

## Research Article

# Deep Learning-Based Trajectory Tracking Control for Unmanned Surface Vehicle

Wenli Sun<sup>1</sup> and Xu Gao<sup>2</sup> 

<sup>1</sup>Navigation College, Dalian Maritime University, Dalian 116026, China

<sup>2</sup>National Engineering Research Center of Maritime Navigation System, Dalian Maritime University, Dalian 116026, China

Correspondence should be addressed to Xu Gao; [gaoxu@hdnav.cn](mailto:gaoxu@hdnav.cn)

Received 6 August 2020; Revised 4 December 2020; Accepted 23 December 2020; Published 9 January 2021

Academic Editor: Giuseppe D'Aniello

Copyright © 2021 Wenli Sun and Xu Gao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Trajectory tracking control based on waypoint behavior is a promising way for unmanned surface vehicle (USV) to achieve autonomous navigation. This study is aimed at the guidance progress in the kinematics; the artificial intelligence method of deep learning is adopted to improve the trajectory tracking level of USV. First, two deep neural network (DNN) models are constructed to evaluate navigation effects and to estimate guidance law parameters in real time, respectively. We then pretrain the DNN using a Gaussian–Bernoulli restricted Boltzmann machine to further improve the accuracy of predicting navigation effect. Finally, two DNNs are connected in parallel with the control loop of USV to provide predictive supervision and auxiliary decision making for traditional control methods. This kind of parallel way conforms to the ship manipulation of habit. Furthermore, we develop a new application on the basis of Mission Oriented Operating Suite Interval Programming named “pDeepLearning.” It can predict the navigation effect online by DNN and adjust the guidance law parameters according to the effect level. The experimental results show that, compared with the original waypoint behavior of USV, the prediction model proposed in this study reduces the trajectory tracking error by 19.0% and increases the waypoint behavior effect level.

## 1. Introduction

Unmanned surface vehicles (USVs) are mainly preferred by missions that are characterized as dull, dangerous, or ill-suited for manned ships. In the future, they will be developed for ocean mapping, hydrographic and meteorological monitoring, maritime search and rescue, etc. Autonomous control is the core technology of USV navigation. It belongs to motion control technology, which includes set-point regulation control [1], path following control [2], and trajectory tracking control.

Trajectory tracking is defined as the control for actual track of vehicle so that the vehicle can track the position of Cartesian coordinate relative to time [3]. The optimal time-varying path for trajectory tracking is derived from the dynamic model of the vehicle and a predefined target [4]. The difficulty is the impact of uncertain sea environment. Trajectory tracking is a fundamental capability for USV to

perform missions such as automatic collision avoidance and cooperative formation. Therefore, we choose trajectory tracking control as the research issue.

This study follows the framework of the Guidance–Navigation–Control (GNC) system to solve the trajectory tracking problem of USV. The GNC system framework is a two-stage process consisting of guidance and control [4]. The guidance process refers to the transformation of a vehicle position to its heading and speed in the kinematics domain by means of guidance law. The control process refers to the transformation of heading to rudder angle and speed to throttle in the kinetics domain by means of control law. The essence of trajectory tracking control is to minimize the position error between the actual track and the reference track. Therefore, the performance of trajectory tracking is mostly dependent on the guidance process [5].

The object of this study is the guidance process of trajectory tracking control for USV. The modeling approach

and experimental platform are founded on Mission Oriented Operating Suite Interval Programming (MOOS-IvP) [6]. MOOS-IvP is a research platform for autonomous maritime vehicles, open-sourced by MIT. It conforms to the framework of GNC system and divides guidance and control processes. The waypoint behavior in MOOS-IvP models the process of trajectory tracking, which is a Line-of-Sight (LOS) guidance method. The feedback effect generated by control law, vehicle, and its interaction with the environment is regarded as a whole. Trajectory tracking control of USV is accomplished by adjusting the LOS guidance parameters. MOOS-IvP can be deployed in real ships, so our work also enables its application to real-world scenarios.

On the other hand, the dramatic technological evolution of deep learning has delivered new insights and approaches for the study of USV autonomous navigation. Deep neural network (DNN) recognizes and extracts the relationship between the combined features. It is suitable for uncertain sea environment and complex vehicle motion control process [7]. In this paper, a dual-DNN model is established, and back propagation method is used to train the USV by taking the navigation data samples under different parameters such as speed and steering angle in a simulated environment. The trained DNN model is capable of predicting the tracking effect and estimating better guidance law parameters, so as to improve the trajectory tracking control process of USV.

During the last decade, a large number of methods for trajectory tracking control have been developed, but in practice there are still many difficulties. Velagic et al. proposed an adaptive fuzzy controller [8] and built a ship dynamics model, a steering equipment model, and a wind-flow disturbance model. All models have been simplified, including reducing dimensions, adding constraints, and removing higher order interference terms. On the other hand, the model parameters were adjusted according to predefined 49 fuzzy rules. If the ship encountered conditions that did not appear in the rules, it would be difficult to control them precisely. Aguiar et al. presented a nonlinear control algorithm according to Lyapunov theory and proved the global convergence of the model without constraints [9]. Lv et al. proposed a hybrid cooperative signal energy control law, which dealt with the speed and course control of ship. However, they did not consider the uncertainty of resistance and disturbance [10]. Xia et al. developed a dynamic model and an adaptive controller to promote tracking performance and convergence speed, but they did not consider the parametric perturbations caused by power devices [11]. Huang et al. decomposed the trajectory tracking problem into guidance law and control law loops, which are easier to solve than direct control methods, but they did not take into account external disturbances from the marine environment [12]. The traditional methods adopted in these studies aimed to build complex models with many parameters, and most of them did not consider the impact brought by real environment and equipment.

In recent years, many studies have used artificial neural network (ANN) in the control issues of USV, especially in trajectory tracking control problems. ANN and data-driven

machine learning method have shown success in the analysis of uncertain and environmentally sensitive ship motion control problems [13, 14]. Cheng et al. used ANN as a replacement model for the traditional model to solve the problem of ship berthing [15]. Shuai et al. constructed two ANNs to extract features for controlling the ship propeller and rudder, respectively, to achieve automatic berthing under different environmental disturbances [16]. Zhang et al. proposed an adaptive robust ANN method for modeling uncertain ship dynamics and external influences and achieved good results in automatic ship berthing [17]. The ANN proposed by Wang et al. was used to control the course of USV and solve the problem of uncertainty in ship motion [18]. In the above literature works, the influence of sea environment has been considered, and the ANN method has been improved over traditional methods. However, since the ANN is shallow, it is difficult to learn the characteristic relationships of the parameters, so they selected a simple scenario of berthing or simplified the problem.

In recent years, DNN has demonstrated unprecedented ability in the field of traffic and control [19]. Kim et al. exploited DNN-based feedback controllers to compensate for the disturbance of curved road and reduced tracking error in lane keeping [20]. Xu et al. used DNN to learn the complex manipulation characteristics of USV based on the visual system [21]. Chen et al. proposed a DNN-based data-driven control method that greatly improved the capability and accuracy of control systems [22]. Deep belief network was proposed by Tan et al. to address the navigational safety of unmanned aerial vehicles [23]. It can be seen that DNN has shown stronger feature learning ability than shallow ANN [24]. However, it is not widely used in the trajectory tracking control problem of USV.

DNN has a gradient dispersion problem, which is generally solved by “pretraining and fine-tuning” method. Hinton et al. proposed a method of pretraining the restricted Boltzmann machine (RBM) in each layer, followed by fine-tuning the DNN [25]. Goudarzi et al. proposed two stacked RBMs for predicting short-term traffic flow by using the “pretraining and fine-tuning” method [26]. Zhao et al. developed a deep belief network consisting of several RBMs stacked, to reduce the risk of vehicle collision in snow and ice conditions on highways [27]. Pretraining of DNNs by using RBM is a common method [25]. However, RBM has been originally developed for binary vector modeling, where both visible and hidden layer variables are binary. In this study, the visible layer variables are types of continuous values, so binary RBM cannot be sufficiently used. Yamashita [28] gave a method for continuous value type vectors, i.e., Gaussian-Bernoulli restricted Boltzmann machine (GB-RBM), so we use GB-RBM for pretraining DNN to improve the performance, and make it applicable to our research.

In the implementation of intelligent navigation systems for USV, MOOS-IvP has been very popular in academic and industrial research fields nowadays, and it provides good support for autonomous navigation [29]. Firstly, MOOS-IvP provides a simulated experimental environment for USV. For example, Dong et al. used the MOOS-IvP platform to experiment with different test items for distributed remote

control of USVs [30]. Secondly, MOOS-IvP provides integrated interfaces for software development and algorithm implementation. For example, the instruction filter control module developed by Djapic et al. was integrated into MOOS-IvP [31]. In addition, a set of algorithms for generating waypoints developed by Benjamin et al. have been used for path planning in MOOS-IvP [32]. This study is inspired by them. On the one hand, MOOS-IvP is used to acquire data and perform experiments. On the other hand, the developed DNN model is integrated into MOOS-IvP. From the view of artificial intelligence computing, this is also an upgrade of MOOS-IvP platform.

In this study, a deep learning methodology is utilized to predict the parameters of the waypoint behavior in MOOS-IvP, and the DNN prediction model is implemented in parallel with the control loop to achieve the trajectory tracking of USV. The whole project is implemented in two stages: In the first stage, a classification model based on DNN was constructed to provide assistance and reference for maneuvering decisions of USV [7]. In the second stage, we regard the feedback effect generated by the control law, vehicle, and its interaction with the environment as a whole and connect DNN in parallel with the control loop of USV, so as to predict LOS guidance law parameters in real time during voyage.

To predict the LOS guidance law parameters accurately in the second stage, we add a pretraining process by using GB-RBM, which improves the accuracy of model classification to 89.9% with an increase of 5% over the previous stage.

On the basis of the above works, the “waypoint behavior effect evaluation model” and the “real-time LOS parameter valuation model” are constructed based on DNN, denoted as DNN-1 and DNN-2, respectively. They are connected in parallel with the trajectory tracking control loop of USV. In the process of voyage, DNN-1 is used to predict the effect of navigation at first, and then the LOS parameters are given by DNN-2 when the effect is not good. The new LOS parameters are configured to adjust the waypoint behavior of USV. In this way, it is not only introducing intelligent computing to the control loop but also maintaining the reliability of the traditional control process as far as possible. At the same time, it also takes into account the traditional habit of steering infrequently in the ship maneuvering. In addition, we develop a new MOOS-IvP application to perform the computing of DNN model, and establish an interface between the trained DNN and the MOOS-IvP platform.

The contribution herein mainly includes the following three aspects:

- (1) With the deep learning training method of “pre-training and fine-tuning” and the model of GB-RBM, the prediction accuracy of classification model is improved. GB-RBM can fit the numerical data to prevent the model falling into local optimum.
- (2) A new predictive-based trajectory tracking control model has been innovatively constructed. The model

consists of two DNNs, i.e., “DNN-1: waypoint behavior effect evaluation model” and “DNN-2: real-time LOS parameter valuation model.” We connect DNNs in parallel with the control loop of USV, which can obviously improve the trajectory tracking effect.

- (3) Intelligent trajectory tracking of USV is achieved by dynamically connecting the DNN model to a new application developed in MOOS-IvP. MOOS-IvP can be plugged into the real vehicle, so the application developed in this study can be employed in real maritime scenarios as well.

The rest of the paper is organized as follows: Section 2 starts with a description of the general process of trajectory tracking control based on waypoint behavior. Then, the method of connecting DNN in parallel with the control loop of USV is given. Finally, the implementation in the MOOS-IvP system architecture is illustrated and the waypoint behavior dataset is described. Section 3 first presents the overall implementation framework for training and accessing the DNN model into the control loop. Then the principle, construction, and training of the DNN model are given, and the implementation method and process of how to access the trained model into the control loop of USV are explained. In Section 4, the experimental simulation results are presented and analyzed in respect of model training, optimization effect of GB-RBM, and trajectory tracking control effect. Section 5 summarizes the paper.

## 2. Problem Formulation and System Architecture

*2.1. General Process of Trajectory Tracking.* The general process for a USV to perform trajectory tracking is to generate a set of waypoints based on the mission of the voyage. Then, the USV sequentially moves towards the waypoints and follows the planned route sailing. It can be divided into 5 phases, as shown in Figure 1:

- (1) Output speed and steering orders by the guidance algorithm to guide USV towards the next waypoint
- (2) Transform speed and steering orders to throttle and rudder actions
- (3) Under wind, waves and current conditions, the USV navigates in the sea
- (4) Send the feedback of USV speed and heading to the control module, which outputs new throttle and rudder angle actions
- (5) Send the feedback of actual position of USV to the guidance module, which outputs new speed and steering orders

According to the general process above, we consider the feedback effects produced by phases (2)-(4) as a whole and model the trajectory tracking control problem of USV as a waypoint behavior based on guidance algorithm.

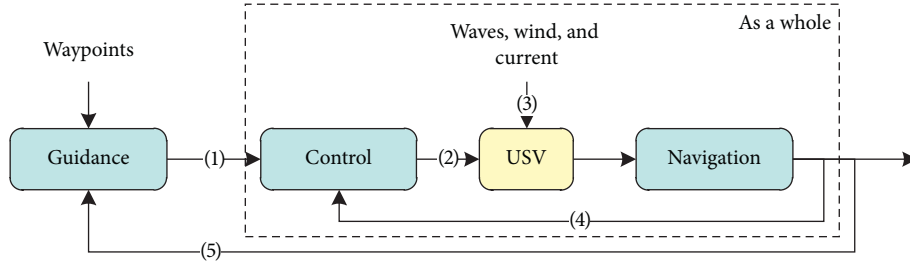


FIGURE 1: The trajectory tracking control scheme for USV.

This study is an optimization of the lookahead-based guidance algorithm by deep artificial neural networks. As shown in Figure 2(a), the lookahead-based guidance algorithm can be formulated as a geometric relationship between the vehicle, the previous waypoint, and the next waypoint [33].

Primarily, the angle formed by the two waypoints is  $\alpha$ , as in

$$\alpha = \tan^{-1}(y_{k+1} - y_k, x_{k+1} - x_k). \quad (1)$$

Then, the lead distance  $l_1$  and lead damper  $l_2$  can be derived from  $\alpha$  and the position of the vehicle  $(x_t, y_t)$ , as shown in

$$l_1 = (x_t - x_k)\cos \alpha + (y_t - y_k)\sin \alpha, \quad (2)$$

$$l_2 = -(x_t - x_k)\sin \alpha + (y_t - y_k)\cos \alpha, \quad (3)$$

where  $l_1$  is the distance between the point of LOS and the vertical foot of the vehicle to the planned route. Generally, it takes 1.5–2.5 times the length of the vehicle.  $l_2$  is the distance from the vehicle to the vertical foot. The trajectory tracking control is to reduce the track error approaching zero by regulating the  $l_1$  and  $l_2$ .

In addition, there are two circles associated with trajectory tracking during the planned route involving several waypoints. The inner circle is called the capture circle, which signifies the arrival of the vehicle to a waypoint while sailing in still water. The outer circle, known as the slip circle, marks the arrival when it is affected by wind, waves, and current, as illustrated in Figure 2(b).

### 2.2. DNN-Based Parallel Process of Trajectory Tracking.

The DNN model proposed in this paper is in parallel with the general control process described above, as shown in Figure 3. We send the data from the navigation system of vehicle to the LOS guidance module and the DNN prediction model simultaneously and send the steering angle of the waypoint to the prediction model in advance. The prediction model consists of two submodels, where DNN-1 predicts the navigational effect firstly. If it works well, there will be no adjustment of parameters. Otherwise, DNN-2 is used to predict the relevant parameters of LOS algorithm, and the lead distance and the lead damper are adjusted to indirectly control the ship navigation to achieve better waypoint behavior effect.

This dual-DNN predictive model of trajectory tracking control using a parallel access approach is different from previous intelligent control models. It is not directly connected to the control loop. There are two advantages:

- (1) It is based on deep learning methodology for predictive model, analogous to ship officer which does not directly change guidance law and control law of the vehicle. If the original algorithm is good, it will not affect the ship navigation. Only when the navigation effect is predicted to be bad, the guidance law parameters are adjusted to improve the navigation effect of vehicle in the kinematic.
- (2) After long-term application and verification, the traditional control model is relatively reliable in engineering. The new model is parallel with the traditional model, which greatly enhances the practical value. In particular, when the navigation effect is good, DNN-2 is not involved in the control. It can also prevent frequent rudder manipulation, which is more in line with the regular mode of ship maneuvering.

**2.3. System Architecture Based on MOOS-IvP.** The autonomous navigation system of USV in this study adopts the system architecture of MOOS-IvP. MOOS-IvP was initially used on the Bluefin Odyssey III vehicle of MIT. The main motivation is to build high-performance autonomous systems [29]. Mission Oriented Operating Suite (MOOS) is a set of software components that provide a framework for the coordinated operation of multiple individual processes. Interval Programming (IvP) is a solution to the problems of multiobjective optimization, which is used for organizing various behaviors to achieve autonomous navigation of USV.

The system architecture of MOOS incorporates publish-subscribe middleware. Each MOOS application (MOOS app) interacts with information by connecting to a MOOS database (MOOSDB), and they form a star topology. As shown in Figure 4, the general process for trajectory tracking control of USV is implemented by a set of MOOS apps. The “pHelmIvP” app is a guidance module, “pMarinePID” app is a control module, “pNodeReport” app is a navigation module, and “uSimMarine” app is a simulation module of wind, waves, currents, and hull effects. All of them are connected to MOOSDB, constituting the autonomous



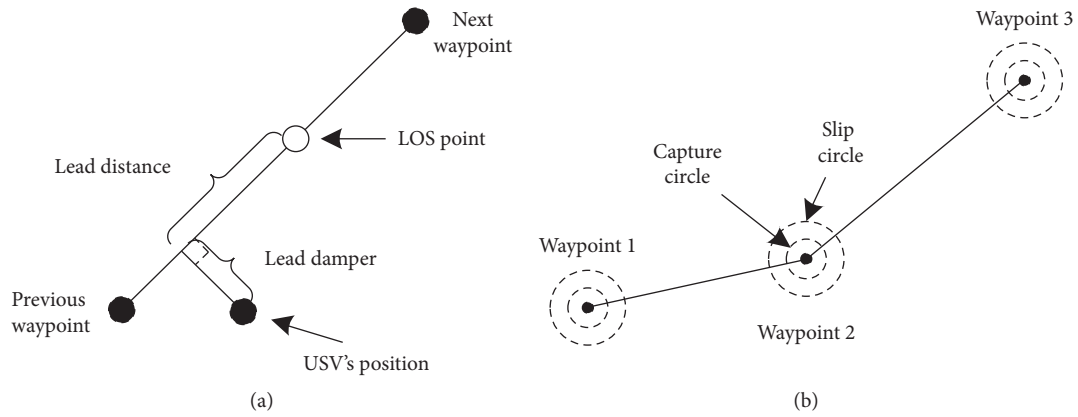


FIGURE 2: Waypoint behavior of USV: (a) lookahead-based guidance law; (b) capture circle and slip circle.

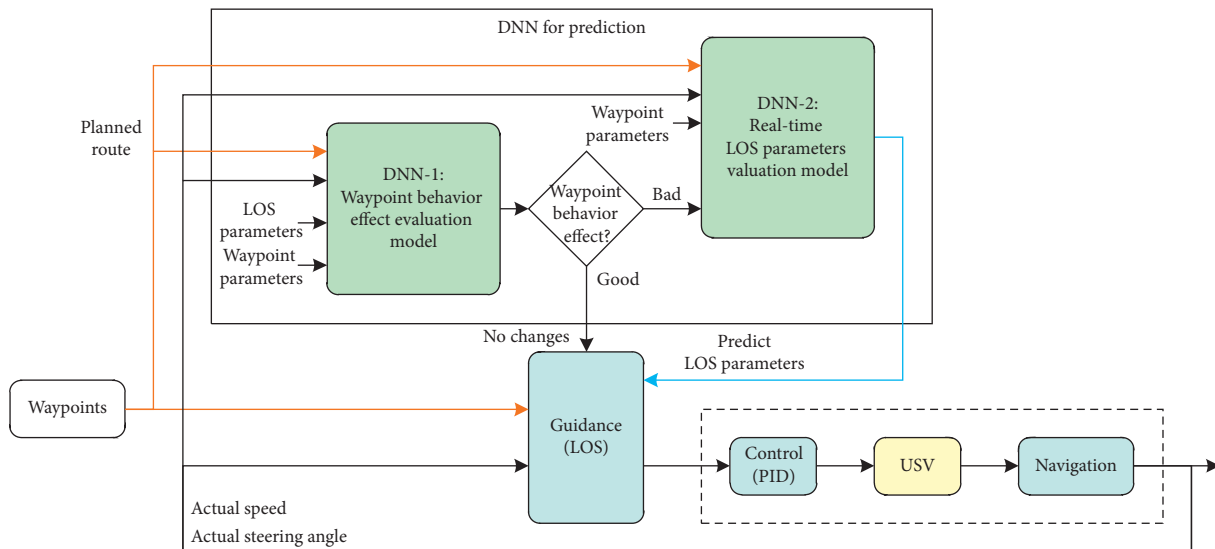


FIGURE 3: The relationship between DNN prediction model and trajectory tracking control process of USV.

navigation simulation system of USV. The DNN-based parallel control process is realized by adding a MOOS app called “pDeepLearning.” “pDeepLearning” is a new MOOS app developed in this study, which subscribes to the vehicle speed and course and publishes the predicted values given by DNN model.

The guidance algorithm of USV is implemented by the instance of waypoint behavior contained in “pHelmIvP.” The configuration parameters of behavior instance correspond to the variables of guidance algorithm. These parameters are stored in a “.bhv” file and invoked during the initialization of the mission. Once the mission is launched, the “.bhv” file can no longer be modified.

In this study, it is necessary to dynamically configure USV waypoint behavior with the predictions given by DNN model. This allows the USV to adjust the guidance parameters according to the navigation state. We use the “updates” parameters to publish the variables to the MOOSDB. To do this, it is necessary to configure a WPT\_UPDATE variable in the “.bhv” file, as shown in Table 1.

In the process of voyage, the WPT\_UPDATE variable is used to publish specific content for changing parameters in the configuration file. For example, if WPT\_UPDATE = “lead\_distance = 6.90,” it would immediately change the lead distance value of the USV from 8.0 to 6.9 in the original configuration file.

**2.4. Dataset and Statistical Analysis.** In previous work, we have made the waypoint behavior dataset [7]. The training samples comprise six features that are speed and steering angle, lead distance, lead damper, capture radius, and slip radius. These features can be classified into 3 categories, which are related to the definition of waypoint, guidance algorithm, and navigation process, respectively, as shown in Table 2. The training labels are different effect levels for USV, belonging to levels I~III; therein, corresponding times vehicle lengths are as shown in Figure 5.

In this study, a preliminary statistical analysis of the dataset is conducted. From the statistical analysis in Figure 6,

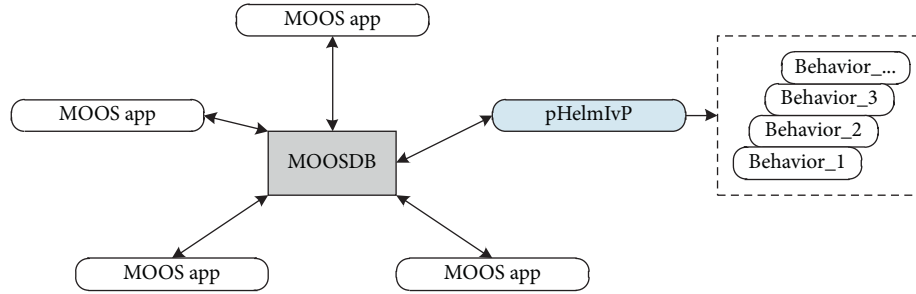


FIGURE 4: The topology structure of MOOSDB for trajectory tracking of USV.

TABLE 1: Waypoint behavior configuration file.

Configured variable	Configured value
Behavior name	BHV_Waypoint
Behavior instance name	waypt_survey
Lead distance	8.0
Lead damper	5.0
Capture circle radius	7.0
Slip circle radius	14.0
Updates parameter	WPT_UPDATE

TABLE 2: Categories and features of waypoint behavior for USV in dataset.

Number	Categories	Features
1	Related to the definition of waypoint	Capture radius
2		Slip radius
3	Related to guidance algorithm	Lead distance
4		Lead damper
5	Related to navigation	Speed
6		Steering angle

the categorical items are evenly distributed, and no special statistical patterns can be seen. However, these parameters do affect the effectiveness of USV. It is necessary to mine them with deep learning methods.

### 3. Methods

**3.1. Dual-DNN Prediction Model.** A predictive DNN model for trajectory tracking control is established. It consists of two submodels. The first submodel is DNN-1, which is used to predict the influence of waypoint behavior parameters on the drift effect. It is a “6-input-5-output” classification model, using waypoint behavior dataset for training. The second submodel is DNN-2, which is used to estimate the values of two guidance parameters: lead distance and lead damper. It is a “4-input-2-output” regression model, using the part of waypoint behavior dataset with better navigation performance for training.

The DNN model is designed to identify the complex effects of various factors and their combinations in the waypoint behavior of USV on high-dimensional spatial planes. In addition, the nonlinear effects are caused by the power unit installed in the vehicle and the effect of wind and

waves on the vehicle. They have been involved in the acquisition of the data of waypoint behavior.

**3.1.1. DNN-1:Waypoint Behavior Effect Evaluation Model.** DNN-1 is a feedforward network with  $N_{6-6-7-7-8-7-6-5}$ , as shown in Figure 7. The input layer corresponds to six features of waypoint behavior. The activation function in hidden layer is ReLU function. The output layer corresponds to five levels of effect. The loss function is cross-entropy.

**3.1.2. DNN-2:Real-Time LOS Parameter Valuation Model.** DNN-2 is a four-layer feedforward network  $N_{4-4-3-2}$ , as shown in Figure 8. The hidden layer also uses the ReLU function, and the output layer uses softmax function, so that the model can predict lead distance and lead damper value. As a regression model, the “mean square” is used as a loss function to determine the deviation between the predicted value and the desired value.

**3.1.3. Model Training and Saving.** The two models are built and trained separately in Keras, and we train the DNNs using a gradient descent optimization algorithm with learning rate of 0.001 and batch size of 100. All experiments are performed on a 2.5 GHz Xeon 4215 CPU and two NVIDIA TITAN RTX GPUs.

The two submodels are trained by using the waypoint behavior dataset and its subsets, respectively. There are 16200 samples in the dataset. We perform 12000 iterations to train the classification model. The training set, validation set, and test set are randomly picked as 9720, 3420, and 3420, respectively. After training, in the validation set, the DNN model with the highest accuracy is stored.

For the regression model, 8000 iterations are trained. The dataset is divided into three parts randomly. 3927 samples are used as the training set. The validation set has 1309 samples, and the remainder are the test set. The DNN model with the lowest mean square error is saved.

In respect of the number of training parameters, the two submodels have 330 and 63 parameters to be trained, respectively. From the training results, the network size is well suited to the extraction of the waypoint behavior features, and the generalization capability satisfies practical applications.

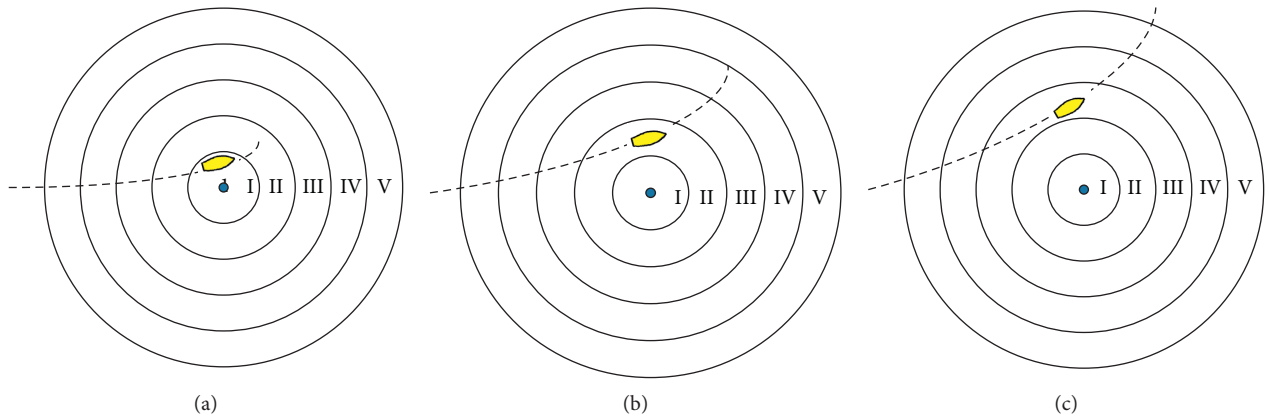


FIGURE 5: Effect of waypoint behavior. (a) The drift range of USV belongs to level I. (b) The drift range of USV belongs to level II. (c) The drift range of USV belongs to level III.

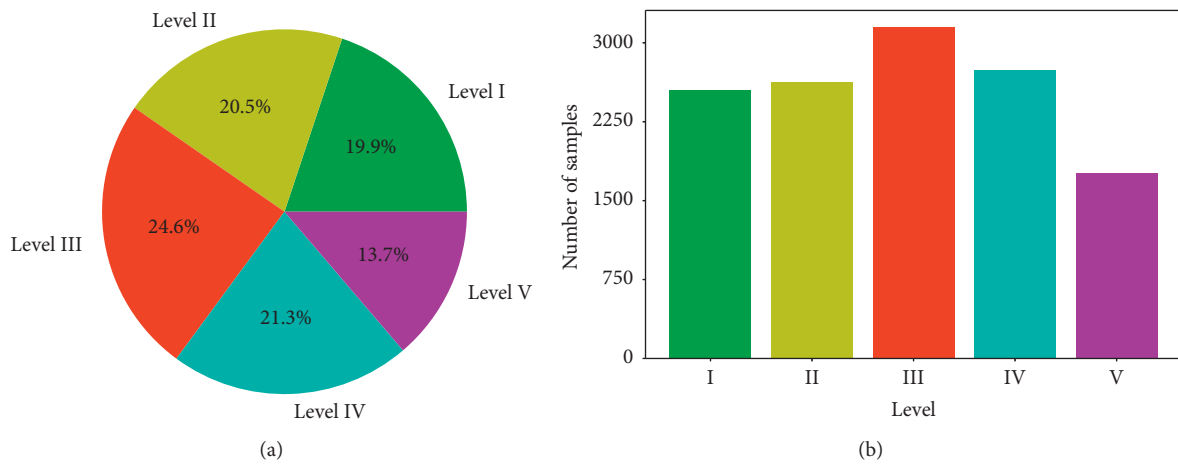


FIGURE 6: The statistical analysis of waypoint behavior dataset: (a) percentage of quantity distribution at five levels; (b) quantity distribution at five levels.

3.2. Optimization by Using GB-RBM. The trained fully connected network model achieves an accuracy of 84.9% in the classification effect of waypoint behavior, which is the result of previous stage [7]. We find that setting different initialization parameters in the first layer of DNN has greatly affected the accuracy. This is due to the fact that DNN is easy to fall into local optimization. Therefore, we hope to further optimize DNN by tuning the initialization parameters.

3.2.1. The Training Method of DNN with GB-RBM. The method of “pretraining and fine-tuning” can effectively solve the difficult problem of neural network training [25]. This is considered both a training method for deep learning and a method for tuning initialization parameters. Therefore, the training is separated into two steps: “pretraining for first layer” and “fine-tuning.” The first step is the pretraining step, using GB-RBM for reconstruction training of the first layer by contrast divergence (CD) algorithm. This step trains 42 and 20 parameters as fixed values for two submodels, respectively. It reduces the calculation complexity, simultaneously ensuring

that the mapping of the feature vector to the feature space is optimal.

The second step is to implement the fine-tuning of the whole network. This is done by taking the pretrained parameters as initial values and applying a back propagation (BP) algorithm in the DNN to learn further on the training set. Then the trained well dual-DNN is used for prediction. Figure 9 summarizes the proposed method.

3.2.2. The Construction of GB-RBM. The RBM is a neural network based on energy. It consists of two layers that are the visible layer and the hidden layer. The visible layer is generally used to describe the observation data, while the hidden layer can be regarded as the feature extraction layer. A six-dimension feature vector as visible layer neurons is utilized to build GB-RBM. Through reconstruction training, the RBM can learn the inner relationship between the 6-dimensional features. However, the RBM was originally developed for binary vector coding and decoding, so both the visible and the hidden layer variables are binary. In this study, the type of visible layer vector is

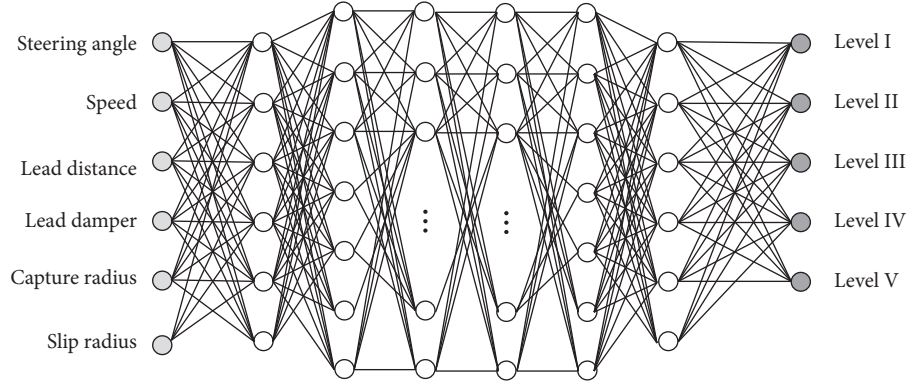


FIGURE 7: The structure of DNN-1.

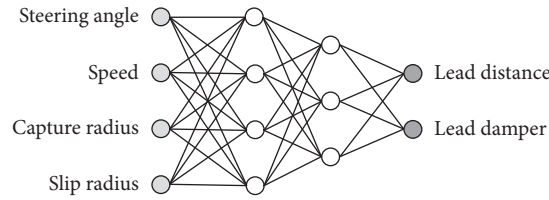


FIGURE 8: The structure of DNN-2.

numerical, so RBM cannot be used. According to the paper [28], the Gaussian–Bernoulli restricted Boltzmann machine (GB-RBM) can solve this problem. As shown in Figure 10, the GB-RBM constructed is of numerical type for its visible layer vector and Boolean type for its hidden layer variable, which conforms to the requirements of numerical type feature vectors.

The GB-RBM is a neural network based on energy. The combined energy functions of the visible and hidden variables are

$$E(x, h) = -h^T W \frac{x}{\sigma} - \frac{(x - c^T)^2}{2\sigma^2} - b^T h, \quad (4)$$

where the visible layer random vector  $x = [x_1, x_2, x_3, \dots, x_6]^T$ ; the hidden layer random vector  $h = [h_1, h_2, h_3, h_4, h_5, h_6]^T$ ; the weight matrix  $W \in R^{6 \times 6}$ , and each element  $w_{jk}$  is the weight of connections between the visible layer variable  $x_k$  and the hidden layer variable  $h_j$ ; the bias  $c \in R^6$  and  $b \in R^6$ ; and  $\sigma$  is the standard deviation associated with Gaussian visible vector  $x$ .

After defining the joint energy function of  $x$  and  $h$ , it can get the joint probability of  $x$  and  $h$ , as shown in (5).  $Z$  in (6) is the normalized factor also known as the partition function, which is the sum numbers  $\alpha$  of all the states of the system.

$$\mathbb{P}(x, h) = \frac{e^{-E(x, h)}}{Z}, \quad (5)$$

where

$$Z = \sum_{\alpha} e^{-E_{\alpha}/KT}. \quad (6)$$

**3.2.3. The Training Process of GB-RBM.** The training of GB-RBM is divided into two processes: (1) the coding process,

also known as forward propagation; (2) the decoding process, also called back propagation or reconstruction process.

In the coding process, given the features in the visible layer, calculate the probability that a neuron in the hidden layer will be activated by sigmoid function, as shown in

$$\mathbb{P}(h_j = 1 | x) = \text{sigmoid}\left(w_j \frac{x}{\sigma^2} + b_j\right). \quad (7)$$

Then, the randomizer generates a number from 0 to 1. If the number is less than the calculated  $h_j$ , then the hidden layer node takes 1; otherwise, it takes 0.

In the decoding process, given the current state of all neurons in the hidden layer, we calculate the probability that a neuron in the visible layer will be activated, as shown in (8). Other than RBM, the mean  $\mu$  and variance  $\sigma^2$  that conform to the Gaussian distribution should be added for GB-RBM.

$$\mathbb{P}(x_k = 1 | h) = N(c_k w_k + x, \sigma^2), \quad (8)$$

where  $N(\mu, \sigma^2)$  denotes the Gaussian probability density function with mean  $\mu$  and standard deviation  $\sigma$ .

Then, the randomizer generates a number from 0 to 1. If the number is less than the calculated  $x_k$ , then the visible layer node is  $x_k$ ; otherwise, it takes that random number.

After training by performing coding process and decoding process alternately, the reconstruction error of  $x$  and  $\mathbb{P}(x_k = 1 | h)$  is very small, which indicates that the GB-RBM tends to stabilize.

In the training process of GB-RBM, an efficient CD algorithm which is common in deep learning has been applied, and it is illustrated in Algorithm 1. In this way, GB-RBM can be trained in the same way as a normal RBM. In



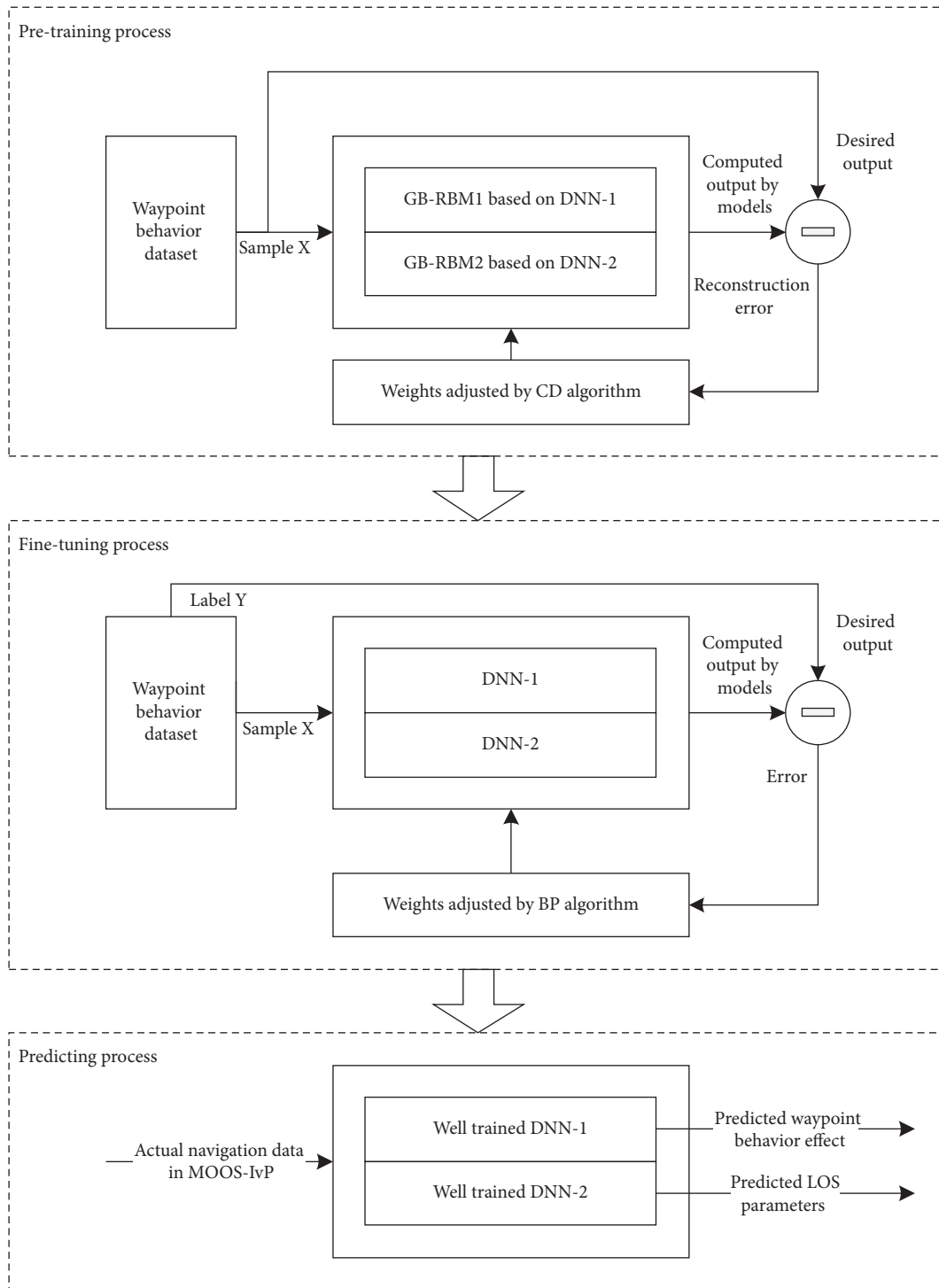


FIGURE 9: The process of DNN training with GB-RBM.

Section 4.1, the experimental results show the improvement of classification accuracy after using GB-RBM.

### 3.3. Model Prediction and Invocation

3.3.1. *Development of Prediction Scripts.* The two DNNs are optimized by GB-RBM, followed by fine-tuning. After

training, the saved model structure and parameters are used to regenerate the prediction script. The essence of the prediction script is a function that calls the neural network model and gives the prediction results according to the input variables.

LevelPredictor.py is a python script used to evaluate the effects of waypoint behavior, whose main function is to call the saved .h5 classification model file to predict. LDPredictor.py is a

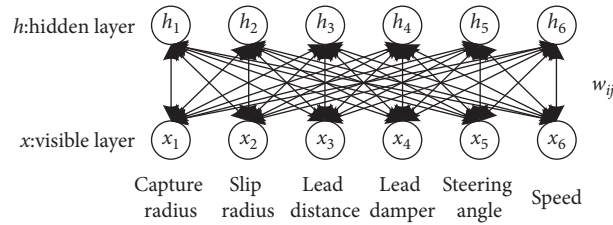


FIGURE 10: The structure of GB-RBM for feature mining in waypoint behavior.

python script used to predict the waypoint behavior parameters, and its main function is to estimate the LOS parameters.

**3.3.2. Invocation of DNN Model by Using “pDeepLearning”.** A new MOOS app is developed to perform the computing of the trajectory tracking control prediction model, so as to establish the interface between the trained deep neural network and the MOOS-IvP platform. “pDeepLearning” is the key module for all types of information interaction. It is a C++ program inherited from the CMOOSApp class in MOOS. On the one hand, pDeepLearning is used to publish and subscribe the data in MOOS. On the other hand, DNN implemented as python scripts is called by pDeepLearning.

We deployed a set of MOOS apps and MOOSDB, for the USV, that perform waypoint behavior. The DNN is then integrated into the MOOS-IvP by loading “pDeepLearning.” Figure 11 shows the system structure of USV with DNN model.

During the voyage, “pDeepLearning” first receives the speed and planned steering angle of USV from MOOS-IvP. Then, “pDeepLearning” calls the LevelPredictor.py and LDPredictor.py scripts that are used in predicting the waypoint behavior effect and LOS parameters. Finally, the WPT\_UPDATE variable described in Section 2.3 is used to publish the parameters lead distance and lead damper into MOOS-IvP. Section 4.3 reveals the performance of pDeepLearning.

## 4. Results and Discussion

The experimental results are carried out in two aspects: Firstly, we take the previous research as the benchmark. The classification accuracy is improved after the DNN model adopts the “pretraining and fine-tuning” method by GB-RBM. Secondly, an experimental platform is constructed based on MOOS-IvP, and the effect of running deep learning application pDeepLearning in MOOS-IvP for trajectory tracking control can be seen.

**4.1. The Effect of Training GB-RBM.** Figure 12 shows that the first layer of the classification model and regression model used GB-RBM matter to carry on the pretraining process, respectively, in which the horizontal axis shows the epoch times, and the vertical axis represents the reconstruction error between the visible layer and the hidden layer. Besides, we choose that mean square error which is commonly used in deep learning training. The

experimental results show that the reconstruction error of the two GB-RBM are smaller through training; the reconstruction error of the classification model tends to be 0.09, as shown in Figure 12(a)), and that of the regression model tends to be 0.03, as shown in Figure 12(b), which suggests that the artificial neural network has the ability to restore the original data after transformation between the visible layer and the hidden layer. The GB-RBM proposed has learned the features of the waypoint behavior.

**4.2. The Effect of Classification Accuracy.** The accuracy varies for different depths and widths of the DNN structure. We compare the accuracy between initialization parameters using GB-RBM pretraining and no pretraining phase. The results are shown in Tables 3 and 4, respectively.

Experiments show that the structure of DNN, which is 6-6-7-7-8-7-6-5 nodes in each layer, has a maximum accuracy of 91.3% and 88.9% on the verification set and test set separately after pretraining.

Then, we employ the well-trained DNN to predict the effect of USV waypoint behavior at different speeds and steering angles. The results of the experiments are presented below.

Figures 13 and 14 show the prediction of the DNN for different steering angles and speeds, respectively, without any changes in other parameters. Figures 13(a) and 14(a) present reference values, Figure 13(b) show the predicted values without pretraining, and Figures 13(c) and 14(c) display the predicted values with pretraining.

It can be seen that the same trends are present in the predictions and the ground truth, and the unsupervised learning process using GB-RBM can improve the predicted accuracy.

### 4.3. The Effect of Trajectory Tracking

**4.3.1. Simulation Preparation.** As mentioned above, the goal of this study is to predict the waypoint behavior of USV through DNN to optimize its trajectory tracking effect. A comparative navigation simulation experiment based on MOOS-IvP platform is conducted. Two USVs with the same type and length (7 m) are deployed. The first USV named alpha does not use DNN, and the second one named Alder uses dual-DNN prediction model. Other than that, the configuration parameters of the two USVs are identical. The two USVs start from the

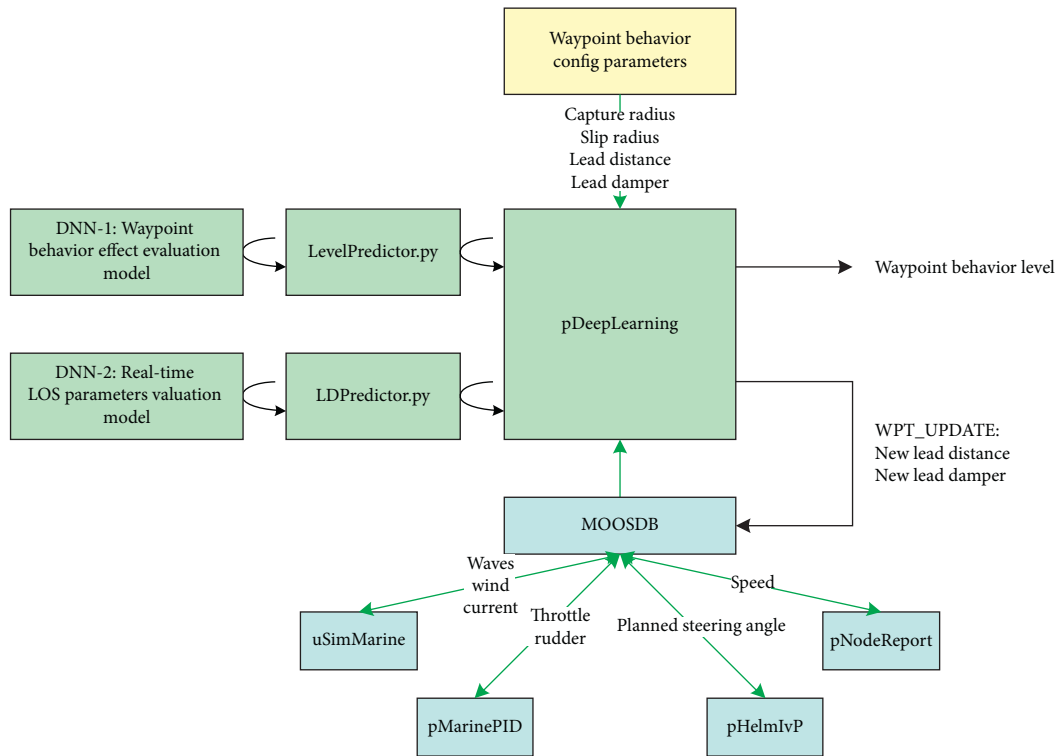


FIGURE 11: The information interaction process between pDeepLearning, environment, and neural network model.

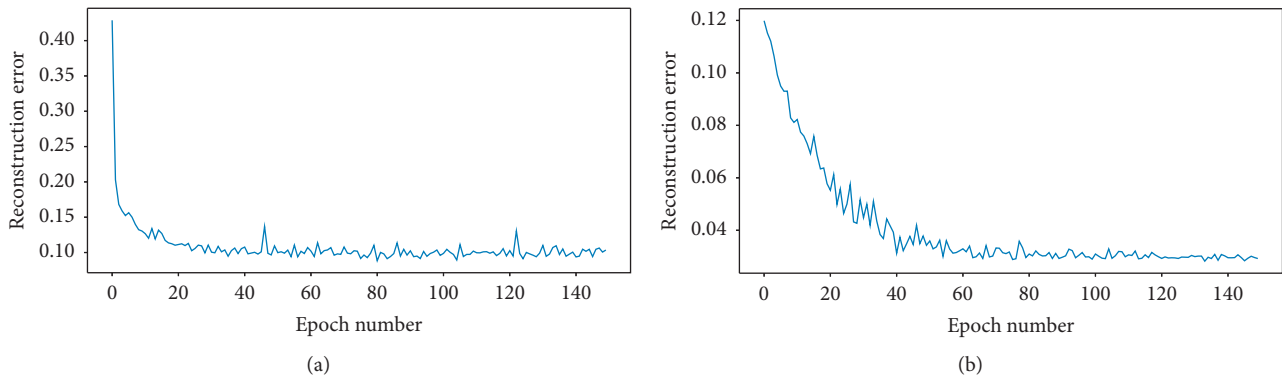


FIGURE 12: The pretraining process by using GB-RBM: (a) training GB-RBM for the 1st layer of classification model; (b) training GB-RBM for the 1st layer of regression model.

same initial point and track a planned route consisting of five waypoints.

The dual-DNN prediction model adopted by the USV Alder is implemented by running pDeepLearning application to predict the behavior of the waypoint in real time. The experimental results are recorded by pLogger application in MOOS and extracted and analyzed by alogview toolbox.

The MOOS-IvP simulation platform is shown in Figure 15, in which the small window shows the situation of USV Alder running pDeepLearning for prediction. The main configuration parameters of two USVs and their waypoint behavior configurations are given in Table 5.

After experiment, we compare the tracking effect and performance index of alpha and Alder, to give the overall evaluation and analysis of the behavior effect of waypoint,

and then we analyze the prediction effect by using DNN model and the influence of the model on the speed and course stability.

4.3.2. *Trajectory Tracking.* Figures 16 and 17 show the trajectory of USV sailing towards five waypoints from the initial position. Figure 16 is the overall picture and Figure 17 is a larger version of each waypoint.

As shown in Figure 16, the center of five red circles is the five waypoints in the planned route, among which the waypoint behavior effect evaluation circle is drawn from the inside to the outside with the radius of 1 to 5 times length of USV. The black dashed line is a planned route connected by five waypoints. The green line is the trajectory of USV Alder

<p><b>Input:</b> Dataset <math>x(n)</math>, <math>n = 1, \dots, N</math>;</p> <p><b>Output:</b> <math>W, c, b</math></p> <p>(1) Set learning rate : <math>\alpha = 0.001</math>, epoch: <math>T = 150</math>;</p> <p>(2) Initial : <math>W \leftarrow 0</math>, <math>c \leftarrow 0</math>, <math>b \leftarrow 0</math>;</p> <p>(3) Calculate mean value and variance <math>\sigma</math> of vectors in dataset;</p> <p>(4) for <math>t = 1 \dots T</math> do</p> <p>(5)   for <math>n = 1 \dots N</math> do</p> <p>(6)     choose an input vector <math>x</math>, calculate <math>p(h_j = 1 x)</math> by using Equation (4), and randomly choose a hidden vector <math>h</math> according to the distribution;</p> <p>(7)     calculate positive gradient <math>xh/\sigma^2</math></p> <p>(8)     according to <math>h</math>, calculate <math>p(x_k = 1 h)</math> by using Equation (5), obtain <math>x'</math></p> <p>(9)     according to <math>x'</math>, calculate <math>p(h_j = 1 x')</math> by using Equation (4), obtain <math>h'</math>;</p> <p>(10)    calculate reverse gradient <math>x'h'/\sigma^2</math></p> <p>(11)    <math>W \leftarrow W + \alpha((xh/\sigma^2) - (x'h'/\sigma^2))</math></p> <p>(12)    <math>c \leftarrow c + \alpha((x/\sigma^2) - (x'/\sigma^2))</math></p> <p>(13)    <math>b \leftarrow b + \alpha(h - h')</math></p> <p>(14)    end</p> <p>(15) end</p>
--

ALGORITHM 1: CD for training GB-RBM in the waypoint behavior dataset.

TABLE 3: Training effect by using different depth structures.

Depth	Accuracy on validation set		Accuracy on test set	
	No pretraining	Pretraining	No pretraining	Pretraining
5	80.1	84.8	72.3	77.8
7	81.4	85.1	73.5	75.2
8	88.2	91.3	84.9	88.9
9	78.2	82.5	73.5	75.9
11	76.7	80.8	67.4	70.1
12	52.1	55.7	40.8	42.2

which uses DNN to predict. The blue line is the trajectory of USV alpha without using the prediction model. It can be seen that after using the model prediction, the USV deviated from the waypoint by a smaller distance with each turn.

Figure 17 is an enlarged picture of USV sailing to each waypoint. It can be seen that, in every waypoint, the green track line is closer to the waypoint than the blue track line, which meets the standard of waypoint behavior effect, sometimes by up to one level. This indicates that the USV with prediction model has learned the relationship between the parameters and effect level in waypoint behavior. However, in Figure 17(d), the effects of two USVs are not good, because the steering angle of the waypoint is an acute angle, which falls outside of the dataset, so the overall deviation is large for both.

**4.3.3. The Analysis of Trajectory Tracking Error.** The error between the actual track and the planned track of two USVs is depicted in Figure 18, where the horizontal axis represents the time in seconds, and the vertical axis represents the tracking error in meters. The green line signifies the trajectory tracking error of USV Alder with using DNN for prediction. The blue line signifies the trajectory tracking error of USV alpha without using the

model prediction. The black dashed line signifies the course of advance. It can be observed that the tracking error is smaller after adjusting by the prediction model. In particular, the error is smaller when the course changes.

Table 6 shows the performance quantitative indicators. From the perspective of mean error and variance, the values of USV by using the prediction model are smaller. From the view of the integrated absolute error (IAE) and the time integrated absolute error (ITAE), the USV with prediction model has better transient and steady-state performance.

**4.3.4. The Effect of Waypoint Behavior.** Figure 19 shows a comparison of waypoint behavior effect after modifying parameters by the prediction model. Among them, the green line is the waypoint behavior effect after optimization of LOS parameters by the prediction model, and the blue line is the waypoint behavior effect without modification of the prediction model. It is clear that the waypoint behavior effect is better after optimization. The black line shows the changes of steering angle, which serves as a reference, indicating that once the steering



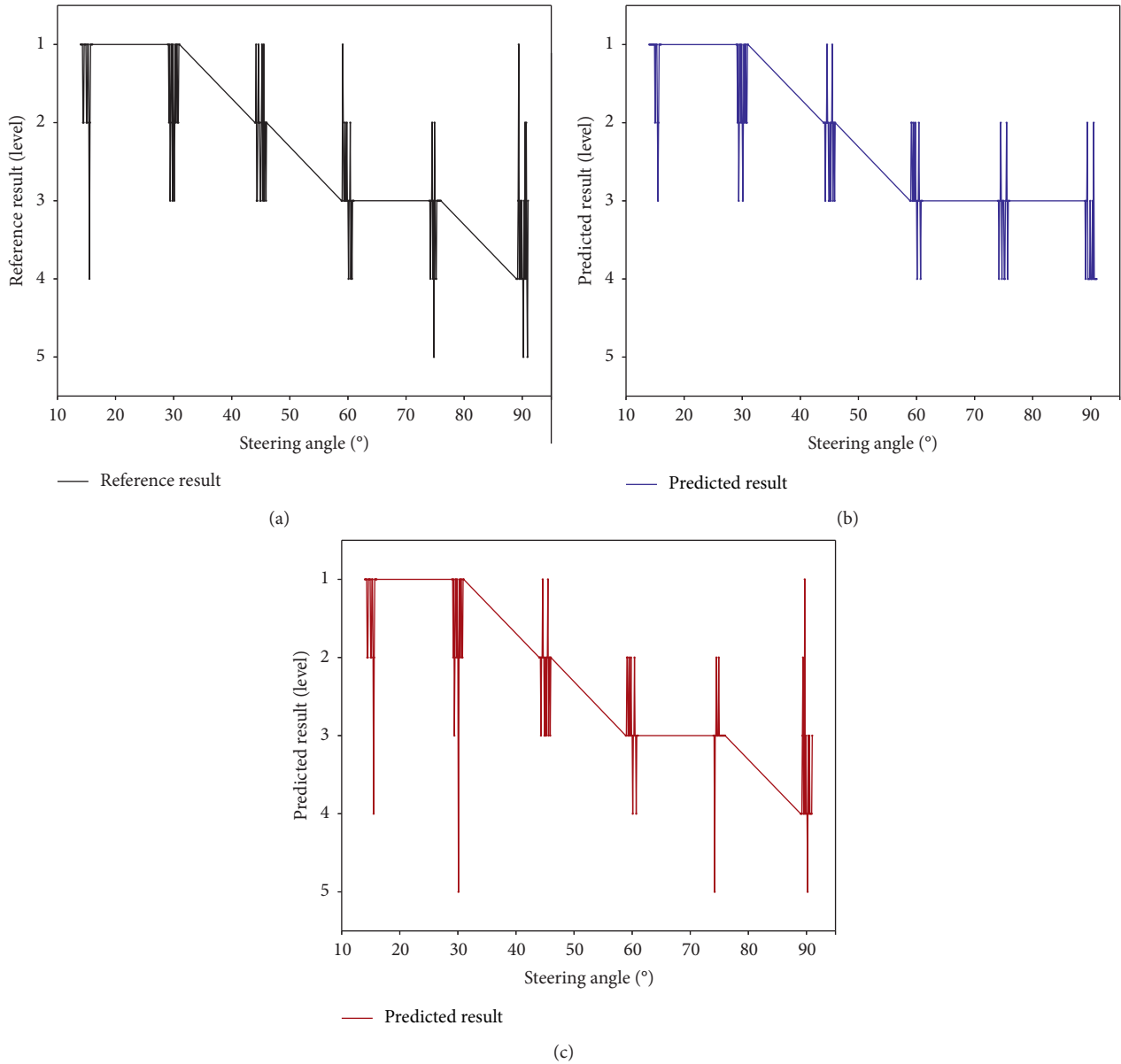


FIGURE 13: Effect level at different steering angles: (a) reference level from test set; (b) predicted level from model without GB-RBM pretraining; (c) predicted level from model with GB-RBM pretraining.

angle changes, the model will predict a new waypoint behavior effect level value.

The effect of waypoint behavior is shown in Table 7; the effect was improved by 1 level after LOS parameters were adjusted by the prediction model.

4.3.5. The Effect of Prediction for LOS Parameters.

Figure 20 shows the lead distance and lead damper values which are related to the guidance law predicted by DNN-2. The green line and blue line signify the predicted value of lead distance and lead damper, respectively. The black line signifies the change of steering angle and speed,

respectively. The black line and the red line are used as the reference, indicating that when the steering angle changes, the model will predict new lead distance and lead damper. In addition, during the voyage, predicted lead damper values have been affected by the change of speed.

4.3.6. The Influence on Speed and Course.

Figure 21 shows the changes of speed and course during the voyage. The scale of Figure 21(a) is the entire navigation process, and Figure 21(b) is the amplification of the second steering process. The green line is the speed and course of USV Alder by using the prediction model, the blue line is the

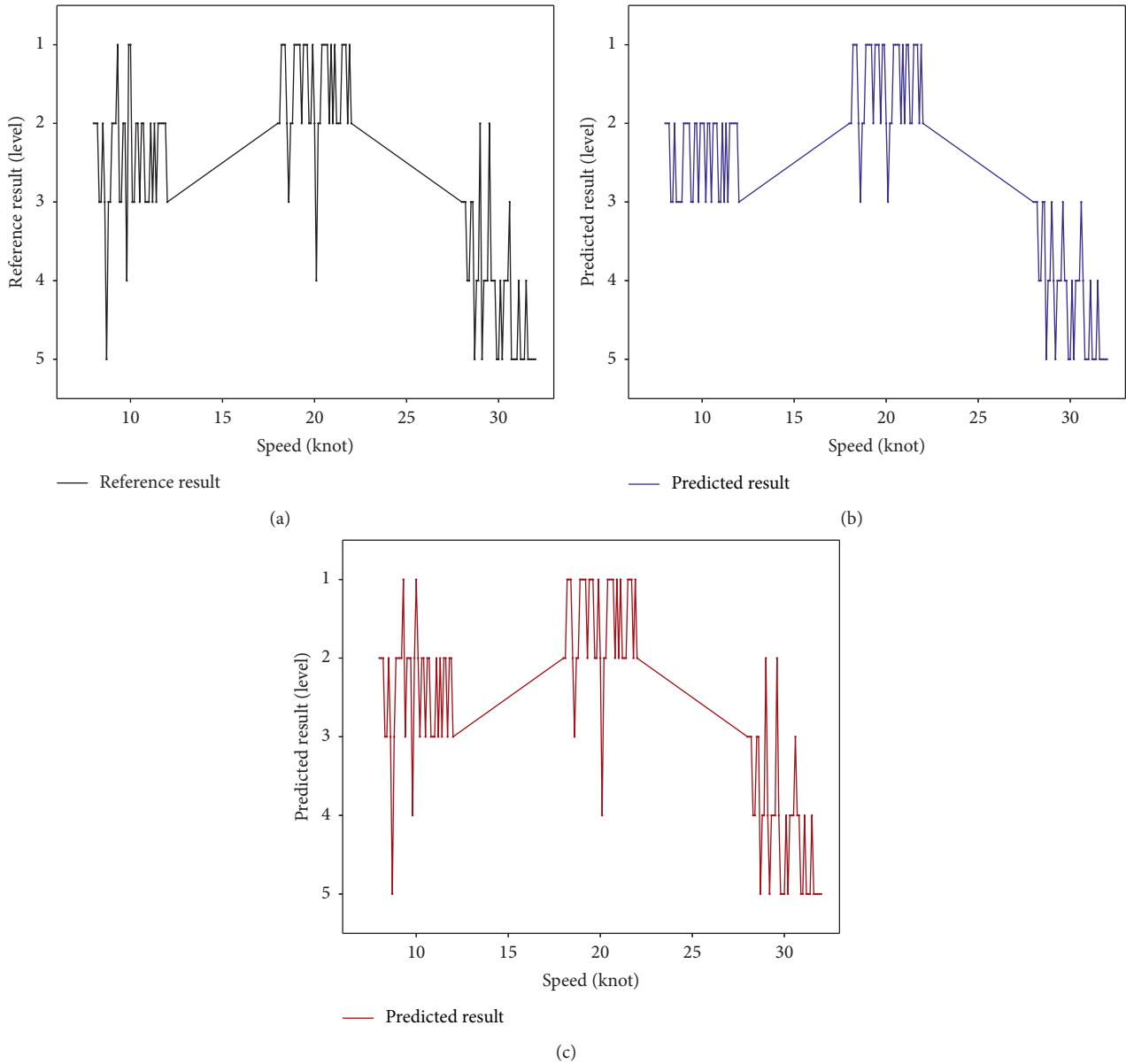


FIGURE 14: Effect level at different speeds: (a) reference level from test set; (b) predicted level from model without GB-RBM pretraining; (c) predicted level from model with GB-RBM pretraining.

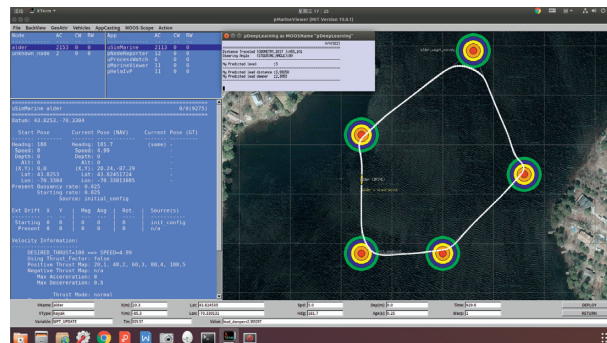


FIGURE 15: The USV simulation environment based on MOOS-IvP.

TABLE 4: Training effect by using different width structures.

Width	Accuracy on validation set		Accuracy on test set	
	No pretraining	Pretraining	No pretraining (%)	Pretraining (%)
6-6-6-6-6-6-5	82.8	83.2	81.0	84.5
6-7-7-7-7-7-5	82.8	87.3	76.2	80.3
6-6-7-7-8-7-6-5	88.2	91.3	84.9	88.9
6-8-8-8-8-8-5	63.9	66.2	61.9	71.8
6-10-10-10-10-10-5	9.7	71.4	66.7	70.2
6-6-6-6-6-6-5	82.8	82.9	81.0	82.6

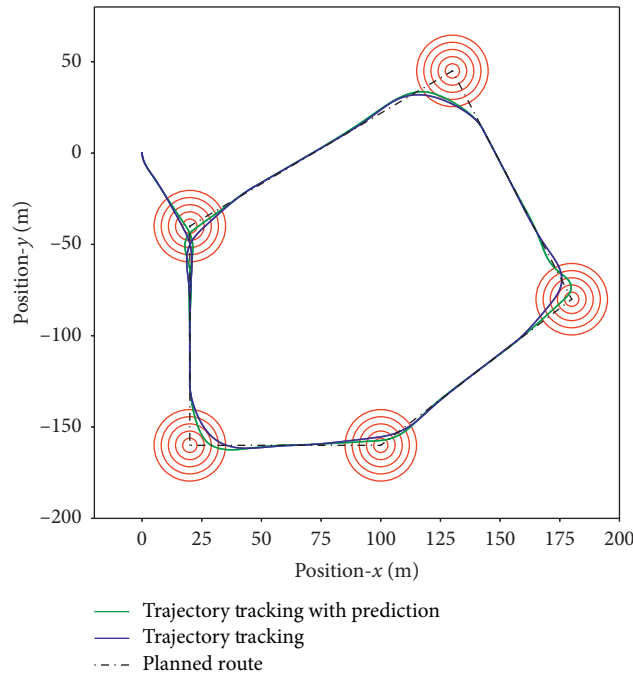


FIGURE 16: The trajectory of USV Alder and alpha in simulation environment.

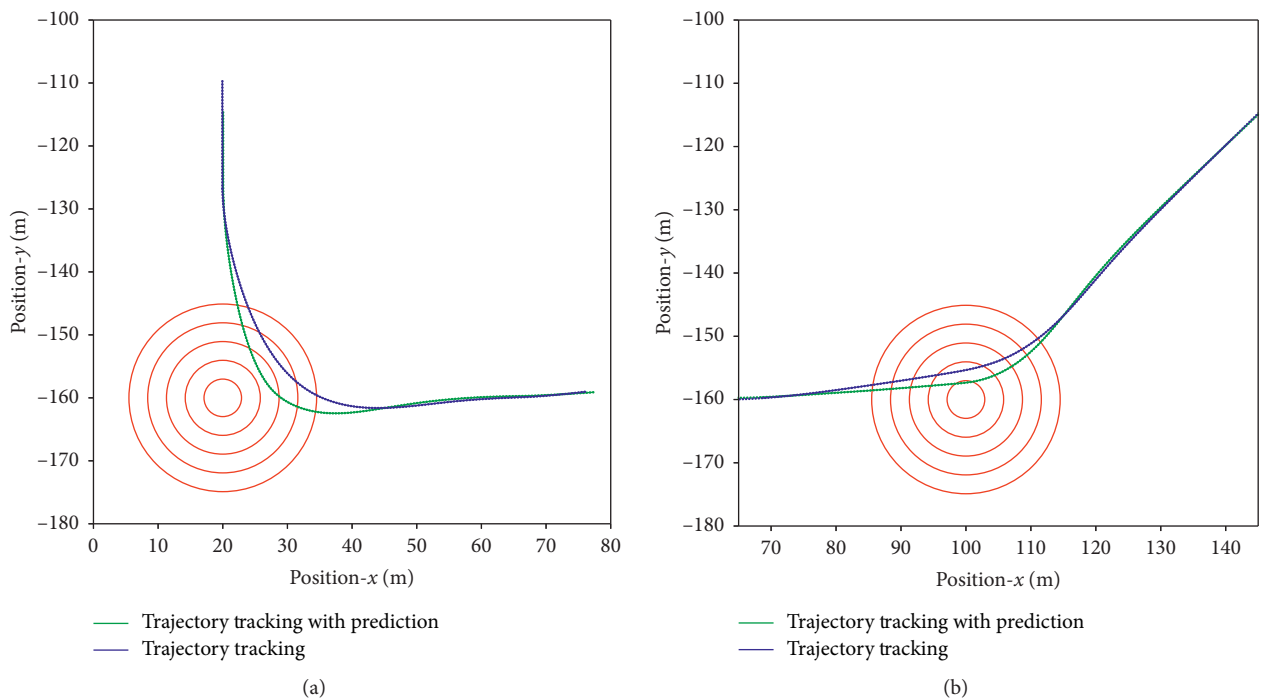


FIGURE 17: Continued.

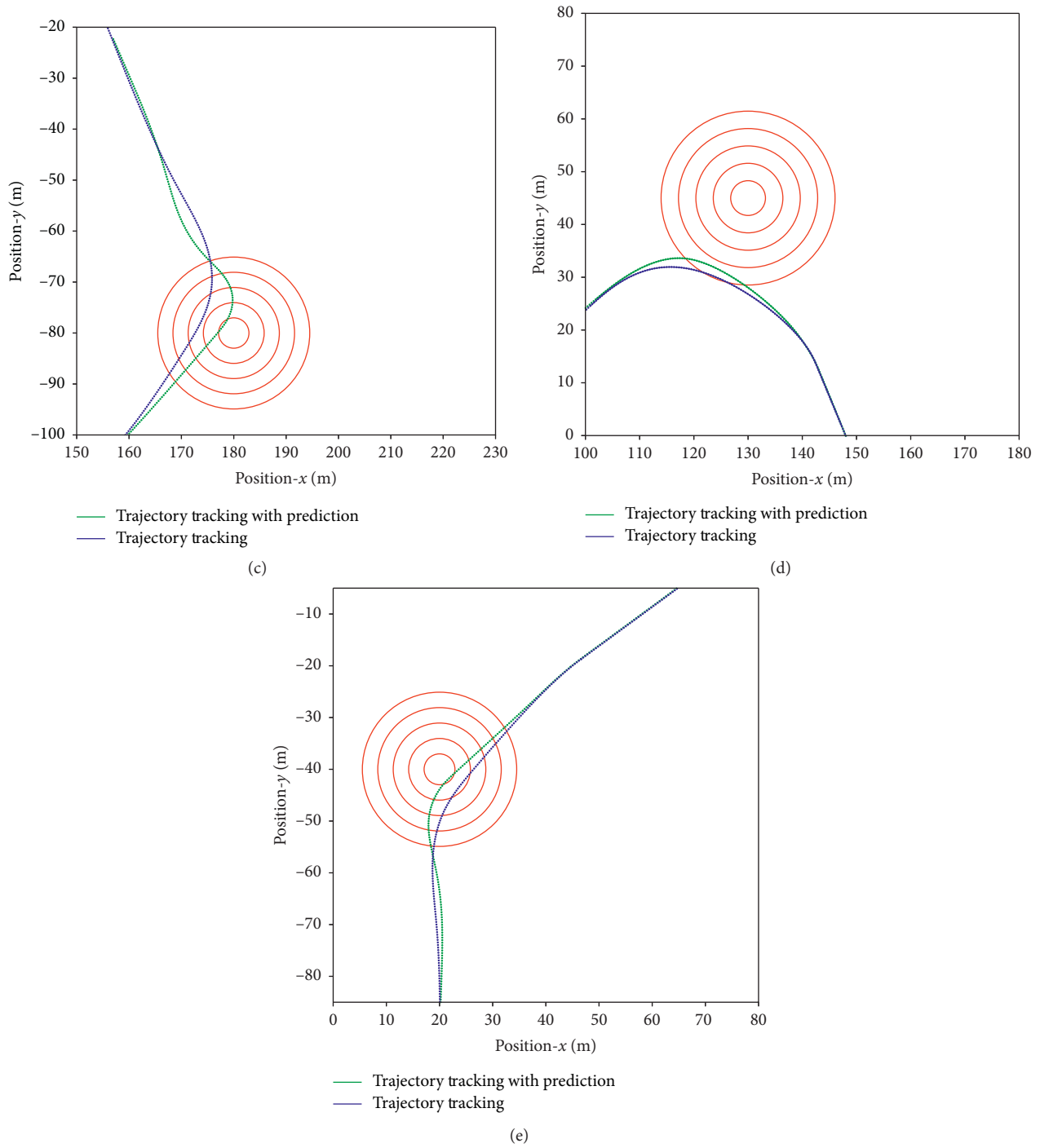


FIGURE 17: The trajectory of USV Alder and alpha in each waypoint: (a) the enlarged picture of waypoint when steering angle is  $90^\circ$ ; (b) the enlarged picture of waypoint when steering angle is  $135^\circ$ ; (c) the enlarged picture of waypoint when steering angle is  $113^\circ$ ; (d) the enlarged picture of waypoint when steering angle is  $74^\circ$ ; (e) the enlarged picture of waypoint when steering angle is  $106^\circ$ .



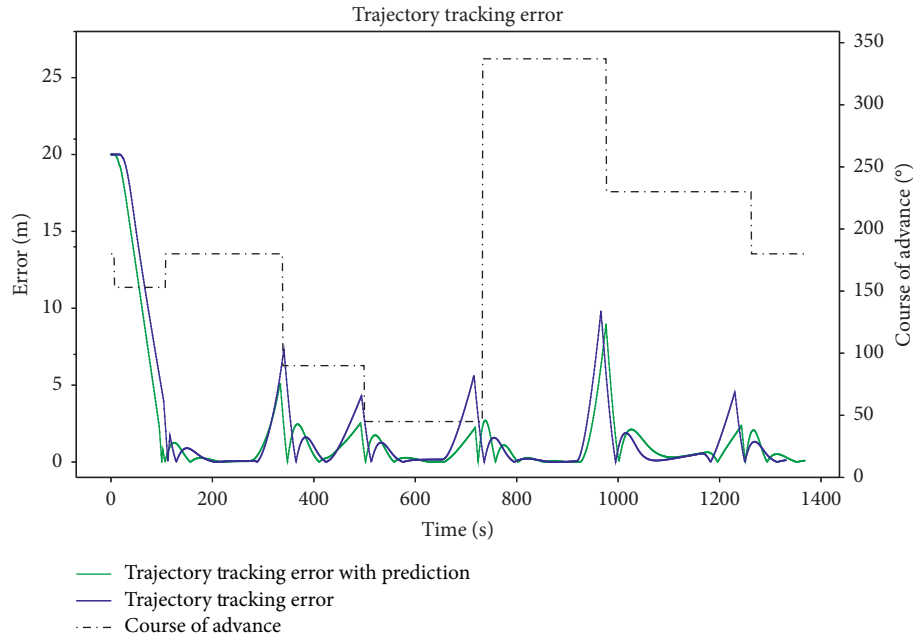


FIGURE 18: The trajectory tracking error of USV Alder and alpha.

TABLE 5: The configuration parameters of USVs.

USV	Length (m)	Capture radius (m)	Slip radius (m)	Lead distance (m)	Lead damper (m)	Speed (m/s)
Parameters	7	35	70	8	1	5

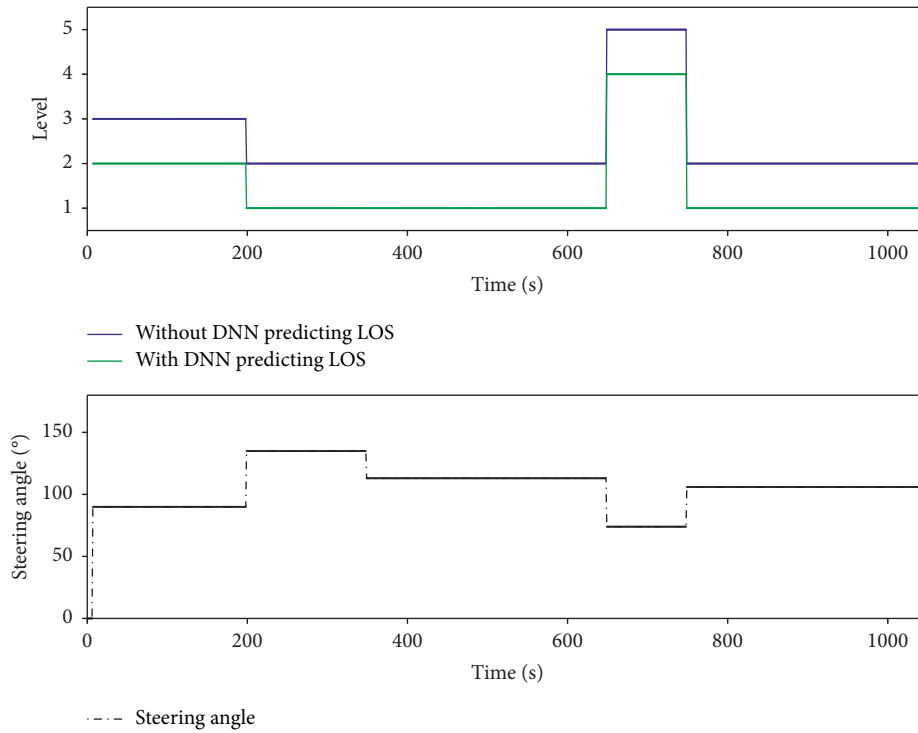


FIGURE 19: The prediction of waypoint behavior effect level.

TABLE 6: Performance index in trajectory tracking.

Vehicle name	Alder (with DNN model)	Alpha (without DNN model)
The average of tracking error	1.75	2.16
The variance of tracking error	12.70	16.43
IAE ( $10^3$ ) ( $\int_0^t  e(t) d\tau$ )	0.685	0.825
ITAE ( $10^4$ ) ( $\int_0^t t e(t) d\tau$ )	4.691	5.499

TABLE 7: Performance indices in waypoint behavior effect evaluation.

Vehicle name	Adjusted by DNN model	Not adjusted
The average of level	1.8	2.8
The variance of level	1.36	1.36

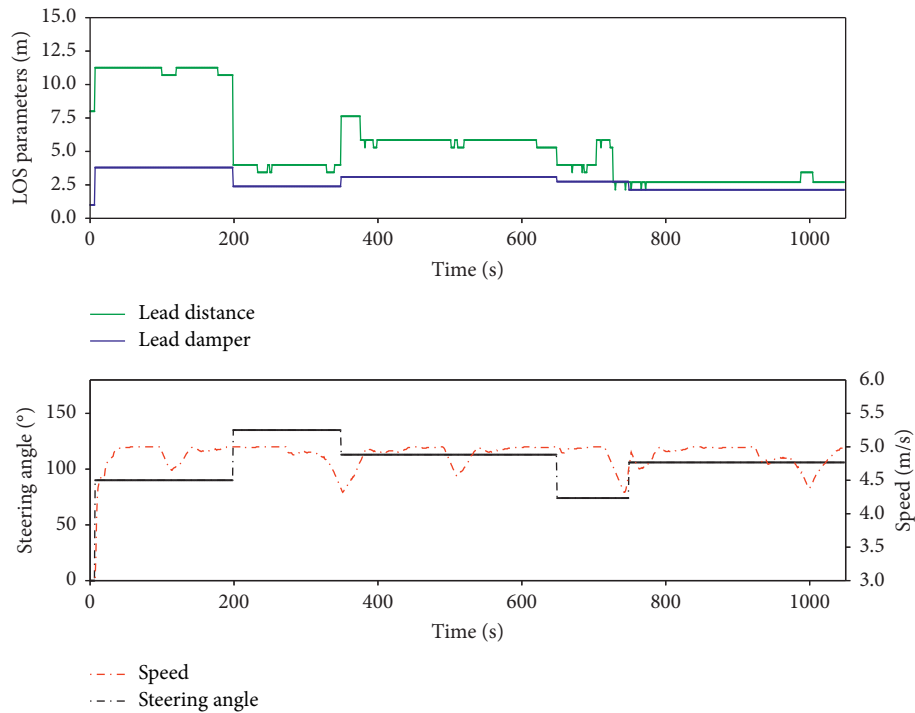
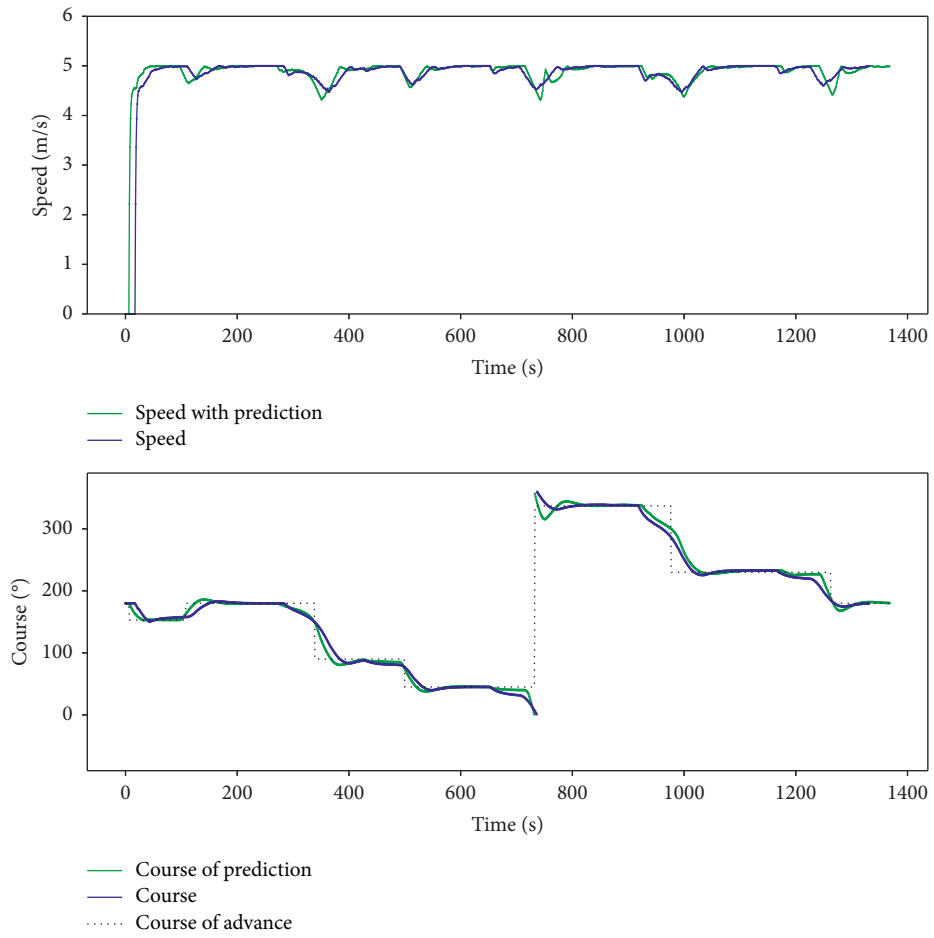


FIGURE 20: The prediction of LOS parameters.

speed and course of USV alpha without prediction model, and the black dashed line is the course of advance. It can be seen that there is little difference between the changes of the two vehicles' speed and course, which indicates that the intervention of waypoint behavior by using DNN does not cause rapid increase or decrease for speed and course. The method does not have too much influence on the control loop. It is reliable.

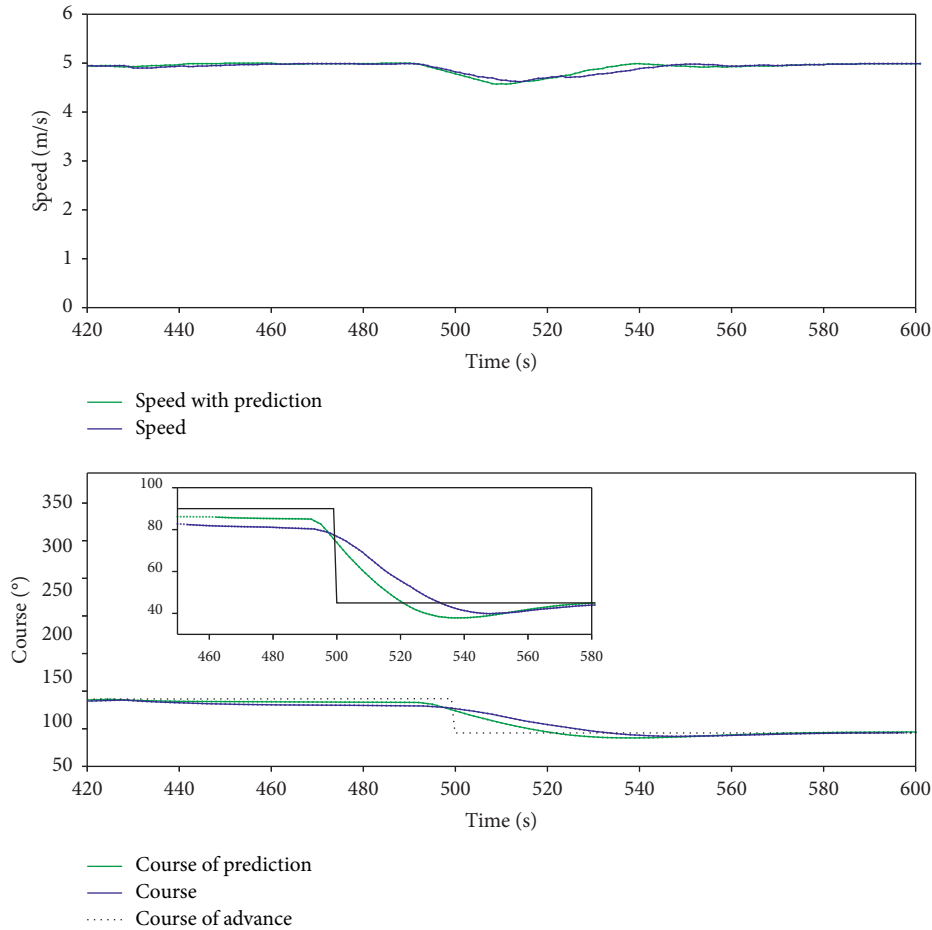
From the above results in Section 4, we can make a summary:

- (1) The ability of DNN model for trajectory tracking control based on GB-RBM optimization to evaluate the waypoint behavior effects has been greatly improved compared with the previous model. The accuracy of the test set has been improved by 5%, reaching 88.9%.
- (2) After correction, the average trajectory tracking error of USV is reduced by 19.0%, and the waypoint behavior effect level has been raised by one level.



(a)

FIGURE 21: Continued.



(b)

FIGURE 21: The changes of speed and course: (a) the changes in entire route; (b) the changes in the process of the second steering.

- (3) The DNN prediction model which is applied to the trajectory tracking control of USV can evaluate the waypoint behavior effect before steering and adjust the parameters of guidance law in real time.

## 5. Conclusions

In this study, two prediction models based on DNN have been constructed, i.e., “DNN-1: waypoint behavior effect evaluation model” and “DNN-2: real-time LOS parameter valuation model.” The models are connected in parallel with the LOS guidance process for trajectory tracking of USV, which improves the effect of trajectory tracking obviously. The experimental results have demonstrated the positive effect of deep learning method on autonomous navigation of USV. The DNN has learned the mapping relationship between different features and effect levels in the waypoint behavior through the training of dataset. Through the real-time prediction of LOS parameters,

the trajectory tracking error is reduced by about 1 times length of the vehicle.

We have developed a new MOOS application. To our knowledge, this is the first time that DNN is integrated dynamically into MOOS-IvP, a well-known marine autonomous platform. Although it only predicts the guidance process of USV, the enhancement is significant. In the future, it can further improve the overall capability of USV’s self-driving.

In addition, a dual-DNN model has been in collaboration with the prediction of trajectory tracking control process, which is also our first attempt. The experimental results have proved its feasibility, and we believe that the way of evaluating the waypoint behavior effect firstly and then executing the maneuvering according to the prediction is more in line with the regular mode of ship maneuvering and control in marine domain. It will be beneficial to improve the reliability of real-world scenarios.



## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare no conflicts of interest.

## References

- [1] S. Campbell, W. Naeem, and G. W. Irwin, "A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres," *Annual Reviews in Control*, vol. 36, no. 2, pp. 267–283, 2012.
- [2] L. E. Kavraki and S. M. LaValle, *Motion Planning*, Springer Handbook of Robotics, Cham, Switzerland, 2016.
- [3] P. Svec, A. Thakur, E. Raboin, B. C. Shah, and S. K. Gupta, "Target following with motion prediction for unmanned surface vehicle operating in cluttered environments," *Autonomous Robots*, vol. 36, pp. 383–405, 2014.
- [4] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*, John Wiley & Sons, Hoboken, NJ, USA, 2011.
- [5] Y. Fan, H. Huang, and Y. Tan, "Robust adaptive path following control of an unmanned surface vessel subject to input saturation and uncertainties," *Applied Sciences*, vol. 9, no. 9, p. 1815, 2019.
- [6] M. R. Benjamin, H. Schmidt, P. M. Newman, and J. J. Leonard, "Nested Autonomy for unmanned marine vehicles with MOOS-IvP," *Journal of Field Robotics*, vol. 27, no. 6, pp. 834–875, 2010.
- [7] W. L. Sun and X. Gao, "Predicting the Trajectory tracking control of unmanned surface vehicle based on deep learning," *Artificial Intelligence in China*, pp. 591–598, 2020.
- [8] J. Velagic, Z. Vukic, and E. Omerdic, "Adaptive fuzzy ship autopilot for track-keeping," *Control Engineering Practice*, vol. 11, no. 4, pp. 433–443, 2003.
- [9] A. P. Aguiar, L. Cremean, and J. P. Hespanha, "Position tracking for a nonlinear underactuated hovercraft: controller design and experimental results," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, vol. 4, pp. 3858–3863, Maui, HI, USA, December 2003.
- [10] C. Lv, H. Yu, J. Chi et al., "A hybrid coordination controller for speed and heading control of underactuated unmanned surface vehicles system," *Ocean Engineering*, vol. 176, pp. 222–230, 2019.
- [11] Y. Xia, K. Xu, Y. Li, G. Xu, and X. Xiang, "Improved line-of-sight trajectory tracking control of under-actuated AUV subjects to ocean currents and input saturation," *Ocean Engineering*, vol. 174, pp. 14–30, 2019.
- [12] H. Huang, M. Gong, Y. Zhuang, S. Sharma, and D. Xu, "A new guidance law for trajectory tracking of an underactuated unmanned surface vehicle with parameter perturbations," *Ocean Engineering*, vol. 175, pp. 217–222, 2019.
- [13] C. Li, Y. S. Zhao, G. Wang, Y. Fan, and Y. Bai, "Adaptive RBF neural network control for unmanned surface vessel course tracking," in *Proceedings of Sixth International Conference on Information Science and Technology*, Vienna, Austria, May 2016.
- [14] R. Skulstad, G. Li, H. Zhang, and T. I. Fossen, "A neural network approach to control allocation of ships for dynamic positioning," *IFAC-PapersOnLine*, vol. 51, no. 29, pp. 128–133, 2018.
- [15] X. Cheng, G. Li, R. Skulstad et al., "Data-driven uncertainty and sensitivity analysis for ship motion modeling in offshore operations," *Ocean Engineering*, vol. 179, pp. 261–272, 2019.
- [16] Y. Shuai, G. Li, X. Cheng et al., "An efficient neural-network based approach to automatic ship docking," *Ocean Engineering*, vol. 191, Article ID 106514, 2019.
- [17] Z. Qiang, Z. Guibing, H. Xin, and Y. Renming, "Adaptive neural network auto-berthing control of marine ships," *Ocean Engineering*, vol. 177, pp. 40–48, 2019.
- [18] Y. Wang, J. Tong, T. Y. Song, and Z. H. Wan, "Unmanned surface vehicle course tracking control based on neural network and deep deterministic policy gradient algorithm," in *Proceedings of Oceans- MTS/IEEE Kobe Techno-Oceans (OTO)*, Port Island, Kobe, May 2018.
- [19] B. Marius, D. D. Testa, D. Dworakowski et al., "End-to-end deep learning for self-driving cars," 2016, <http://devblogs.nvidia.com/deep-learning-self-driving-cars>.
- [20] J. S. Kim, D. J. Kim, S. H. Lee, and C. C. Chung, "Autonomous driving vehicles with unmatched disturbance compensation using deep neural networks," in *Proceedings of 2019 19th International Conference on Control, Automation and Systems*, Jeju, South Korea, October 2019.
- [21] Q. Xu, Y. Yang, C. Zhang, and L. Zhang, "Deep convolutional neural network-based autonomous marine vehicle maneuver," *International Journal of Fuzzy Systems*, vol. 20, no. 2, pp. 687–699, 2018.
- [22] Y. Z. Chen, Y. Y. Shi, and B. S. Zhang, "Optimal control via neural networks: a convex approach," in *to Appear in International Conference on Learning Representations*, Addis Abba, Ethiopia, 2019.
- [23] X. Tan, S. Su, Z. Zuo, X. Guo, and X. Sun, "Intrusion detection of UAVs based on the deep belief network optimized by PSO," *Sensors*, vol. 19, no. 24, p. 5529, 2019.
- [24] R. Eldan and O. Shamir, "The power of depth for feedforward neural networks," *JMLR: Workshop and Conference Proceedings*, vol. 49, pp. 1–34, 2016.
- [25] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [26] S. Goudarzi, M. N. Kama, M. H. Anisi, S. A. Soleymani, and F. Doctor, "Self-organizing traffic flow prediction with an optimized deep belief network for internet of vehicles," *Sensors*, vol. 18, Article ID 18103459, 2018.
- [27] W. Zhao, L. Xu, J. Bai, M. Ji, and T. Runge, "Sensor-based risk perception ability network design for drivers in snow and ice environmental freeway: a deep learning and rough sets approach," *Soft Computing*, vol. 22, no. 5, pp. 1457–1466, 2018.
- [28] T. Yamashita, M. Tanaka, E. Yoshida, Y. Yamauchi, and H. Fujiyoshi, "To be Bernoulli or to be Gaussian, for a Restricted Boltzmann Machine," in *Proceedings of 2014 22nd International Conference on Pattern Recognition*, Stockholm, Sweden, August 2014.
- [29] M. R. Benjamin, P. Newman, H. Schmidt, and J. J. Leonard, "An overview of moos-ivp and a brief users guide to the ivp helm autonomy software," vol. 28, Technical Report, MIT Department of Mechanical Engineering, Cambridge, MA, USA, 2009.
- [30] L. Y. Dong and H. L. Xu, "Design of heading control system for USV based on MOOS-IvP," in *Proceedings of the IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chengdu, China, December 2017.
- [31] N. Dula and D. Vladimir, "Command Filtered backstepping design in MOOS-IvP helm framework for trajectory tracking

- of USVs,” in *Proceedings of the American Control Conference*, Marriott Waterfront, Baltimore, MD, USA, July 2010.
- [32] M. R. Benjamin, M. Defilippo, P. Robinette, and M. Novitzky, “Obstacle avoidance using multiobjective optimization and a dynamic obstacle manager,” *IEEE Journal of Oceanic Engineering*, vol. 44, no. 2, pp. 331–342, 2019.
- [33] F. A. Papoulias, “Bifurcation analysis of line of sight vehicle guidance using sliding modes,” *International Journal of Bifurcation and Chaos*, vol. 01, no. 4, pp. 849–865, 1991.