Advances in Artificial Intelligence, Big Data and Algorithms
G. Grigoras and P. Lorenz (Eds.)
© 2023 The Authors.
This article is published online with Open Access by IOS Press and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).
doi:10.3233/FAIA230811

Teleoperation Formation Control of AUVs with State and Input Delays: A Board Learning-Based Solution

Tianming GAO[a], Jing YAN[a,1], Xian YANG[a], and Xinping GUAN[b]
[a] *Institute of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China*
[b] *Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China*

Abstract. Formation control of autonomous underwater vehicles is regarded as a promising way in situ sensing and monitoring of marine activities. However, due to the harsh marine environment, the full autonomy is still unreachable to fulfill complex marine tasks. This letter develops a teleoperation formation control system toward human-on-the-loop for AUVs. A board learning (BL) based estimator is first designed to estimate the real-time states of master operator and slave AUVs, through which the BL-based formation controller is developed to steer AUVs to keep specific formation shape. Compared with the previous works, the BL-based estimator can capture the real-time states even with time delays, and meanwhile the BL-based formation controller can achieve bilateral teleoperation without model parameters. Simulation results are conducted to verify our solution.

Keywords. Teleoperation control, time delay, board learning system, AUV.

1. Introduction

The unique characteristics of underwater environment make it challenging to construct underwater teleoperation formation system, including 1) Long time delays: The propagation delay of underwater acoustic communication is five orders of magnitude higher than in radio frequency channels [1], and hence, the state and input delays cannot be ignored. 2) Unknown model parameters: Due to complex underwater environment, it is difficult to acquire the accurate dynamic models of AUVs [2]. With regards to the first challenge, refs. [3, 4] considered the delayed states as the desired values, through which the bounded conditions of the formation controller can be derived. The above design is feasible for low time delay, however it is not valid for long time delay since the delayed state can induce large synchronization error. To this end, Zhou et al. developed an estimator to capture real-time states under input and state delays, however the selection of estimator gain requires solving complex linear matrix-inequality (LMI) equation [5]. Meanwhile, some system model parameters need to be known. For the second challenge, the deep reinforcement learning (DRL) and adaptive dynamic programming (ADP) have been employed to develop model-free formation controllers for AUVs, e.g., [6, 7]. However, the training process in DRL consumes plenty of time due to the large number

[1] Corresponding Author: Jing YAN, Institute of Electrical Engineering, Yanshan University; E-mail: jyan@ysu.edu.cn.

of parameters in the hidden layer, while the selection of basis functions in ADP relies on lots of experiences to obtain an accurate approximation.

In this work, we first design a BL-based estimator to capture the real-time states. Based on this, a BL-based formation controller is developed for the underwater teleoperation system which has the merits of saved training time via flat network and autonomous approximation via incremental learning. Major contributions lie in three aspects: 1) BL-based state estimator can capture the real-time states without solving LMI equation; 2) BL-based formation controller can achieve model-free formation under time delay; 3) simulation results are conducted to verify its practical value.

2. Problem Formulation

Consider the formation teleoperation system with single master human operator and N slave AUVs. Denote $\boldsymbol{\eta}_{\mathrm{m}} = [x_{\mathrm{m}}, y_{\mathrm{m}}, z_{\mathrm{m}}, \varphi_{\mathrm{m}}]^{\mathrm{T}}$ and $\mathbf{v}_{\mathrm{m}} = [u_{\mathrm{m}}, v_{\mathrm{m}}, w_{\mathrm{m}}, r_{\mathrm{m}}]^{\mathrm{T}}$ as the state and velocity vectors of master human operator, respectively. $\boldsymbol{\eta}_i = [x_i, y_i, z_i, \varphi_i]^{\mathrm{T}}$ and $\mathbf{v}_i = [u_i, v_i, w_i, r_i]^{\mathrm{T}}$ are the state and velocity vectors of slave AUV $i \in \{1, 2, ..., N\}$, respectively. The elements in these vectors are the states in surge, sway, heave and yaw, respectively. Thereby, the dynamic model of the system can be described as

$$\begin{aligned} &\mathbf{M}_{\mathrm{m}}(\mathbf{v}_{\mathrm{m}})\dot{\mathbf{v}}_{\mathrm{m}} + \mathbf{C}_{\mathrm{m}}(\mathbf{v}_{\mathrm{m}})\mathbf{v}_{\mathrm{m}} + \mathbf{G}_{\mathrm{m}}(\boldsymbol{\eta}_{\mathrm{m}}) = \boldsymbol{\tau}_{\mathrm{m}} + \mathbf{J}_{\mathrm{h}}\mathbf{F}_h, \\ &\mathbf{M}_i\dot{\mathbf{v}}_i + \mathbf{C}_i(\mathbf{v}_i)\mathbf{v}_i + \mathbf{D}_i(\mathbf{v}_i)\mathbf{v}_i + \mathbf{G}_i(\boldsymbol{\eta}_i) = \boldsymbol{\tau}_i, \end{aligned} \tag{1}$$

where $\mathbf{M}_l$ is inertial matrix, $\mathbf{C}_l$ is coriolis and centrifugal matrix, $\mathbf{G}_l$ is gravity matrix. $\boldsymbol{\tau}_l$ is input torque for $l \in \{m, i\}$. Besides, $\mathbf{J}_{\mathrm{h}}$ is the transformation matrix of human operator, $\mathbf{F}_h$ is the human-operator force, and $\mathbf{D}_i$ is the hydrodynamic damping matrix. After that, the system kinematic model can be expressed as

$$\dot{\boldsymbol{\eta}}_{\mathrm{m}} = \mathbf{J}_{\mathrm{m}}\mathbf{v}_{\mathrm{m}}, \quad \dot{\boldsymbol{\eta}}_i = \mathbf{J}_i\mathbf{v}_i, \tag{2}$$

where $\mathbf{J}_{\mathrm{m}}$ and $\mathbf{J}_i$ denote the transformation matrices.

In order to depict the topology relationship of slave AUVs, the undirected graph $\mathcal{G} \in (\mathcal{V}, \mathcal{E})$ is employed here. Denote $\mathcal{V} = \{1, 2, ..., N\}$ and $\mathcal{E} = \{(i, j) : i, j \in \mathcal{V}\}$ as the vertex set and edge set, respectively. (i, j) denotes AUV i can receive data from AUV j. Besides, the neighbor set of the i-th slave AUV is $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$. Hence, the adjacency matrix $\mathcal{A} = [a_{ij}] \in \mathcal{R}^{N \times N}$ is defined, where $a_{ij} = 1$ if $j \in \mathcal{N}_i$ else $a_{ij} = 0$.

For the slave AUVs, we consider the leader-follower formation, where AUV 1 acts as the leader and the other ones are followers. Then, the buoy is employed to relay the data from master human operator to leader AUV 1, including WiFi link from master human operator to buoy and acoustic link from buoy to leader AUV 1. Since the AUVs in slave site are not far from each other, the time delays among AUVs are ignored. Hence, the time delay from master human operator to slave AUVs can be denoted by d.

Particularly, denote $\mathbf{X}_l = [\boldsymbol{\eta}_l; \mathbf{v}_l]$ as the augmented state vector. r_{ij} is the relative position vector between AUVs i and j. Accordingly, the objective of underwater teleoperation formation control is to achieve: 1) *States estimation*: $\hat{\mathbf{X}}_{\mathrm{m}}(t) \to \mathbf{X}_{\mathrm{m}}(t)$ and $\hat{\mathbf{X}}_1(t) \to \mathbf{X}_1(t)$. 2) *Stability*: $\mathbf{X}_{\mathrm{m}} \to \hat{\mathbf{X}}_1, \mathbf{X}_1 \to \hat{\mathbf{X}}_{\mathrm{m}}$ and $\mathbf{X}_i \to \hat{\mathbf{X}}_j + r_{ij}$.

3. Design and Analysis

We first design BL-based state estimator to capture the real-time states. To this end, an admissible control policy $\tilde{\boldsymbol{\tau}}_1$ to steer slave AUV 1 from time t to T. Denote the generated states as $\mathcal{D}_1 = \{(\mathbf{X}_1(t), \ldots, \mathbf{X}_1(T)\}$. Simultaneously, these states and corresponding control inputs are relayed to the master human operator, which are denoted as $\mathcal{D}_2 = \{(\mathbf{X}_1(t-d), \ldots, \mathbf{X}_1(T-d)\}$ and $\mathcal{D}_3 = \{\tilde{\boldsymbol{\tau}}_1(t-d), \ldots, \tilde{\boldsymbol{\tau}}_1(T-d)\}$. Then, divide the above data via $(T-t)/\Delta t$ time windows with window length Δt. We select n windows of data as the labels, i.e., $\mathcal{D}_1^* = \{\mathbf{X}_{1,1}^*, \ldots, \mathbf{X}_{1,n}^*\}$, $\mathcal{D}_2^* = \{\mathbf{X}_{1,1}^{\#*}, \ldots, \mathbf{X}_{1,n}^{\#*}\}$ and $\mathcal{D}_3^* = \{\tilde{\boldsymbol{\tau}}_{1,1}^*, \ldots, \tilde{\boldsymbol{\tau}}_{1,n}^*\}$, where $1 \ll n < (T-t)/\Delta t$ is the random number of windows, $\mathbf{X}_{1,a}^*, \mathbf{X}_{1,a}^{\#*}$ and $\tilde{\boldsymbol{\tau}}_{1,a}^*$ denote the state data, delayed state data, and delayed control input data in the a-th $\in [1,n]$ window, respectively. In this way, the label set $\mathcal{Y} = \left\{\mathcal{D}_1^*\right\}$ and feature set $\mathcal{X} = \left\{\mathcal{D}_2^*, \mathcal{D}_3^*\right\}$ can be acquired for estimator training

Then, the BL network is employed to estimate the real-time states of slave AUV 1. To this end, the feature set is regarded as the input, and the estimated states of slave AUV 1 are considered as the output. The label set is to evaluate the accuracy of the estimated states. One constructs the p-th feature node $\mathcal{F}_p$ and the q-th enhancement node $\mathcal{E}_q$ as

$$\mathcal{F}_p = \phi_p(\mathcal{X}\boldsymbol{\omega}_p + \boldsymbol{\beta}_p), p \in \{1,2,\ldots,K_1\}, \mathcal{E}_q = \zeta_q(\mathcal{F}^{K_1}\boldsymbol{\omega}_q^{\#} + \boldsymbol{\beta}_q^{\#}), q \in \{1,2,\ldots,K_2\}, \quad (3)$$

where ϕ_p is the feature mapping function. ζ_q is the activation function. $\boldsymbol{\omega}_p$ and $\boldsymbol{\beta}_p$ are weight and bias of feature nodes, respectively. $\boldsymbol{\omega}_q^{\#}$ and $\boldsymbol{\beta}_q^{\#}$ are the weight and bias of enhancement nodes, respectively. Denote the stacked feature nodes and enhancement nodes as $\mathcal{F}^{K_1} = [\mathcal{F}_1, \ldots, \mathcal{F}_{K_1}]$ and $\mathcal{E}^{K_2} = [\mathcal{E}_1, \ldots, \mathcal{E}_{K_2}]$, respectively.

From (3), the *initialization*, *network reconfiguration*, and *real-time state estimation* are conducted to acquire the weight matrix $\mathbf{W}$ of the network, such that the estimated states of slave AUV 1 can be calculated since $\mathcal{O} = \mathbf{AW}$, where $\mathcal{O}$ and $\mathbf{A}$ are the state output and activation function for the estimator, respectively.

1) *Initialization.* Denote $\mathbf{A}_0 = \left[\mathcal{F}^{K_1} \,|\, \mathcal{E}^{K_2}\right]$ and $\mathbf{W}_0 = ((\mathbf{A}_0)^{\mathrm{T}}\mathbf{A}_0 + \epsilon\mathbf{I})^{-1}(\mathbf{A}_0)^{\mathrm{T}}\mathcal{Y}$ as the initial activation function and weight of network, respectively. Randomly generate the weight and bias. One can get the initial output of the estimator by $\mathcal{O}_0 = \mathbf{A}_0\mathbf{W}_0$.

2) *Network reconfiguration.* Add enhancement node to reduce the estimation error. Denote the b-th added enhancement nodes as $\mathcal{P}_b = \zeta_{\#b}(\mathcal{F}^{K_1}\boldsymbol{\omega}_{\#b} + \boldsymbol{\beta}_{\#b}), b \in \{1,2,\ldots,K_3\}$, where $\zeta_{\#b}$ is the activation function. $\boldsymbol{\omega}_{\#b}$ and $\boldsymbol{\beta}_{\#b}$ are the random weight and bias for the added enhancement node, respectively. Stack all the added enhancement nodes as

$\mathcal{P}^{K_3} = [\mathcal{P}_1, \ldots, \mathcal{P}^{K_3}]$. The network parameters can be updated as

$$\mathbf{A}_{\varrho+1} = \left[\mathbf{A}_{\varrho} \mid \mathcal{P}^{K_3}\right], \mathbf{W}_{\varrho+1} = \left[\mathbf{W}_{\varrho} - \mathbf{H}\mathbf{L}^{\mathrm{T}}\mathcal{Y}; \mathbf{L}^{\mathrm{T}}\mathcal{Y}\right], \mathbf{H} = \mathbf{A}_{\varrho}^{+}\mathcal{P}^{K_3}, \mathbf{N} = \mathcal{P}^{K_3} - \mathbf{A}_{\varrho}\mathbf{H},$$
$$\text{If } \mathbf{N} \neq \mathbf{0},\ \mathbf{L} = (\mathbf{N}^{+})^{\mathrm{T}} \text{ else } \mathbf{L} = ((1 + \mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{L}^{\mathrm{T}}(\mathbf{A}_{\varrho}^{+}))^{\mathrm{T}}. \tag{4}$$

3) *Real-time state estimation*. The termination condition of iteration satisfies $\left\|\mathcal{O}_{\varrho+1} - \mathcal{O}_{\varrho}\right\|_2^2 < \sigma$. After iterating, the estimated states are output by $\mathcal{O}_{\varrho+1} = \mathbf{A}_{\varrho+1}\mathbf{W}_{\varrho+1}$.

Note that the design of master human operator state estimator is the same as the above steps, which is omitted here.

After that, we employ the BL system to design a model-free teleoperation formation controller. For ease of analysis, the system dynamic model (1) can be rearranged as

$$\begin{aligned} \mathbf{X}_{\mathrm{m}}(k+1) &= f_{\mathrm{m}}(\mathbf{X}_{\mathrm{m}}(k)) + c_{\mathrm{m}}(\mathbf{X}_{\mathrm{m}}(k))\boldsymbol{\tau}_{\mathrm{m}}, \\ \mathbf{X}_i(k+1) &= f_i(\mathbf{X}_i(k)) + c_i(\mathbf{X}_i(k))\boldsymbol{\tau}_i, \end{aligned} \tag{5}$$

where k is the sampling time. Denote $\mathbf{e}_{1,\mathrm{m}}(k) = \mathbf{X}_{\mathrm{m}}(k) - \hat{\mathbf{X}}_1(k)$, $\mathbf{e}_{\mathrm{m},1} = \mathbf{X}_1(k) - \hat{\mathbf{X}}_{\mathrm{m}}(k)$ and $\mathbf{e}_{i,h} = \mathbf{X}_i(k) - \mathbf{X}_f(k) - r_{ih}$ as the tracking errors of the master human operator, slave AUV 1, and slave AUV $h \in \mathcal{V}\ (h \neq 1)$. Accordingly, define the cost function as

$$\begin{aligned} g_{\mathrm{m}} &= \mathbf{e}_{1,\mathrm{m}}^{\mathrm{T}}\mathbf{Q}_{1,\mathrm{m}}\mathbf{e}_{1,\mathrm{m}} + \boldsymbol{\tau}_{\mathrm{m}}^{\mathrm{T}}\mathbf{R}_{\mathrm{m}}\boldsymbol{\tau}_{\mathrm{m}}, g_1 = \mathbf{e}_{\mathrm{m},1}^{\mathrm{T}}\mathbf{Q}_{\mathrm{m},1}\mathbf{e}_{\mathrm{m},1} + \boldsymbol{\tau}_1^{\mathrm{T}}\mathbf{R}_1\boldsymbol{\tau}_1, \\ g_1(\mathbf{X}_h, \boldsymbol{\tau}_h) &= \sum\nolimits_{i\in\mathcal{N}_h} \mathbf{e}_{i,h}^{\mathrm{T}}\mathbf{Q}_{i,h}\mathbf{e}_{i,h} + \boldsymbol{\tau}_h^{\mathrm{T}}\mathbf{R}_h\boldsymbol{\tau}_h, \end{aligned} \tag{6}$$

where $\mathbf{Q}_{1,\mathrm{m}}$, $\mathbf{Q}_{\mathrm{m},1}$, $\mathbf{Q}_{i,h}$, $\mathbf{R}_{\mathrm{m}}$, $\mathbf{R}_1$ and $\mathbf{R}_h$ are positive definite matrices. Accordingly, the teleoperation formation control optimal problem can be described as

$$\boldsymbol{\tau}_s^*(k) = \arg\min\nolimits_{\boldsymbol{\tau}_s} \left\{g_s(\mathbf{X}_s(k), \boldsymbol{\tau}_s(k)) + J(\mathbf{X}_s(k+1))\right\}, s \in \{m, 1, h\}. \tag{7}$$

To solve optimization problem (7), the *model preprocessing*, *autonomous approximation* and *network reconfiguration* are conducted to obtain the optimal policy.

1) *Model preprocessing*. We introduce the admissible policy $\tilde{\boldsymbol{\tau}}_s$, such that (5) can be rearranged in the ρ-th iteration steps

$$\mathbf{X}_s(k+1) = f_s(\mathbf{X}_s(k)) + c_s(\mathbf{X}_s(k))(\tilde{\boldsymbol{\tau}}_s(k) - \boldsymbol{\tau}_s^{\rho}(k)) + c_s(\mathbf{X}_s(k))\boldsymbol{\tau}_s^{\rho}(k). \tag{8}$$

Based on this, one has the following Bellman error

$$J^{\rho}(\mathbf{X}_s(k+1)) - J^{\rho}(\mathbf{X}_s(k)) = -g_s - 2\boldsymbol{\tau}_s^{\rho+1}(k)\mathbf{R}_s(\tilde{\boldsymbol{\tau}}_s(k) - \boldsymbol{\tau}_s^{\rho}(k)). \tag{9}$$

Next, we adopt ADP with BL to design critic-actor networks, through which $J^{\rho}(\mathbf{X}_s(k))$ and $\boldsymbol{\tau}_s^{\rho+1}(k)$ can be approximated without choosing basis function.

2) *Autonomous approximation.* The tracking errors are employed to construct the ϑ -th feature node $\mathcal{F}_{s\vartheta}$ and the μ -th enhancement node $\mathcal{E}_{s\mu}$. Then, stack all the feature nodes and enhancement nodes as $\mathcal{F}_s^{K_4} = [\mathcal{F}_{s1},\ldots,\mathcal{F}_{sK_4}]$ and $\mathcal{E}_s^{K_5} = [\mathcal{E}_{s1},\ldots,\mathcal{E}_{sK_5}]$, respectively. Based on this, the unknown items can be approximated by

$$\hat{J}^{\rho}(\mathbf{X}_s(k)) = \mathbf{W}_{\mathrm{c},s}^{\mathrm{T}}\mathbf{A}_{\mathrm{c},s}(k),\ \hat{\boldsymbol{\tau}}_s^{\rho+1}(k) = \mathbf{W}_{\mathrm{a},s}^{\mathrm{T}}\mathbf{A}_{\mathrm{a},s}(k), \tag{10}$$

where $\mathbf{W}_{\mathrm{c},s}^{\mathrm{T}}$ and $\mathbf{A}_{\mathrm{c},s} = [\mathcal{F}_{\mathrm{c},s}^{K_4} \mid \mathcal{E}_{\mathrm{c},s}^{K_5}]$ are the weight vector and BL-based basis function for the critic network, respectively. $\mathbf{W}_{\mathrm{a},s}^{\mathrm{T}}$ and $\mathbf{A}_{\mathrm{a},s} = [\mathcal{F}_{\mathrm{a},s}^{K_4} \mid \mathcal{E}_{\mathrm{a},s}^{K_5}]$ represent the weight vector and BL-based basis function for the actor network, respectively. Hence, one has the temporal difference error

$$E(t) = (\Delta\mathbf{A}_{\mathrm{c},s}^{\mathrm{T}} \otimes I)\mathbf{W}_{\mathrm{c},s} + g(\mathbf{X}_s(t),\boldsymbol{\tau}_s^{\rho}(k)) + 2(\Delta\boldsymbol{\tau}_s^{\mathrm{T}}\mathbf{R}_s) \otimes \mathbf{A}_{\mathrm{a},s}^{\mathrm{T}}(k)\mathrm{vec}(\mathbf{W}_{\mathrm{a},s}), \tag{11}$$

where $\Delta\mathbf{A}_{\mathrm{c},s} = \mathbf{A}_{\mathrm{c},s}(k+1) - \mathbf{A}_{\mathrm{c},s}(k)$ and $\Delta\boldsymbol{\tau}_s = \tilde{\boldsymbol{\tau}}_s(k) - \boldsymbol{\tau}_s^{\rho}(k)$. Denote the weight as $\mathbf{W}_{\mathrm{all}}^{\#\varrho} = [\mathbf{W}_{\mathrm{c},s}^{\#\varrho};\mathbf{W}_{\mathrm{a},s}^{\#\varrho}]$. Then, the gradient descent method is adopted to update it.

3) *Network reconfiguration.* To improve approximation ability, one adds enhancement node $\mathcal{P}_s$ into critic-actor networks. Denote the stacked added enhancement nodes as $\mathcal{P}_s^{K_6} = [\mathcal{P}_{s1},\ldots,\mathcal{P}_{sK_6}]$. Similar to (4), update the weight matrices of networks.

To ensure the transparency, the feedback force $\tilde{\mathbf{F}}$ is perceived by human operator

$$\tilde{\mathbf{F}} = \varpi\left(\frac{1}{1+e^{-\|\boldsymbol{\eta}_{\mathrm{l}}-\boldsymbol{\eta}_{\mathrm{m}}\|}} - 0.5\right), \varpi > 0. \tag{12}$$

Theorem 1: Consider the collected label set $\mathcal{Y}$ and feature set $\mathcal{X}$, one constructs the additional reinforcement nodes in for incremental learning, then the estimated states are convergent as the weight matrix update, i.e., $\lim_{\varrho\to\infty}\|\mathcal{O}_{\varrho+1} - \mathcal{Y}\|_2^2 = 0$.

Proof. Define $U(\mathcal{X}) \triangleq \mathcal{Y}$ and $P(\mathcal{X}) \triangleq \mathcal{O}_0$. The initial output of estimator can be denoted as $P(\mathcal{X}) = f_{\mathcal{F}^{K_1}}(\mathcal{X}) + f_{\mathcal{E}^{K_2}}(\mathcal{X})$, where $f_{\mathcal{F}^{K_1}}(\mathcal{X})$ and $f_{\mathcal{E}^{K_2}}(\mathcal{X})$ are outputs of the feature node and enhancement node, respectively. Then the distance between $P(\mathcal{X})$ and $U(\mathcal{X})$ on the compact set $\Lambda \subset \mathbf{I}^{\varepsilon}$ can be represented as

$$\Lambda(P(\mathcal{X}),U(\mathcal{X})) = \sqrt{E\left[\int_{\Lambda}(P(\mathcal{X}) - U(\mathcal{X}))^2\right]d(\mathcal{X})} \tag{13}$$

where $\mathbf{I}^{\varepsilon} = [0;1]^{\varepsilon} \subset \mathcal{R}^{\varepsilon}$ is the standard hypercube. Since the feature mapping function is bounded, and $f_{\mathcal{F}^{K_1}}(\mathcal{X})$ is bounded and integrable. One can deduce that the resident function $f_e(\mathcal{X}) = U(\mathcal{X}) - f_{\mathcal{F}^{K_1}}(\mathcal{X})$ is bounded and integrable. Following this, one has a continuous function f_{λ} to make $\forall\kappa > 0$, i.e., $\Lambda(f_e(\mathcal{X}), f_{\lambda}) < \kappa/2$.

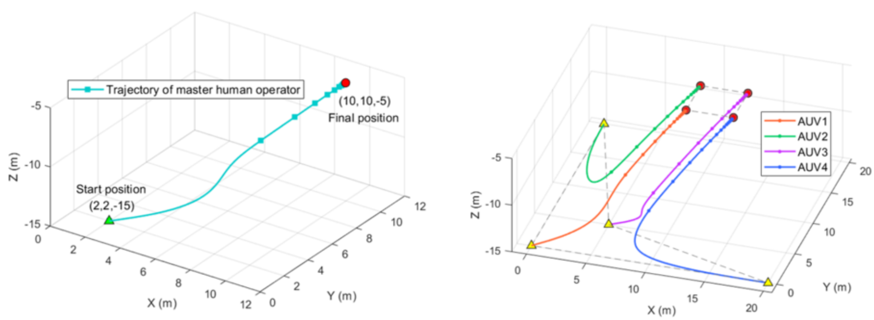
From (14), one can further deduce

$$\Lambda(P(\mathcal{X}),U(\mathcal{X})) \leq \Lambda(f_e(\mathcal{X}), f_\lambda) + \Lambda(f_{\mathcal{E}^{K_2}}(\mathcal{X}), f_\lambda) \leq \kappa, \tag{14}$$

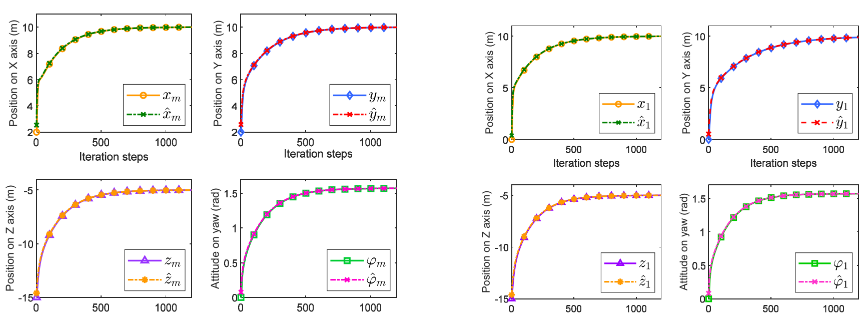
which means the initial output of estimator is convergent. Since the network architecture has not changed after ϱ -th incremental learning, one has $\lim_{\varrho\to\infty}\left\|\mathcal{O}_{\varrho+1}-\mathcal{Y}\right\|_2^2 = 0$.

4. Simulation and Experiment Results

In simulation studies, we consider the slave site consisting of four AUVs to perform formation task. The topology relationship of slave AUVs is $a_{12} = a_{14} = a_{32} = a_{34} = 1$. The parameters are set as $d = 0.5$, $\sigma = 0.25$, and $\varpi = 30$. The desired shape of slave AUVs is a square with side length of 4 meters. The trajectories of master human operator and slave AUVs are shown in Figs. 1(a)-(b). It is shown that the slave AUVs track the movement of master human operator and form the desired formation shape. In the above process, the real-time states of master human operator and slave AUV 1 captured by BL-based bilateral estimators are shown in Figs. 1(c)-(d). Clearly, the estimators capture the bilateral real-time states successfully under time delay. This verifies the effectiveness of BL-based real-time states estimator in this paper.



(a) Trajectory of master site. (b) Trajectory of slave AUVs.



(c) Estimation states of master site. (d) Estimation states of slave AUVs.

Figure 1. Simulation results for underwater teleoperation formation control

5. Conclusion

In this paper, the teleoperation formation control of AUVs under time delay has been investigated. The BL-based estimators have been designed to capture bilateral states, through which the slave AUVs can form formation by BL-based formation controller. Simulation results are given to verify the effectiveness of our solution. In the future, we consider the co-design of detection, communication and control of underwater teleoperation system. Moreover, more complex scenario will be considered.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants 62222314, 61973263, and 62033011; by the Youth Talent Program of Hebei under Grant BJ2020031; by the Distinguished Young Foundation of Hebei Province under Grant F2022203001; by the Excellent Youth Project for NSF of Hebei Province under Grant F2021203056; by the Central Guidance Local Foundation of Hebei Province under Grant 226Z3201G; and by the Three-Three-Three Foundation of Hebei Province under Grant C20221019.

References

[1] Z. Sun, H. Guo, and I. Akyildiz. High-data-rate long-range underwater communications via acoustic reconfigurable intelligent surfaces, *IEEE Commun. Mag.*, 2022 (60): 96-102.
[2] Z. Zhou, J. Liu, and J. Yu. A survey of underwater multi-robot systems, *IEEE/CAA J. Autom. Sinica*, 2022(9): 1-18.
[3] E. Rodriguez-Seda, J. Troy, C. Erignac, et al. Bilateral teleoperation of multiple mobile agents: coordinated motion and collision avoidance, *IEEE Trans. Control Syst. Technol.*, 2010(18): 984-992.
[4] R. Baranitha, R. Mohajerpoor, and R. Rakkiyappan. Bilateral teleoperation of single-master multislave systems with semi-markovian jump stochastic interval time-varying delayed communication channels, *IEEE Trans. Cybern.*, 2021(51): 247-257.
[5] B. Zhou, Q. Liu, F. Mazenc. Stabilization of linear systems with both input and state delays by observer-predictors, *Automatica*, 2017(83): 368-377.
[6] Z. Fang, D. Jiang, J. Huang, et al. Autonomous underwater vehicle formation control and obstacle avoidance using multi-agent generative adversarial imitation learning, *Ocean Eng.*, 2022(262): 1-13.
[7] G. Che and Z. Yu. Neural-network estimators based fault-tolerant tracking control for AUV via ADP with rudders faults and ocean current disturbance, *Neurocomputing*, 2020(411): 442-454.