^{t+\Delta t} r(X,u^*)d\tau + \min_{u(X)\in \mathrm{Y}(\Omega)} \int_{t+\Delta t}^{\infty} r(X,u^*)d\tau \end{aligned} \tag{17}$$

Because of Dynamic programming principle, it leads to

$$V^*(X(t)) = \min_{u(X)\in \mathrm{Y}(\Omega)} \int_t^{t+\Delta t} r(X,u^*)d\tau + V^*(X(t+\Delta t)) \tag{18}$$

This Bellman function can be rewritten as

$$\min_{u(X)\in \mathrm{Y}(\Omega)} \left\{ \frac{1}{\Delta t}\int_t^{t+\Delta t} r(X,u)d\tau + \frac{V^*(t+\Delta)-V^*(t)}{\Delta t} \right\} = 0 \tag{19}$$

As the convergence of $\Delta t \to 0^+$, one can derive that

$$\min_{u(X)\in \Upsilon(\Omega)} \left\{ r(X(\tau),u(\tau)) + \frac{\partial V^*}{\partial X}(F(X)+G(X)u) \right\} = 0 \tag{20}$$

Remark 2: It is worth noting that in general case of the time-varying control policy and Bellman function $V^*(X(t),t)$, the equation (16) is modified with the right side being $\frac{\partial}{\partial t}V^*(X(t),t)$. On the other hand, due to the time-varying desired trajectory $\eta_d(t)$, the closed systems need to be considered as a non-autonomous system. In order to overcome this challenge, the proposed method in [35] developed the solutions of ARL based time-varying optimal controller using the combination of data collection and function approximation technique under Newton-Leibniz formula. However, the proposed algorithm in [35] is more complicated as handling for the term $\frac{\partial}{\partial t}V^*(X(t),t)$. It is worth noting that the advantage of the proposed method is that one only need to deal with optimal control problem for an autonomous system (11) using the framework of kinematic controller (8), feed-forward (10) and a new state variables vector (12)

b) ADP-based Control Design

It is worth emphasizing that, due to the nonlinear property of HJB equation (16) , it is hard or impossible to solve analytically for obtaining the optimal controller. Hence, a Neural Networks (NN) based approximation method is utilized to develop the ARL algorithm in control design of SVs. As we all known, since the Bellman function $V^*(X)$ and optimal control input $u^*(X)$ can be considered smooth functions of the state X, they are represented over any compact domain $C \subseteq \mathbb{R}^9$

$$V^*(x) = \mathrm{W}^T\phi(x) + \varepsilon(x) \tag{21}$$

$$u^*(x) = -\frac{1}{2}R^{-1}G^T(x)\left(\frac{\partial\phi(x)}{\partial x}^T W + \frac{\partial\varepsilon(x)}{\partial x}^T\right) \tag{22}$$

where $\mathrm{W} \in \mathbb{R}^N$ is a vector of unknown ideal NN weights, N is the number of neurons of the proposed Neural Network, $\Psi(X) \in \mathbb{R}^N$ is a smooth NN activation function vector with $\psi_j(0) = 0$ and $\frac{\partial \psi_j}{\partial x}|_{x=0} = 0 \forall j = 1,...,N$, $\varepsilon(X) \in \mathbb{R}^N$ is the reconstruction error of the Bellman function $V^*(X)$. It is because of uncertain ideal NN weights, one need to find appropriate updating laws $\mathrm{W}_a, \mathrm{W}_c$ with the purpose of approximating the actor/critic parts and obtaining the optimal controller without solving analytically the HJB equation (more details see [22]). In addition, the smooth NN activation function vector $\Psi(X) \in \mathbb{R}^N$ is chosen based on the description of SVs (see Section 2.1). In [22], the Weierstrass approximation theorem is able to uniformly approximate not only $V^*(X)$ but also $\frac{\partial V^*(X)}{\partial X}$ with $\varepsilon(x), \left(\frac{\partial\varepsilon(x)}{\partial x}\right) \to 0$ as $N \to \infty$. For a fix number N, the estimated Bellman function of critic part $\hat{V}(X)$ and the estimated optimal control policy of actor part $\hat{u}(X)$ are employed to approximate the Bellman function and the optimal control input as

$$\hat{V}(X) = \hat{W}_c^T \Psi(X) \tag{23}$$

$$\hat{u}(X) = -\frac{1}{2} R^{-1} G^T(X) (\frac{\partial \Psi}{\partial x})^T \hat{W}_c \tag{24}$$

To this step of analysis, based on the properties (16), (20) of Hamiltonian $H(X, u, \frac{\partial V}{\partial X}) = r(X(\tau), u(\tau)) + \frac{\partial V}{\partial X}(F(X) + G(X)u)$ under the optimal control input $u^*(X)$ and associated value function $V^*(X)$, the adaptation laws of critic $\hat{W}_a$, $\hat{W}_c$ weights are simultaneously trained to minimize the squared Bellman error δ_{hjb} and the corresponding integral, respectively. Due to the error between estimated functions $\hat{V}(X)$, $\hat{u}(X)$ and optimal results $V^*(X)$, $u^*(X)$ the Bellman error δ_{hjb} can be computed as

$$\begin{aligned} \delta_{hjb} &= \hat{H}(X, \hat{u}, \frac{\partial \hat{V}}{\partial X}) - H^*(X, u^*, \frac{\partial V^*}{\partial X}) \\ &= \hat{W}_c^T \sigma(X, \hat{u}) + \frac{1}{2} X^T Q X + \frac{1}{2} \hat{u}^T R \hat{u} \end{aligned} \tag{25}$$

where $\sigma(X, \hat{u}) = \frac{\partial \Psi}{\partial X}(F(X) + G(X)\hat{u})$ is the regression vector of critic part.

Similar to the work in [22], the adaptation law of Critic weights is given

$$\frac{d}{dt}\hat{W}_c = -k_c \lambda \frac{\sigma}{1 + \nu \sigma^T \lambda \sigma} \delta_{hjb} \tag{26}$$

where $\nu, k_c \in \mathbb{R}$ are constant positive gains, and $\lambda(t) \in \mathbb{R}^{N \times N}$ is a estimated symmetric gain matrix obtained from the differential equation as

$$\frac{d}{dt}\lambda = -k_c \lambda \frac{\sigma \sigma^T}{1 + \nu \sigma \lambda \sigma^T} \lambda; \quad \lambda(t_s^+) = \lambda(0) = \varphi_0 I \tag{27}$$

where t_s^+ is resetting time satisfying the property of eigenvalue $\lambda_{\min}\{\lambda(t)\} \le \varphi_1, \varphi_0 > \varphi_1 > 0$. In [22], to ensure $\lambda(t)$ is positive definite and prevent the covariance wind-up problem, the covariance matrix $\lambda(t)$ can be satisfied as

$$\varphi_1 I \leqslant \lambda(t) \le \varphi_0 I \tag{28}$$

In addition, the adaptation law of actor NN part is proposed using the minimization of squared Bellman error.

$$\frac{d}{dt}\hat{W}_a = -\frac{k_{a1}}{\sqrt{1 + \sigma^T \sigma}} \frac{\partial \Psi}{\partial X} G R^{-1} G^T \frac{\partial \Psi}{\partial X}^T (\hat{W}_a - \hat{W}_c) \delta_{hjb} - k_{a2}(\hat{W}_a - \hat{W}_c) \tag{29}$$

Remark 3: It can be seen that many ARL based optimal control methods have been investigated, such as off-policy Integral Reinforcement Learning [23, 25], Q Learning [27], etc. Compared with the work of off-policy IRL [23, 25], the control design (23), (24), (26), (29) is only based on the instantaneous time instead of considering the time interval to collect data in finding the optimal controller. Moreover, the learning method is simultaneously trained, being different from sequentially implemented in [23, 25]. The fact is that this paper utilized the property of HJB equation (16), while off-policy IRL method considered the deviation of integrals at two sampling time by dynamic programming principle [24, 25]. The Q Learning method establishes the Q-function with respect to both state variables and control

inputs, containing Bellman function and performance index [27]. Therefore, the Q-learning technique is appropriate for discrete time systems and it is hard to implement Q-learning for continuous time systems as in this work.

c) Convergence and Stability Analysis

It is necessary to utilize several following assumptions in considering the stability and tracking problem of proposed algorithm [22].

Assumption 2: The matrix $G(X)$ in (11) is known and bounded, it means that there exists a known positive constant $\bar{G}$, such that $0 < \|G(X)\| \le \bar{G}$.

According to (21), (22), (23), (24), (25) the Bellman error δ_{hjb} is also described by a function of state variables vector $X(t)$ as:

$$\delta_{hjb} = \hat{W}_c^T \omega - \frac{\partial \phi}{\partial X}(F(X) + G(X)u^*) - u^{*T} R u^* + \hat{u}^T R \hat{u} - \frac{\partial \varepsilon}{\partial X}(F(X) + G(X)u^*) \tag{30}$$

Replacing (30) in (26), it leads to the dynamics of critic weight error $\tilde{W}_c = W - \hat{W}_c$ being represented as:

$$\begin{aligned} \dot{\tilde{W}}_c &= -\eta_c \Gamma \psi \psi^T \tilde{W}_c + \eta_c \Gamma \frac{\omega}{1 + \upsilon \omega^T \Gamma \omega}\left(\frac{1}{4}\tilde{W}_c^T \left(\frac{\partial \phi}{\partial X}\right) G R^{-1} G^T \left(\frac{\partial \phi}{\partial X}\right)^T \tilde{W}_a \right. \\ &\left. -\frac{1}{4}\left(\frac{\partial \varepsilon}{\partial X}\right) G R^{-1} G^T \left(\frac{\partial \varepsilon}{\partial X}\right)^T - \frac{\partial \varepsilon}{\partial X}(F(X) + G(X)u^*)\right) \end{aligned} \tag{31}$$

where $\tilde{W}_a = W - \hat{W}_a$ and $\psi(t) = \dfrac{\omega}{\sqrt{1 + \upsilon \omega^T \Gamma \omega}}$ is bounded by

$$\| \psi \| \le \frac{1}{\sqrt{\upsilon \varphi_1}} \tag{32}$$

By eliminating the influence of actor weight error, one can obtain the nominal system as:

$$\dot{\tilde{W}}_c = -\eta_c \Gamma \psi \psi^T \tilde{W}_c \tag{33}$$

Similar to [22], it can be seen that if the $\psi(t)$ satisfies the persistence of excitation (PE) condition (34) then $\tilde{W}_c$ exponentially converges to origin

$$\mu_2 I \ge \int_{t_0}^{t_0 + \delta} \psi(s)\psi(s)^T ds \ge \mu_1 I, \qquad \forall t_0 \ge 0 \tag{34}$$

Theorem 1: Consider the surface vessel (1) with Assumption 1, 2, the bound conditions of ideal NN weights, activation function and its derivative are described in [22], and the signal vector $\psi(t)$ satisfies PE condition as well as the following condition is mentioned:

$$\frac{c_3}{k_{a1}} > k_1 k_2 \tag{35}$$

where the parameters in (35) are considered in (36), (37), (29) .Let's consider the ARL based control scheme (9), using the kinematic controller (8) for the feedforward term (10) and the updating laws (26) (29) for the actual controller (24) then:

1. The actor-critic weight errors $\tilde{W}_a$ and $\tilde{W}_c$ are UUB.

2. The tracking of both $\tilde{z}_v$ and $\tilde{z}_\eta$ in SV systems are also UUB.

Proof: Thanks to the kinematic controller (8), the feedforward term (10) in proposed control scheme (9) enables us to achieve the corresponding model of surface vessels (11) .In order to choose the appropriate Lyapunov function candidate, a part of the completed function can be utilized by a function $V_c : R^N \times [0,\infty) \to R$ satisfying several following inequalities:

$$\begin{aligned} & c_1 \left\| \tilde{W}_c \right\|^2 \le V_c \left(\tilde{W}_c, t \right) \le c_2 \left\| \tilde{W}_c \right\|^2 \\ & \frac{\partial V_c}{\partial t} + \frac{\partial V_c}{\partial \tilde{W}_c} (-\eta_c \Gamma \psi \psi^T \tilde{W}_c) \le -c_3 \left\| \tilde{W}_c \right\|^2 \\ & \left\| \frac{\partial V_c}{\partial \tilde{W}_c} \right\| \le c_4 \left\| \tilde{W}_c \right\| \end{aligned} \tag{36}$$

where $c_1, c_2, c_3, c_4 \in R$ are positive constant coefficients. Based on Assumption 2, and the bound conditions of ideal NN weights, activation function and its derivative [22], one can achieve the bounds of following functions as:

$$\begin{aligned} & \left\| \tilde{W}_c \right\| \le k_1 \\ & \left\| \frac{\partial \phi}{\partial X} G R^{-1} G^T (\frac{\partial \phi}{\partial X})^T \right\| \le k_2 \\ & \left\| \frac{1}{4} \tilde{W}_a^T (\frac{\partial \phi}{\partial X}) G R^{-1} G^T (\frac{\partial \phi}{\partial X})^T \tilde{W}_a - \frac{1}{4} (\frac{\partial \varepsilon}{\partial X}) G R^{-1} G^T (\frac{\partial \varepsilon}{\partial X})^T - \frac{\partial \varepsilon}{\partial X} (F(X) + G(X) u^* \right\| \le k_3 \end{aligned} \tag{37}$$

For considering the stability of the whole of cascade ADP-based control system as well as the convergence of the weights of Actor NN and Critic NN, choose a Lyapunov function candidate as:

$$V_L \triangleq \frac{1}{2} \rho z_n^T z_n + V^*(X) + V_c(\tilde{W}_c, t) \frac{1}{2} \tilde{W}_a^T \tilde{W}_a \tag{38}$$

where $V^*(X)$ is the optimal function associated with optimal control input $u^*(X)$ and $V_c(\tilde{W}_c, t)$ is satisfied the inequality (36).It can be noted that the term $\frac{1}{2} \rho z_n^T z_n$ (ρ is a positive constant coefficient) is added to consider the tracking of the whole control system. Because $V^*(X)$ is a smooth function and positive definite, there exist two class functions α_1, α_2 such that:

$$\alpha_1(\|X\|) \le V^*(X) \le \alpha_2(\|X\|) \tag{39}$$

According to (36) and (39), one can imply that:

$$\frac{1}{2} \rho \|z\eta\|^2 + \alpha_1(\|X\|) + c_1 \left\| \tilde{W}_c \right\|^2 + \frac{1}{2} \left\| \tilde{W}_a \right\|^2 \le V_L \le \frac{1}{2} \rho \|z\eta\|^2 + \alpha_2(\|X\|) + c_2 \left\| \tilde{W}_c \right\|^2 + \frac{1}{2} \left\| \tilde{W}_a \right\|^2 \tag{40}$$

Taking the derivative of V_L along the system trajectory under the control input $\hat{u}(X)$, one can obtain that:

$$\begin{aligned}\dot{V}_L &= \rho z_n^T (J(n) z_v - \beta_n z_n) + \frac{\partial V^*}{\partial X}(F(X) + G(X)\hat{u}) + \frac{\partial V_c}{\partial t} \\ &+ \frac{\partial V}{\partial \tilde{W}_c}(\Omega_{nom} + \Delta_{per}) - \tilde{W}_a^T \dot{\hat{W}}_a + \frac{\partial V^*}{\partial X} G(X)\Delta\end{aligned} \tag{41}$$

where

$$\Omega_{nom} = -\eta_c \Gamma \psi \psi^T \tilde{W}_c \tag{42}$$

$$\begin{aligned}\Delta_{per} &= \eta_c \Gamma \frac{w}{1+\nu\omega^T \Gamma \omega}\left(\frac{1}{4} W_a^T \left(\frac{\partial \phi}{\partial X}\right) G R^{-1} G^T \left(\frac{\partial \phi}{\partial X}\right)^T W_a\right. \\ &\left. - \frac{1}{4}\left(\frac{\partial \epsilon}{\partial X}\right) G R^{-1} G^T \left(\frac{\partial \epsilon}{\partial X}\right)^T - (F(X) + G(X)u^*)\right)\end{aligned} \tag{43}$$

According to (16), it leads that:

$$\frac{\partial V^*}{\partial X} F(X) = -\frac{\partial V^*}{\partial X} G(X) u^*(X) - X^T Q_T X - u^{*T}(X) R u^*(X) \tag{44}$$

Replacing (44) and optimal control (22), (29), (36) in (41), one can obtain that:

$$\begin{aligned}\dot{V}_L &= -\rho\beta_n \|z_n\|^2 + \rho z_n^T J(n) z_v - z_v^T Q z_v - u^{*T} R u^* + 2u^{*T} R(u^* - \hat{u}) - c_3 \|W_c\|^2 \\ &+ c_4 \|W_c\| \|\Delta_{per}\| + \eta_{a2} W_a^T (W_a - W_c) - 2u^{*T} R \\ &+ \frac{\eta_{a1}}{\sqrt{1+w^T w}} W_a^T \frac{\partial \phi}{\partial X} G(X) R^{-1} G^T(X) \left(\frac{\partial \phi}{\partial X}\right)^T (W_a - W_c)\delta_{hjb}\end{aligned} \tag{45}$$

Using Young inequality, one obtain that:

$$\rho z_n^T J(n) z_v \le \frac{\rho}{2}\|z_n\|^2 + \frac{\rho}{2} z_n^T J^T(n) J(n) z_v = \frac{\rho}{2}\|z_n\|^2 + \frac{\rho}{2}\|z_v\|^2 \tag{46}$$

From (24), (16) and (37), one obtain:

$$\begin{aligned}2u^{*T} R(u^* - \hat{u}) &= \frac{1}{2} W^T \frac{\partial \phi}{\partial X} G R^{-1} G^T \left(\frac{\partial \epsilon}{\partial X}\right)^T + \frac{1}{2} W^T \frac{\partial \phi}{\partial X} G R^{-1} G^T \left(\frac{\partial \phi}{\partial X}\right)^T W_a \\ &+ \frac{1}{2}\frac{\partial \epsilon}{\partial X} G R^{-1} G^T \left(\frac{\partial \phi}{\partial X}\right)^T W_a + \frac{1}{2}\frac{\partial \epsilon}{\partial X} G R^{-1} G^T \left(\frac{\partial \epsilon}{\partial X}\right)^T \le \kappa_4\end{aligned} \tag{47}$$

and the term Δ_{per} is bounded by:

$$\|\Delta_{per}\| \le \frac{\eta_c \varphi_0}{\sqrt{\nu \varphi_1}} \kappa_3 \tag{48}$$

According (37) and (30),one can achieve

$$\frac{\eta_{a1}}{\sqrt{1+w^T w}}\tilde{W}_a^T \frac{\partial \Phi}{\partial X} G(X) R^{-1} G^T(X) \Big(\frac{\partial \Phi}{\partial X}\Big)^T (\tilde{W}_a - \tilde{W}_c)\delta_{hjb}$$
$$= \frac{\eta_{a1}}{\sqrt{1+w^T w}}\tilde{W}_a^T \frac{\partial \Phi}{\partial X} G(X) R^{-1} G^T(X) \Big(\frac{\partial \Phi}{\partial X}\Big)^T (\tilde{W}_a - \tilde{W}_c)$$
$$\times\Big(-\tilde{W}_c^T w + \frac{1}{4}\tilde{W}_a^T \Big(\frac{\partial \Phi}{\partial X}\Big) G R^{-1} G^T \Big(\frac{\partial \Phi}{\partial X}\Big)^T \tilde{W}_a - \frac{1}{4}\Big(\frac{\partial \epsilon}{\partial X}\Big) G R^{-1} G^T \Big(\frac{\partial \epsilon}{\partial X}\Big)^T \tag{49}$$
$$-(F(X) + G(X)u^*)\Big) \le \eta_{a1}\kappa_1\kappa_2 \left\| \tilde{W}_c \right\|^2 + \eta_{a1}\kappa_1^2\kappa_2 \left\| \tilde{W}_c \right\|$$
$$+\eta_{a1}\kappa_1\kappa_2\kappa_3 \left\| \tilde{W}_c \right\| + \eta_{a1}\kappa_1^2\kappa_2\kappa_3$$

It can be seen that:

$$\eta_{a2} W_a^T (W_a - W_c) = \eta_{a2} W_a^T (W_c - W_a) \le \eta_{a2}\kappa_1 \left\| W_c \right\| - \eta_{a2} \left\| W_a \right\|^2 \tag{50}$$

Moreover, it should be noted that:

$$-u^{*T} R u^* - 2u^{*T} R\Delta \le \Delta^T R \Delta \le \lambda_{max}(R)\bar{\Delta}^2 \tag{51}$$

Replacing (46) and (50) in (45), one can obtain the estimation that:

$$\dot{V}_L \le -\rho(\beta_\eta - \frac{1}{2})\left\|z_\eta\right\|^2 - z_v^T (Q - \frac{1}{2}\rho I) z_v - (c_3 - \eta_{a1}\kappa_1\kappa_2)\left\| W_c \right\|^2 - \eta_{a2}\left\| W_a \right\|^2$$
$$+\eta_{a1}\kappa_1^2\kappa_2\kappa_3 + \kappa_4 + \Big(\frac{c_4\eta_c\varphi_0}{2\sqrt{\nu\varphi_1}}\kappa_3 + \eta_{a1}\kappa_1\kappa_2\kappa_3 + \eta_{a1}\kappa_1^2\kappa_2 + \eta_{a2}\kappa_1\Big)\left\| W_c \right\| + \lambda_{max}(R)\bar{\Delta}^2 \tag{52}$$

Using the classical inequality $ab \le \gamma a^2 + \frac{1}{4\gamma} b^2$, we have:

$$\dot{V}_L \le -\rho(\beta_\eta - \frac{1}{2})\left\|z_\eta\right\|^2 - z_v^T (Q - \frac{1}{2}\rho I) z_v + \lambda_{max}(R)\bar{\Delta}^2$$
$$-(1-\theta)(c_3 - \eta_{a1}\kappa_1\kappa_2)\left\| W_c \right\|^2 - \eta_{a2}\left\| W_a \right\|^2 + \eta_{a1}\kappa_1^2\kappa_2\kappa_3 \tag{53}$$
$$+\kappa_4 + \frac{1}{4\theta(c_3 - \eta_{a1}\kappa_1\kappa_2)}\Big(\frac{c_4\eta_c\varphi_0}{2\sqrt{\nu\varphi_1}}\kappa_3 + \eta_{a1}\kappa_1\kappa_2\kappa_3 + \eta_{a1}\kappa_1^2\kappa_2 + \eta_{a2}\kappa_1\Big)^2$$

Let's choose the parameters satisfying $\beta_n = \frac{1}{2}, \rho < 2\lambda_{min}(Q), 0 < \theta < 1, c_3 > \eta_{a1}\kappa_1\kappa_2$

Define the vector $z = [z_n^T, z_v^T, \tilde{W}_c^T, \tilde{W}_a^T]^T$ to analyze the tracking problem of the closed system. It can be seen that there exist two K class functions α_3, α_4 satisfying:

$$\alpha_3 \|z\| \le \rho(\beta_\eta - \frac{1}{2})\|z_n\|^2 + z_v^T (Q - \frac{1}{2}\rho I) z_v$$
$$+(1-\theta)(c_3 - \eta_{a1}\kappa_1\kappa_2)\left\| W_c \right\|^2 + \eta_{a2}\left\| W_a \right\|^2 \le \alpha_4 \| z \| \tag{54}$$

Based on (54), the inequality (53) can be written as:

$$\dot{V}_L \le -\alpha_3 \|z\| + \lambda_{max}(R)\bar{\Delta}^2 + \eta_{a1}\kappa_1^2\kappa_2\kappa_3 + \kappa_4 + \frac{1}{4\theta(c_3 - \eta_{a1}\kappa_1\kappa_2)}\left(\frac{c_4\eta_c\varphi_0}{2\sqrt{\nu\varphi_1}}\kappa_3 + \eta_{a1}\kappa_1\kappa_2\kappa_3 + \eta_{a1}\kappa_1^2\kappa_2 + \eta_{a2}\kappa_1\right)^2 \tag{55}$$

It can be evident that $\frac{d}{dt}V_L$ is negative if $z(t)$ lies outside the attraction region as:

$$\Omega_z \triangleq \left\{z : \|z\| \le \alpha_3^{-1}\left(\frac{1}{4\theta(c_3 - \eta_{a1}\kappa_1\kappa_2)}\left(\frac{c_4\eta_c\varphi_0}{2\sqrt{\nu\varphi_1}}\kappa_3 + \eta_{a1}\kappa_1\kappa_2\kappa_3 + \eta_{a1}\kappa_1^2\kappa_2 + \eta_{a2}\kappa_1\right)^2 + \eta_{a1}\kappa_1^2\kappa_2\kappa_3 + \kappa_4 + \lambda_{max}(R)\bar{\Delta}^2\right)\right\} \tag{56}$$

Therefore, one can conclude that $\|z\|$ is UUB with the attraction region (56). Similar to analysis in [22], it can be seen that the size of attraction region is reduced by increasing the number of neurons in Critic NN. The proof of Theorem 1 is completed. □

Remark 4: It is obviously different from the existing methods in [22], the proposed Lyapunov function candidate is added more $\frac{1}{2}\rho z_n^T z_n$ to consider the stability of whole of cascade control system with the additionally proposed estimations (46), (52), and (53). Furthermore, the state variables $z = [z_n^T, z_v^T, \tilde{W}_c^T, \tilde{W}_a^T]^T$ is eliminated the term X_d in estimating the Lie derivative of Lyapunov function. It is noteworthy that the term $\lambda_{max}(R)\bar{\Delta}^2$ of attraction region (56) in dealing with uncertainties/input disturbances $\Delta(\eta, v)$, (1) has a better performance compared with [22]. The purpose of learning the Actor/Critic is to ensure the convergence to optimal controller, optimal value function: $\hat{u}(X) \to u^*(X), \hat{V}(X) \to V^*(X)$ It can be seen that satisfying the PE (Persistent Excitation) condition of $\frac{w}{\sqrt{1+\nu\omega^T\Gamma\omega}} \in R^N$ guarantees the convergences of estimated actor/critic weights $\hat{W}_c$ and $\hat{W}_a$ [22]. It is noteworthy that because this algorithm do not mentioned the identifier design and it focuses on the control design for SVs, the adaptation law of actor NN (29) and the estimations (37) Lie derivative of Lyapunov function candidate (38) have differences compared with the method in [22].

Remark 5: It should be noted that unlike the training method in [20, 21, 33], this paper handles the updating law of Critic NN using the optimality principle of integral of squared Bellman error. It is obviously different from the existing methods in [33], the proposed cascade ADP-based controller based on feed-forward term (10) is established to obtain the equivalent model (11) with the smaller number of variables. The ARL based cascade control design is also mentioned in [20]. However, the tracking problem in [20] is implemented by back-stepping technique with the modified Critic NN, since both of control loops are designed by ARL law. The advantage of this paper is that it can obtain the tracking of the whole closed system by the additional feed-forward term (11) without using classical back-stepping technique. Additionally, this paper considers the influence of Uncertainties/Disturbances $\Delta(\eta, v)$ in SV systems (1) with the proposed attraction region (56)

.

2.3 FORMATION CONTROL OF MULTIAGENT SYSTEMS

2.3.1 Multiagent Systems

Agent: An **entity** which is placed in an **environment** and senses different **parameters** that are used to make a decision based on the goal of the entity. The entity performs the necessary **action** on the environment based on this decision.

The above definition comprises four keywords which can be further elaborated:

1. Entity: Entity refers to the type of the agent. An agent can be a software, e.g. daemon security agents, a hardware component, e.g. thermostat, or a combination of both, e.g. a robot.
2. Environment: This refers to the place where the agent is located. The environment can be a network in the case of traffic monitoring agents, a software when the agent is monitoring the actions of software components, etc. An agent uses the information sensed from the environment for decision making. The environment has multiple features that affect the complexity of an agent-based system: Accessibility, Determinism, Dynamism, Continuity.
3. Parameters: The different types of data that an agent can sense from the environment are referred to as parameters. For instance, the parameters for a soccer robot agent are the position and speed of the team members and opponents, and the position of the ball.
4. Action: Each agent can perform an action that results in some changes in the environment. For example, when a soccer robot kicks a ball the position of the ball changes. An agent can perform a set of discrete or continues actions. In a continues set of actions, the agent can perform unlimited actions, e.g. a soccer game. A discrete set of actions in contrast has a finite set of actions, e.g. an agent controlling a thermostat in a room.

The goal of each agent is to solve its allocated task with some additional constraints, e.g. a deadline. To achieve this aim, the agent first senses parameters from the environment. Empowered with this data, the agent can build up knowledge about the environment. An agent might also use the knowledge of its neighbors. This knowledge along with the history of the previous actions taken and the goal are fed to an inference engine which decides on the appropriate action to be taken by the agent.

While an agent working by itself is capable of taking actions (based on autonomy), the real benefit of agents can only be harnessed when they work collaboratively with other agents. Multiple agents that collaborate to solve a complex task are known as Multi-Agent Systems (MAS).

Multi-Agent Systems (MAS) consist of autonomous entities known as agents which collaboratively solve tasks yet they offer more flexibility due to their inherent ability to learn and make autonomous decisions. Agents use their interactions with neighbour agents or with the environment to learn new contexts and actions. Subsequently, agents use their knowledge to decide and perform an action on the environment to solve their allocated task. It is this flexibility that makes MAS suited to solve problems in a variety of disciplines including computer science, civil engineering, and electrical engineering. To develop MAS require addressing a diverse range of complex challenges such as coordination among agents, learning, and security.

To study MAS, agents and their relations are modeled using graphs. Graphs have been extensively used in computer science for modeling complex systems, e.g. social media, and analyzing them mathematically. When MAS are modeled as a graph, each vertex represents an agent and an edge between two vertices indicates that the two agents are communicating with each other. The actions taken by an agent may potentially change the relations between agents and thus change the structure of the graph. The final decision made by an agent applies to the corresponding graph that might change the edges or structure of the graph.

A graph $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathcal{A})$ consists of a nonempty finite set of nodes $i\in\{1,\ldots,n\}:=\mathcal{V}$, a set of edges or arcs $\mathcal{E}\subset\mathcal{V}\times\mathcal{V}$, and an associated weighted adjacency matrix $\mathcal{A}=[a_{ij}]\in\mathbb{R}^{n\times n}$. In this paper, the considered graphs are assumed to be time invariant, i.e., $\mathcal{A}$ is constant. If $(i,j)\in\mathcal{E}$, agent i can receive information from agent j and agent j is a neighbor of agent i. The set of neighbors for agent i is denoted as $\mathcal{N}_i=\{j\in\mathcal{V}:(i,j)\in\mathcal{E}\}$. Each element a_{ij} of adjacency matrix is the weight associated with edge (i,j) and $a_{ij}>0$ if $(i,j)\in\mathcal{E}$. Otherwise, $a_{ij}=0$. Define the in-degree of node i as $d_i=\sum_{j=1}^{n}a_{ij}$ and indegree matrix as $D=\text{diag}\{d_i\}\in\mathbb{R}^{n\times n}$. Then, the graph Laplacian matrix is $\mathcal{L}=\mathcal{D}-\mathcal{A}$, $\mathcal{L}\in\mathbb{R}^{n\times n}$.

Let $p_i\in\mathbb{R}^d$ be the position of an agent and $p=[p_1^T,\ldots,p_n^T]^T\in\mathbb{R}^{dn}$. For a given motion synchronization task, let $e(p)$ be the synchronization error vector of appropriate dimensions so that $e(p)=0$ when the task is achieved. Consider a continuously differentiable Lyapunov function $V(e)$ satisfying $V(e)\geq 0,\forall e$ and $V(e)=0\Leftrightarrow e=0$. The corresponding gradient control law is

$$\dot{p}_i=-\nabla_{p_i}V:=f_i(e,p),\qquad i\in\mathcal{V}. \tag{57}$$

The original gradient control law in (57) contributes to $\dot{V}(e)=\sum_{i\in\mathcal{V}}-f_i^T f_i\leq 0$ only depending on the positions of agent i and its neighbors. The error dynamics of (57) is

$$\dot{e}=\frac{\partial e}{\partial p}f(e,p) \tag{58}$$

where $f=[f_1^T,\ldots,f_n^T]^T\in\mathbb{R}^{dn}$. Let $\Omega(r)=\{e:V(e)\leq r\}$ where $r\geq 0$ be the level set. The gradient control (57) is convergent if there exists $r_0\geq 0$ such that the trajectory of (58) converges to $e=0$ for any initial error $e_0\in\Omega(r_0)$. In this case, $\Omega(r_0)$ is called the attraction region.

Equation (57) does not consider motion constraints such as nonholonomic dynamics and velocity saturation so that real agents may not be able to follow the gradient flow f_i correctly in some applications. Therefore, the convergence of the entire synchronization system may not be guaranteed. In this paper, we consider general synchronization control tasks that comply with the following gentle assumption. Let $\|\cdot\|$ symbolize the Euclidian norm of a vector or the spectral norm of a matrix.

Assumption 3: For a given synchronization task, functions $V(e)$ and $e(p)$ satisfy the following conditions.

1. $\Omega(r)$ is compact for any $r \geq 0$.
2. There exists $r_0 \geq 0$ such that $e = 0 \Leftrightarrow f = 0$ in $\Omega(r_0)$.
3. $\|\partial e(p)/\partial p\|$ and $\|f(e,p)\|$ are bounded for bounded $\|e\|$.
4. $f(e,p)$ is continuous in e and uniformly continuous in p.

Assumption 3 indicates that $e = 0$ is asymptotically stable and $\Omega(r_0)$ is the attraction region according to the invariance principle. The attraction region may be the full space or a sufficiently small neighborhood of $e = 0$. The synchronization system is globally stable if the attraction region is the entire space; otherwise, the system is locally stable.

The underlying graphs are assumed to be bidirectional and connected. If the graph is not bidirectional, the control laws may still work, but they may not be gradient control laws. For the sake of simplicity, suppose the weight for each edge to be one and let $m = |\mathcal{E}|/2$ denote the number of undirected edges.

2.3.2 Formation Control

This section proposed a modified gradient control law in (57) to cope with the nonholonomic constraint such that the velocity direction of each agent must align with its heading vector.

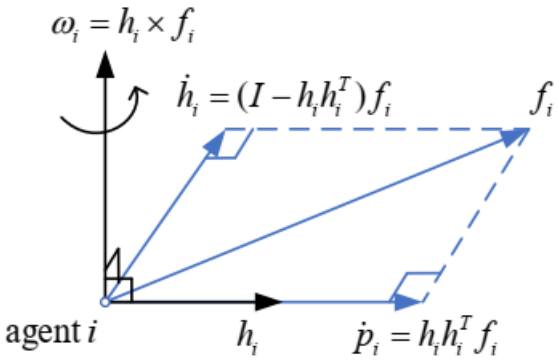

Figure 2 Illustration of the modified gradient control law in (60)

a) Modified Gradient Control Law

The proposed modified gradient control law is

$$\begin{aligned} \dot{p}_i &= h_i h_i^T f_i \\ \dot{h}_i &= \omega_i \times h_i, \qquad i \in \mathcal{V} \end{aligned} \tag{59}$$

where $\omega_i \in \mathbb{R}$ is the angular velocity to be designed, $h_i(t) \in \mathbb{R}^d$ is the unit-length heading vector of agent i, and $\times$ symbolizes the cross product. In (59), because $h_i h_i^T$ is an orthogonal projection matrix, the velocity $\dot{p}_i$ is the orthogonal projection of f_i onto h_i. Therefore, the velocity is aligned with the heading vector h_i and the nonholonomic constraint is satisfied. Because $\omega_i \times h_i$ is always orthogonal to h_i, the magnitude of h_i is invariant. To guarantee the entire multiagent system stays convergent in the sense that $V \to 0$, ω_i is appropriately designed as follows

$$\omega_i = h_i \times f_i \tag{60}$$

Equation (60) implies that ω_i make an attempt to rotate h_i to align with f_i (see Figure 2 for an illustration). Refer to $[\cdot]_\times$ as the skew-symmetric matrix of a vector. For any $x=[x_1,x_2,x_3]^T\in\mathbb{R}^3$

$$[x]_\times := \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \tag{61}$$

Then, we have $x\times y=[x]_\times y$ for any $x,y\in\mathbb{R}^3$. Substituting (60) into (59) gives

$$\dot{h}_i = -[h_i]_\times \omega_i = -[h_i]_\times^2 f_i = (I-h_ih_i^T) f_i$$

where the last equability follows from the fact that $-[x]_\times^2 = I - xx^T$ for any unit vector $x\in\mathbb{R}^3$. Then, the modified gradient control law is achieved

$$\begin{aligned} \dot{p}_i &= h_ih_i^T f_i \\ \dot{h}_i &= (I-h_ih_i^T) f_i, \qquad i\in\mathcal{V} \end{aligned} \tag{62}$$

It is worth noting that $I-h_ih_i^T$ is an orthogonal projection matrix that projects any vector onto the orthogonal complement of h_i. Although derived in $\mathbb{R}^3$, control law (62) is also valid in $\mathbb{R}^2$ because the case of $\mathbb{R}^2$ can be viewed as a special case of $\mathbb{R}^3$ by treating the plane spanned by h_i and f_i as the $x-y$ plane in $\mathbb{R}^3$.

Theorem 2: Under Assumption 3, the modified gradient synchronization control law (62) is convergent with the same attraction region as (57).

Proof: The error dynamics corresponding to (62) is known as $\dot{e}=(\partial e/\partial p)Mf$ where $M=\mathrm{diag}(h_1h_1^T,\dots h_nh_n^T)\in\mathbb{R}^{(dn)\times(dn)}$. The time derivative of V is

$$\dot{V} = -\sum_{i\in\mathcal{V}} f_i^T\dot{p}_i = -\sum_{i\in\mathcal{V}} f_i^T h_ih_i^T f_i \le 0 \tag{63}$$

It follows that $\Omega(V(e_0))\subseteq\Omega(r_0)$ is positively invariant for any $e_0\in\Omega(r_0)$. Let $\mathcal{M}=\{e:\dot{V}(e)=0\}$. Then, the system trajectory starting from any point in $\Omega(V(e_0))$ converges to the largest invariant set in $\mathcal{M}\cap\Omega(V(e_0))$ by the invariance principle. For any point in $\mathcal{M}$, $h_i^T f_i=0\forall i$, which indicates either $f_i=0$ for all i or $h_i\perp f_i$ but $f_i\neq 0$ for certain i. The first case follows that $e=0$ by condition 2) in Assumption 3. Therefore, the error converges to zero and the is proved. The second case is unfeasible. Assume $h_i\perp f_i$ but $f_i=0$. Then, $\dot{p}_i=h_ih_i^Tf_i=0\forall i$, which implies that all the agents are immobile. As a result, f_i is time invariant for all i. However, it follows from $h_i\perp f_i$ that $h_i=(I-h_ih_i^T)f_i=f_i\neq 0$. As a result, h_i is rotating. It is impossible to maintain $h_i\perp f_i$ if f_i is time invariant while h_i is rotating. Finally, the system trajectory will escape from $\mathcal{M}$. □

Theorem 2 reveals that if $\Omega(r_0)$ is the attraction region of the gradient system, then it remains an attraction region for the modified gradient system. As a result, if the original gradient control is globally (respectively, locally) stable, then the modified one is also

globally (respectively, locally) stable. The initial values of the heading vectors $\{h_i(0)\}_{i\in\mathcal{V}}$ do not affect the convergence. The final values $\{h_i(\infty)\}_{i\in\mathcal{V}}$ are not specified.

b) Application to Surface Vessel Models

Surface vessel models are new topic which can be considered in multiagent synchronization control. We apply (64) to analyze the specific control law for surface vessel agents moving in the plane. However, it is noticeable that (65) serves agents moving in both two and three dimensions. Let $\eta_i = [p_i^T, \psi_i]^T \in \mathbb{R}^3$ include the planar position $p_i = [x_i, y_i]^T \in \mathbb{R}^2$ and heading angle $\psi_i \in \mathbb{R}$ in the earth-fixed frame of agent i. The surface vessel model described the motion of agent i as follows

$$\begin{aligned} \dot{x}_i &= v_i \cos\psi_i \\ \dot{y}_i &= v_i \sin\psi_i \\ \dot{\psi}_i &= \omega_i \end{aligned} \tag{66}$$

where $v_i \in \mathbb{R}$ and $\omega_i \in \mathbb{R}$ are the linear and angular velocities. Control law for the surface vessel model is depicted as

$$\begin{aligned} v_i &= [\cos\psi_i, \sin\psi_i] f_i \\ \omega_i &= [-\sin\psi_i, \cos\psi_i] f_i \end{aligned} \tag{67}$$

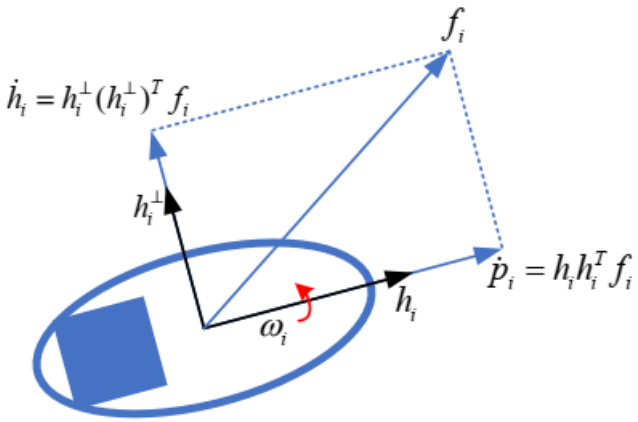

Figure 3 Geometric interpretation of the control law in (9)

Theorem 3: Under Assumption 3, control law (67) designed for the surface vessel in (66) is convergent with the same attraction region as (57).

The convergence of the control law is proved below.

Proof: Let $h_i = [\cos\psi_i, \sin\psi_i]^T$ and $h_i^{\perp} = [\sin\psi_i, \cos\psi_i]^T$. Note that $h_i \perp h_i^{\perp}$. Substituting control law (67) into the surface vessel yields $\dot{p}_i = h_i h_i^T f_i$ and $\dot{h}_i = h_i^{\perp}(h_i^{\perp})^T f_i$. Because $h_i^{\perp}(h_i^{\perp})^T = I - h_i h_i^T$, the closed-loop system has the same expression as (62). The convergence property then follows from Theorem 2. □

The geometric illustration of the control law in (67) is shown in Figure 3. The initial values of the heading angles do not affect the convergence. The final values are not specified. Equation (67) is used to derive a displacement-based formation control law for surface vessels.

Displacement-Based Formation Control

In displacement-based formation control, the objective is to guide the agents from some initial states to converge to a desired geometric shape defined by constant relative positions $\{p_i^* - p_j^*\}_{(i,j)\in\mathcal{E}}$. This formation control problem degenerates to the rendezvous problem when $p_i^* = p_j^*$. A Lyapunov function is depicted as

$$V = \frac{1}{4}\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}_i}\left\|(p_i - p_j) - (p_i^* - p_j^*)\right\|^2 \tag{68}$$

The target pattern is achieved if and only if $V = 0$ because the graph is bidirectional and connected. The displacement-based formation control law [24], [30] is described as follows

$$\dot{p}_i = f_i = \sum_{j\in\mathcal{N}_i}\left[(p_i - p_j) - (p_i^* - p_j^*)\right] \tag{69}$$

Consider any oriented graph and define the error state as $e_k = (p_i - p_j) - (p_i^* - p_j^*)$ with $k = 1,\ldots m$ and $e = (\mathcal{L}\otimes I)(p - p^*)$. Then, $V(e) = 1/2\sum_{i=1}^{m}\|e_k\|^2$, $\partial e/\partial p = \mathcal{L}\otimes I$ is constant, f is continuous in e, and $\|f\|$ is bounded when $\|e\|$ is bounded. Since $V(e) = 1/2(p - p^*)^T(\mathcal{L}\otimes I)(p - p^*)$ and $\dot{p} = f = -(\mathcal{L}\otimes I)(p - p^*)$ we have $f = 0 \Leftrightarrow V = 0 \Leftrightarrow e = 0$ and the attraction region $\Omega(r_0)$ is the entire space $\mathbb{R}^{dm}$. Therefore, all the conditions in Assumption 1 are satisfied.

Substituting f_i into (67) yields

$$\begin{aligned} v_i &= [\cos\theta_i, \sin\theta_i]\textstyle\sum_{j\in\mathcal{N}_i}(p_j - p_i - p_j^* + p_i^*) \\ &= [\cos\theta_i, \sin\theta_i]\left(-(\mathcal{L}\otimes I)(p - p^*)\right) \\ \omega_i &= [-\sin\theta_i, \cos\theta_i]\textstyle\sum_{j\in\mathcal{N}_i}(p_j - p_i - p_j^* + p_i^*) \\ &= [-\sin\theta_i, \cos\theta_i]\left(-(\mathcal{L}\otimes I)(p - p^*)\right) \end{aligned} \tag{70}$$

The agents will move in a square formation if an appropriate vector $a^* = \left[0, 0, d, 0, d, -d, 0, -d\right]^T$ is added to

$$\begin{aligned} v_i &= [\cos\theta_i, \sin\theta_i]\textstyle\sum_{j\in\mathcal{N}_i}(p_j - p_i - p_j^* + p_i^*) \\ &= [\cos\theta_i, \sin\theta_i]\left(-(\mathcal{L}\otimes I)(p - p^* - a^*)\right) \\ \omega_i &= [-\sin\theta_i, \cos\theta_i]\textstyle\sum_{j\in\mathcal{N}_i}(p_j - p_i - p_j^* + p_i^*) \\ &= [-\sin\theta_i, \cos\theta_i]\left(-(\mathcal{L}\otimes I)(p - p^* - a^*)\right) \end{aligned} \tag{71}$$

where a^* is the length of the square.

2.4 OVERALL CONTROL SYSTEM OF SURFACE VESSELS

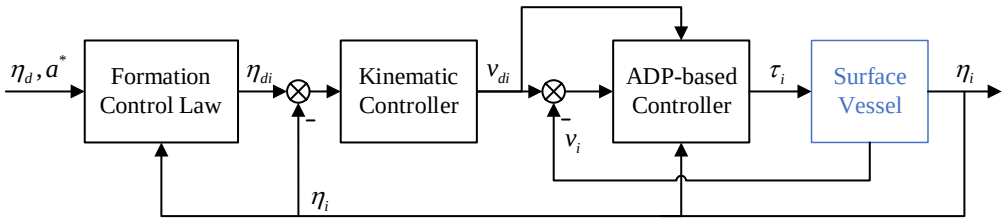

Figure 4 Overall control scheme of surface vessel formation

Figure 4 describes the overall control scheme of surface vessel formation. Generally, two control loops are depicted in this paper but three control loops can be analyzed as well.

Inputs of the system are reference trajectory of virtual leader η_d and the formation which is refered as the displacement of each SV relative to virtual leader a^*.

The outer control loop with formation control law produces the desired trajectory for each inner control loops. By this way, the multiagent system of surface vessels would maintain a formation while moving.

The inner control structure using kinematic controller and then ADP-based controller guarantees the desired trajectory tracking produced by the outer loop control.

2.5 SIMULATION RESULTS

In this section, a simulation example are provided to demonstrate the effectiveness of the developed procedures. The objective is a predefined formation of four SVs synchronously moving along desired trajectories. The parameters of each surface vessel are chosen as in [20] with the following inertia, Coriolis

$$M = \begin{bmatrix} 20 & 0 & 0 \\ 0 & 19 & 0.72 \\ 0 & 0.72 & 2.7 \end{bmatrix}$$

$$C(v) = \begin{bmatrix} 0 & 0 & -19v_y - 0.72v_z \\ 0 & 0 & 20v_x \\ 19v_y + 0.72v_z & -20v_x & 0 \end{bmatrix}$$

$$D(v) = \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & d_{23} \\ 0 & d_{32} & d_{33} \end{bmatrix}$$

$$d_{11} = 0.72 + 1.3|v_x| + 5.8v_x^2$$

$$d_{22} = 0.86 + 36|v_y| + 3|v_z|$$

$$d_{23} = -0.1 - 2|v_y| + 2|v_z|$$

$$d_{32} = -0.1 - 5|v_y| + 3|v_z|$$

$$d_{33} = 6 + 4|v_y| + 4|v_z|$$

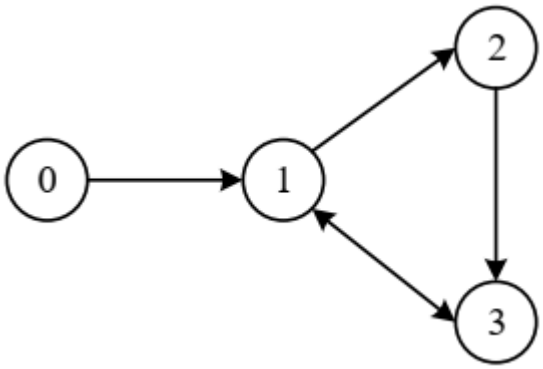

Figure 5 Communication graph of four agents

Using the graph definition in Session 2.3.1 the cooperative configuration of the multiple robots is presented by the distributed communication graph as Figure 5. The graph Laplacian matrix is

$$\mathcal{L} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 2 & 0 & -1 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & -1 & 2 \end{bmatrix}$$

Due to considering only on the surface, the potential vector $g(\eta)$ can be assumed to be 0. The desired trajectory is a circle which is described by $\eta_d(t) = [12\sin(0.2t), -12\cos(0.2t), 0.2t]^T$. The parameters of proposed controller for each agent consist of kinematic control law, feedforward, ARL algorithm being chosen as

$$\beta_\eta = 2 \quad k_c = 10 \quad k_{a1} = 10 \quad k_{a2} = 20$$
$$\upsilon = 0.01 \quad \varphi_0 = 20 \quad \varphi_1 = 12 \quad Q = I_3 \quad R = 1.$$

For the training of Actor-Critic architecture to achieve ARL based optimal control, the dual NNs are designed with 12 nodes in Actor and Critic part of each agent. The smooth activation function $\Psi(X) \in \mathbb{R}^{12}$ is chosen as

$$\Psi(X) = [X_1^2, X_1X_2, X_1X_3, X_2^2, X_2X_3, X_3^2,$$
$$X_1^2X_7^2, X_2^2X_8^2, X_3^2X_9^2, X_1^2X_4^2, X_2^2X_5^2, X_3^2X_6^2]^T$$

The updating laws are implemented based on (26), (27) for critic NN and (29) for actor NN, respectively.

Initial states of four agents are as follows

$$\eta_1(0) = [0,0,0]^T$$
$$\eta_2(0) = [20,0,0]^T$$
$$\eta_3(0) = [10,0,0]^T$$
$$\eta_4(0) = [\text{-}10,0,0]^T$$
$$v_i(0) = [0,0,0]^T, \qquad i = 1,2,3,4$$

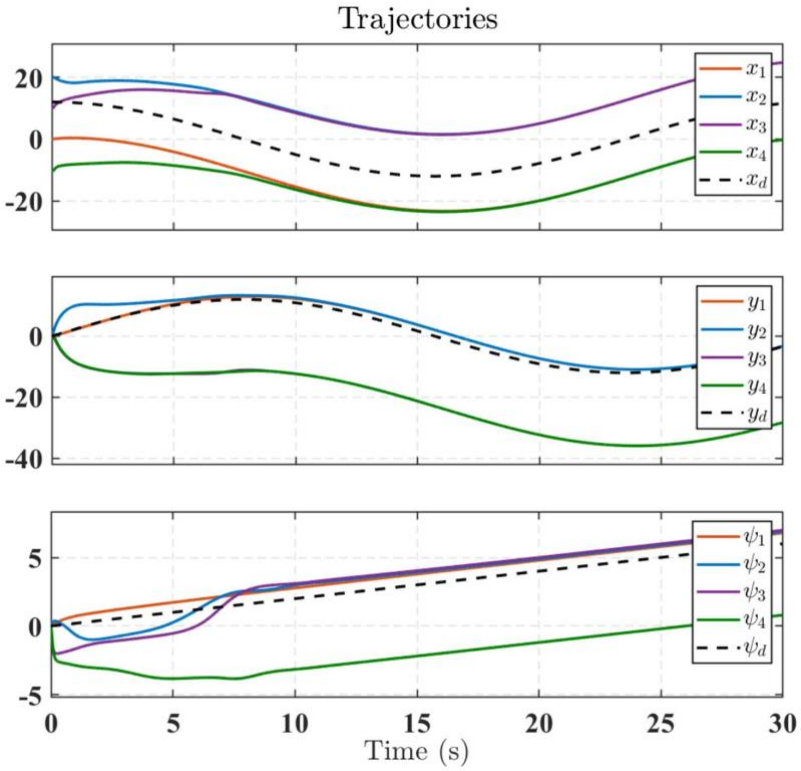

Figure 6 Tracking trajectories of four agents

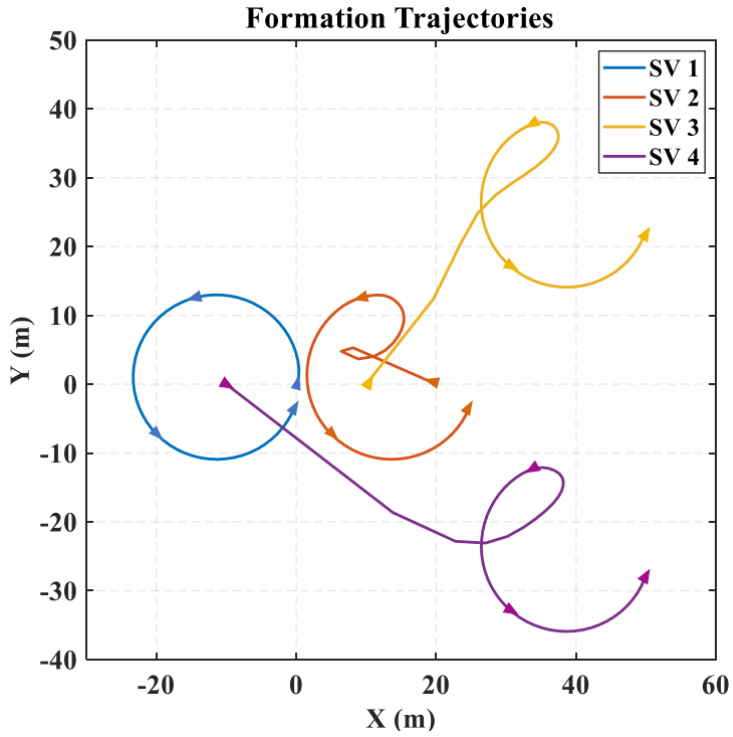

Figure 7 Formation illustration of four agents

The first simulation establishes a formation as depicted in Figure 5 by setting $a^* = [-25, 0, 0, 0, 25, 25, 25, -25]^T$. Figure 6 and Figure 7 shows that the synchronised formations in x_i and y_i, $i = 1, 2, 3, 4$, are maintained. The arrows on the trajectories in Figure 7 indicate the states at $t = 0, 10, 20, 30(s)$ respectively.

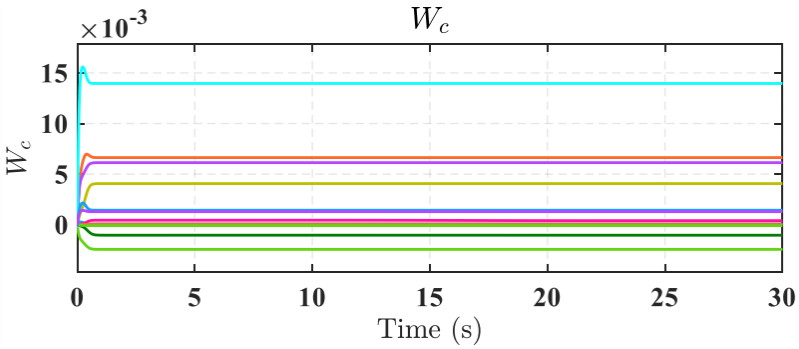

Figure 8 The convergences of critic weights

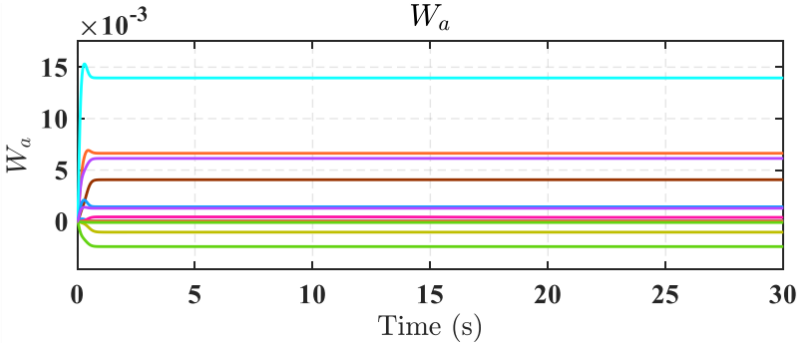

Figure 9 The convergences of actor weights

The trained weights in Critic and Actor part of the first agent is shown in Figure 8 and Figure 9 respectively. It is clear that all weights rapidly converge to the optimal solution. The other agents behave similarly.

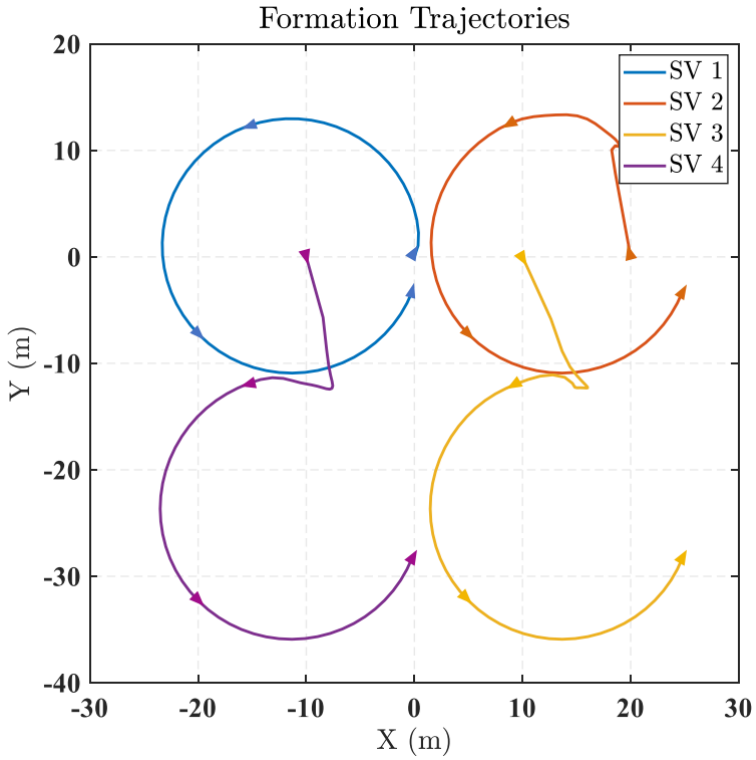

Figure 10 Second formation illustration of four agents

In the second simulation, a square formation by $a^* = [0, 0, 25, 0, 25, -25, 0, -25]^T$, shown in X is well-maintained.

The proposed formation controller applied on the built-in multiple SVs model reaffirms the correctness of the designed controller. It is noteworthy that the proposed ARL control law has an advantage in computation compared with [20, 33]. Through two simulation scenarios, the controller brought the ship system back to the setting formation.

III. CONCLUSION

In this article, a novel ARL based trajectory tracking cascade control design has been proposed for uncertain surface vessel systems. Research on a system of four SV objects and propose a method to control the formation, solving the problem of the distribution of agents in space. The unification of optimality and UUB stability of the closed system is proven by appropriate Lyapunov function candidate. The simulation further demonstrated the effectiveness of the proposed ARL based control scheme. The result is that the actual trajectory of the object has adhered quite well to the pre-set orbital and the agents are distributed according to the formation placed in space.

IV. REFERENCES

1. Park, Bong Seok and Kwon, Ji-Wook and Kim, Hongkeun, "Neural network-based output feedback control for reference tracking of underactuated surface vessels," Automatica, vol. 77, pp. 353–359, 2017.

2. Wang, Ning and Su, Shun-Feng and Pan, Xinxiang and Yu, Xiang and Xie, Guangming, "Yaw-guided trajectory tracking control of an asymmetric underactuated surface vehicle," IEEE Transactions on Industrial Informatics, vol. 16, no. 6, pp. 3502–3513, 2018.

3. Wang, Ning and Xie, Guangming and Pan, Xinxiang and Su, Shun-Feng, "Full-state regulation control of asymmetric underactuated surface vehicles," IEEE Transactions on Industrial Electronics, vol. 66, no. 11, pp. 8741–8750, 2019.

4. Li, Jia-Wang, "Robust adaptive control of underactuated ships with input saturation," International Journal of Control, pp. 1–10, 2019.

5. Qin, Hongde and Li, Chengpeng and Sun, Yanchao and Li, Xiaojia and Du, Yutong and Deng, Zhongchao, "Finitetime trajectory tracking control of unmanned surface vessel with error constraints and input saturations," Journal of the Franklin Institute, in Press.

6. Zhang, Jingqi and Yu, Shuanghe and Yan, Yan "Fixedtime output feedback trajectory tracking control of marine surface vessels subject to unknown external disturbances and uncertainties," ISA transactions, vol. 93, pp. 145–155, 2019.

7. Zhang, Jingqi and Yu, Shuanghe and Yan, Yan, "Fixedtime velocity-free sliding mode tracking control for marine surface vessels with uncertainties and unknown actuator faults," Ocean Engineering, vol. 201, in Press.

8. Van, Mien, "An enhanced tracking control of marine surface vessels based on adaptive integral sliding mode control and disturbance observer," ISA transactions, vol. 90, pp. 30–40, 2019.

9. Van, Mien, "Adaptive neural integral sliding-mode control for tracking control of fully actuated uncertain surface vessels," International Journal of Robust and Nonlinear Control, vol. 29, no. 5, pp. 1537–1557, 2019.

10. Wang, Ning and Karimi, Hamid Reza and Li, Hongyi and Su, Shun-Feng, "Accurate trajectory tracking of disturbed surface vehicles: A finite-time control approach," IEEE/ASME Transactions on Mechatronics, vol. 24, no. 3, pp. 1064–1074, 2019.

11. Xie, Wenjing and Ma, Baoli and Huang, Wei and Zhao, Yixin, "Global trajectory tracking control of underactuated surface vessels with non-diagonal inertial and damping matrices," Nonlinear Dynamics, vol. 92, no. 4, pp. 1481-1492, 2018.

12. Huang, Jiangshuai and Wen, Changyun and Wang, Wei and Jiang, Zhong-Ping, "Adaptive output feedback tracking control of a nonholonomic mobile robot," Automatica, vol. 50, no. 3, pp. 821–831, 2014.

13. Gao, Zhenyu and Guo, Ge, "Command-filtered fixed-time trajectory tracking control of surface vehicles based on a disturbance observer," International Journal of Robust and Nonlinear Control, vol. 29, no. 13, pp. 4348-4365, 2019.

14. Tuo, Yulong and Wang, Yuanhui and Yang, Simon X and Biglarbegian, Mohammad and Fu, Mingyu, "Robust adaptive dynamic surface control based on structural reliability for a turret-moored floating production storage and offloading vessel," International Journal of Control, Automation and Systems, vol. 16, no. 4, pp. 1648–1659, 2018.

15. Wu, Rui and Du, Jialu, “Adaptive robust course-tracking control of time-varying uncertain ships with disturbances,” International Journal of Control, Automation and Systems, vol. 17, no. 7, pp.1847–1855, 2019.

16. Xia, Guoqing and Sun, Chuang and Zhao, Bo and Xue, Jingjing, “Cooperative control of multiple dynamic positioning vessels with input saturation based on finite-time disturbance observer,” International Journal of Control, Automation and Systems, vol. 17, no. 2, pp. 370–379, 2019.

17. Zheng, Zewei and Huang, Yanting and Xie, Lihua and Zhu, Bing, “Adaptive trajectory tracking control of a fully actuated surface vessel with asymmetrically constrained input and output,” IEEE Transactions on Control Systems Technology, vol. 26, no. 5, pp.1851-1859, 2017.

18. Yang, Yang and Du, Jialu and Liu, Hongbo and Guo, Chen and Abraham, Ajith, “A trajectory tracking robust controller of surface vessels with disturbance uncertainties,” IEEE Transactions on Control Systems Technology, vol. 22, no. 4, pp. 1511-1518, 2013.

19. Qu, Yaohong and Xiao, Bing and Fu, Zhenzhou and Yuan, Dongli, “Trajectory exponential tracking control of unmanned surface ships with external disturbance and system uncertainties,” ISA transactions, vol. 78, pp. 47–55, 2018.

20. Wen, Guoxing and Ge, Shuzhi Sam and Chen, CL Philip and Tu, Fangwen and Wang, Shengnan “Adaptive tracking control of surface vessel using optimized backstepping technique,” IEEE transactions on cybernetics, vol. 49, no. 9, pp. 3420–3431, 2018.

21. Huang, Yuzhu and Wang, Ding and Liu, Derong, “Bounded robust control design for uncertain nonlinear systems using single-network adaptive dynamic programming,” Neurocomputing, vol. 266, pp. 128–140, 2017.

22. Bhasin, Shubhendu and Kamalapurkar, Rushikesh and Johnson, Marcus and Vamvoudakis, Kyriakos G and Lewis, Frank L and Dixon, Warren E, “A novel actor– critic–identifier architecture for approximate optimal control of uncertain nonlinear systems,” Automatica, vol. 49, no. 1, pp. 82–92, 2013.

23. Zhu, Yuanheng and Zhao, Dongbin and Li, Xiangjun, “Using reinforcement learning techniques to solve continuoustime non-linear optimal tracking problem without system dynamics,” IET Control Theory \& Applications, vol. 10, no. 12, pp. 1339–1347, 2016.

24. Guo, Xinxin and Yan, Weisheng and Cui, Rongxin, “Integral reinforcement learning-based adaptive NN control for continuous-time nonlinear MIMO systems with unknown control directions,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, Early Access, 2020.

25. Yang, Xiong and He, Haibo and Liu, Derong and Zhu, Yuanheng, “Adaptive dynamic programming for robust neural control of unknown continuous-time non-linear systems,” IET Control Theory \& Applications, vol. 11, no. 14, pp. 2307–2316, 2017.

26. Dornheim, Johannes and Link, Norbert and Gumbsch, Peter, “Model-free adaptive optimal control of episodic fixed-horizon manufacturing processes using reinforcement learning,” International Journal of Control, Automation and Systems, vol. 18, no. 6, pp. 1593–1604, 2020.

27. Guo, Linyuan and Rizvi, Syed Ali Asad and Lin, Zongli, “Optimal control of a two-wheeled self-balancing robot by reinforcement learning,” International Journal of Robust and Nonlinear Control, Early Acess, 2020.

28. Lv, Yongfeng and Ren, Xuemei and Hu, Shuangyi and Xu, Hao, "Approximate Optimal Stabilization Control of Servo Mechanisms based on Reinforcement Learning Scheme," International Journal of Control, Automation and Systems, vol. 17, no. 10, pp. 2655–2665, 2019.

29. Na, Jing and Lv, Yongfeng and Zhang, Kaiqiang and Zhao, Jun, "Adaptive Identifier-Critic-Based Optimal Tracking Control for Nonlinear Systems With Experimental Validation," IEEE Transactions on Systems, Man, and Cybernetics: Systems, Early Access, 2020.

30. Na, Jing and Lv, Yongfeng and Zhang, Kaiqiang and Zhao, Jun, "Adaptive Identifier-Critic-Based Optimal Tracking Control for Nonlinear Systems With Experimental Validation," IEEE Transactions on Systems, Man, and Cybernetics: Systems, Early Access, 2020.

31. Yang, Xiong and Liu, Derong and Wang, Ding, "Reinforcement learning for adaptive optimal control of unknown continuous-time nonlinear systems with input constraints," International Journal of Control, vol. 87, no. 3, pp. 553–566, 2014.

32. Sun, Tao and Sun, Xi-Ming, "An Adaptive Dynamic Programming Scheme for Nonlinear Optimal Control with Unknown Dynamics and Its Application to Turbofan Engines," IEEE Transactions on Industrial Informatics, Early Access, 2020.

33. Sun, Tao and Sun, Xi-Ming, "An Adaptive Dynamic Programming Scheme for Nonlinear Optimal Control with Unknown Dynamics and Its Application to Turbofan Engines," IEEE Transactions on Industrial Informatics, Early Access, 2020.

34. Dierks, T. and Jagannathan, S. "Neural network output feedback control of robot formations," IEEE Trans, Systems, Man, and Cybernetics, Part B: Cybernetics, 40(2), pp. 383–399, 2010.

35. Khoo, S., Xie, L. and Man, Z. "Robust finite-time consensus tracking algorithm for multi-robot systems," IEEE/ASME Transactions on Mechatronics, 14(2), pp. 219–228, 2009.

36. Dierks, T., Brenner, B. and Jagannathan, S. "Neural Network-Based Optimal Control of Mobile Robot Formations With Reduced Information Exchange," IEEE Transactions on Control Systems Technology, 21(4), pp. 1407– 1415, 2013.

37. Movric, K.H. and Lewis, F.L. "Cooperative optimal control for multi-agent systems on directed graph topologies," IEEE Transactions on Automatic Control, 59(3), pp. 769–774, 2014.

38. Dong, W. "Tracking control of multiple-wheeled mobile robots with limited information of a desired trajectory," IEEE Transactions on Robotics, 28(1), pp. 262–268, 2012.

39. Liu, T.F. and Jiang, Z.P. "Distributed output-feedback control of nonlinear multi-agent systems," IEEE Transactions on Automatic Control, 58(11), pp. 2912– 2917, 2013.

40. Peng, Z., Wang, D., Zhang, H. and Sun, G. "Distributed neural network control for adaptive synchronization of uncertain dynamical multi-agent systems," IEEE Transactions on Neural Networks and Learning Systems, 25(8), pp. 1508-1519, 2014.

41. Vamvoudakis, K.G., Lewis, F.L. and Hudas, G.R. "Multi-agent differential graphical games: Online adaptive learning solution for synchronization with optimality," Automatica, 48, pp. 1598–1611, 2012.

42. Zhang, H. and Lewis, F.L. "Optimal design for synchronization of cooperative systems: state feedback, observer and output feedback," IEEE Transactions on Automatic Control, 56(8), pp.1948-1952, 2011.

43. Zhang, H. and Lewis, F.L. "Adaptive cooperative tracking control of higher-order nonlinear systems with unknown dynamics," Automatica, 48(7), pp. 1432–1439, 2012.

44. Das, A. and Lewis, F.L. "Cooperative adaptive control for synchronization of second-order systems with unknown nonlinearities," International Journal of Robust and Nonlinear Control, 21(13), pp.1509–1524, 2011.

45. Cheng, L., Hou, Z., Tan, M., Lin, Y. and Zhang, W. "Neural networkbased adaptive leader following control for multi-agent systems with uncertainties," IEEE Transactions on Neural Networks, 21(8), pp.1351–1358, 2010.

46. Sun, D., Wang, C., Shang, W., Feng, G . A synchronization approach to trajectory tracking of multiple mobile robots while maintaining time-varying formations. IEEE Transactions on Robotics, 25(5), 1074-1086, 2009.

47. Movric, K. H., Lewis, F. L. Cooperative optimal control for multi-agent systems on directed graph topologies. IEEE Transactions on Automatic Control, 59(3), 769-774, 2013.

48. Peng, Z., Wang, D., Zhang, H., Sun, G. Distributed neural network control for adaptive synchronization of uncertain dynamical multiagent systems. IEEE transactions on neural networks and learning systems, 25(8), 1508-1519, 2013.

49. Zhao, S., Dimarogonas, D. V., Sun, Z., Bauso, D. A general approach to coordination control of mobile agents with motion constraints. IEEE Transactions on Automatic Control, 63(5), 1509-1516, 2017.

50. Hu, X., Wei, X., Gong, Q., Gu, J. Adaptive synchronization of marine surface ships using disturbance rejection without leader velocity. ISA Transactions, 2020.

51. H. K. Khalil, Nonlinear Systems, 3rd ed. Englewood Cliffs, NJ, USA: Prentice Hall, 2002.

52. Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, An Invitation to 3D Vision. New York, NY, USA: Springer, 2004.

53. Yin, Zhao and He, Wei and Yang, Chenguang and Sun, Changyin, "Control design of a marine vessel system using reinforcement learning," Neurocomputing, vol. 311, pp. 353–362, 2018.