

Article

Hyperparameter Optimization for the LSTM Method of AUV Model Identification Based on Q-Learning

Dianrui Wang¹, Junhe Wan² , Yue Shen¹ , Ping Qin^{1,*} and Bo He¹

¹ School of Information Science and Engineering, Ocean University of China, Qingdao 266000, China; dianrui314@163.com (D.W.); shenyue@ouc.edu.cn (Y.S.); bhe@ouc.edu.cn (B.H.)

² Institute of Oceanographic Instrumentation, Qilu University of Technology (Shandong Academy of Sciences), Qingdao 266000, China; wan_junhe@qlu.edu.cn

* Correspondence: appletsin@ouc.edu.cn

Abstract: An accurate mathematical model is a basis for controlling and estimating the state of an Autonomous underwater vehicle (AUV) system, so how to improve its accuracy is a fundamental problem in the field of automatic control. However, AUV systems are complex, uncertain, and highly non-linear, and it is not easy to obtain through traditional modeling methods. We fit an accurate dynamic AUV model in this study using the long short-term memory (LSTM) neural network approach. As hyper-parameter values have a significant impact on LSTM performance, it is important to select the optimal combination of hyper-parameters. The present research uses the improved Q-learning reinforcement learning algorithm to achieve this aim by improving its recognition accuracy on the verification dataset. To improve the efficiency of action exploration, we improve the Q-learning algorithm and choose the optimal initial state according to the Q table in each round of learning. It can effectively avoid the ineffective exploration of the reinforcement learning agent between the poor-performing hyperparameter combinations. Finally, the experiments based on simulated or actual trial data demonstrate that the proposed model identification method can effectively predict kinematic motion data, and more importantly, the modified Q-Learning approach can optimize the network hyperparameters in the LSTM.

Keywords: autonomous underwater vehicle; model identification; long short term memory; hyperparameter optimization; reinforcement learning; Q-learning



Citation: Wang, D.; Wan, J.; Shen, Y.; Qin, P.; He, B. Hyperparameter Optimization for the LSTM Method of AUV Model Identification Based on Q-Learning. *J. Mar. Sci. Eng.* **2022**, *10*, 1002. <https://doi.org/10.3390/jmse10081002>

Academic Editor: Myung-Il Roh

Received: 12 May 2022

Accepted: 16 July 2022

Published: 22 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The ocean occupies most of the earth's surface and is an important area for commercial activities, scientific research, and resource extraction, and its impact is critical to all aspects [1,2]. Over the past half-century, oceanographic research has shown that the oceans and seafloor are critical to understanding the planet. Exploring the marine environment provides valuable knowledge for many areas of science and engineering. AUVs have been widely used in marine engineering due to their unique advantages. Autonomous underwater vehicles (AUVs) equipped with various sensors have broad applications in scientific, military, and commercial missions, such as deep-sea exploration, cable/pipe tracking, feature tracking, and more [3].

AUVs are complex nonlinear coupled systems, and it is challenging to model them accurately [4,5]. Therefore, studying the non-linear mechanism of AUV systems and finding mathematical methods that can accurately express them have become the focus of AUV research, which has essential educational and economic value [6–9]. Neural network models have recently been widely used in AUV system model identification. In the identification networks, recursive structures are adapted to acquire dynamic information and improve the communication between neurons [10]. The experimental results show that the method can fit the AUV system. The weight values of the network were tuned using a hybrid algorithm

of the genetic algorithm (GA) and the error backpropagation algorithm (BP) [11]. Moreover, the neural network with the hybrid learning algorithm improves the learning speed of convergence and identification accuracy. An approximation of the lumped disturbance and estimation of the parametric uncertainty is achieved by a dynamic neural network [12]. The Bayesian network is used to construct a state observer for robot control when both the actuator and sensor models are used [13]. A multi-scale attention-based long short-term memory (LSTM) model is adopted to identify the ship's non-linear model under different ocean conditions. The experiment results show that the LSTM model has higher prediction accuracy than the traditional support vector regression (SVR) and radial basis function (RBF) model [14]. Ref. [15] also shows that the LSTM method has a high quality of time-series prediction and has important practical applications. Therefore, this study uses an LSTM neural network approach for AUV system identification.

Neural networks have good non-linear mapping ability and have been widely used in system identification, especially in non-linear systems [16–18]. The value of the neural network hyperparameters can significantly affect its performance [19]. However, its value has no theoretical guidance, and different systems often require different values. Therefore, it is of great practical significance to study a method for automatic optimization of hyperparameters of system identification algorithm [20,21].

In response to the above problems, some optimization methods of neural network model structure based on reinforcement learning have been proposed. Meta-modeling based on reinforcement learning enables automated generation of high-performing convolutional neural network (CNN) architectures for a given learning task [22]. A novel multi-objective reinforcement learning method is proposed for hyperparameter optimization to solve the limitations in the actual environment, such as latency and CPU utilization [23]. A context-based meta-RL approach is used to maximize the accuracy of the validation set [24]. It can tackle the data-inefficiency problem of hyperparameter optimization. The above methods have proved the effectiveness of reinforcement learning in hyperparameter optimization. If the hyperparameter optimization problem of a neural network is regarded as a reinforcement learning problem of continuous state-action space, such as deep Q network (DQN) [25] or proximal policy optimization (PPO) [26], the algorithm needs to go through hundreds or even thousands of times of learning to ensure convergence. When the computing resources are minimal, it is not suitable for the situation where the search space of parameters is ample or the performance evaluation of the algorithm is very expensive. Therefore, the Q-learning [27] algorithm is selected in this paper to solve the hyperparameter optimization problem of the LSTM neural network.

Our main objective in this paper is to identify the AUV model and optimize hyperparameters using an improved Q-learning LSTM neural network method. The main contributions of this paper lie in the following three points: (1) We adopt the LSTM method to identify the AUV dynamic model and conduct an in-depth analysis of its principle; (2) Reinforcement learning framework to solve the hyperparameter optimization problem of the LSTM method; (3) The method's effectiveness in this paper is verified by verifying the actual AUV dataset.

The rest of the paper is organized as follows: Section 2 provides the AUV's hydrodynamic model and force analysis. Section 3 describes in details of the proposed method. The experimental results are presented and discussed in Section 4. Finally, Section 5 gives the main conclusions and discusses future work.

2. AUV Model Analysis

2.1. AUV Hydrodynamic Model

This research focuses on the "Sailfish" 210 AUV developed by the Underwater Vehicle Laboratory of Ocean University of China (Qingdao, China), as shown in Figure 1. The cabin structure comprises four sections: the bow, the navigation cabin, the electronic energy cabin, and the propulsion system cabin. A unified electrical and mechanical interface is used between the cabins, which is beneficial for us to configure different loads for the AUV for

different task requirements. The bow is generally equipped with an underwater acoustic communication machine; the navigation cabin includes attitude and heading reference system (AHRS), global positioning system (GPS), doppler velocity log (DVL), etc. The electronic energy cabin includes batteries, industrial computer systems, etc.; the propulsion system cabin mainly includes steering gear and thruster motors. The experiments carried out in this paper are based on the "Sailfish" 210 AUV platform, which has a maximum speed of 5 kn (2.5 m/s). The parameters are shown in Table 1.

Table 1. The parameters of the AUV model.

Definition	Symbol	Unit	Numerical Value
Mass	m	kg	73
Diameter	d	mm	210
Length	l	m	2.2
Center of gravity	x_G, y_G, z_G	mm	0, 0, 0.02
Center of buoyancy	x_B, y_B, z_B	mm	0, 0, 0
Moment of inertia	I_x, I_y, I_z	kg·m ²	0.4, 24.5, 24.5

The general motion of the AUV was described with two coordinate systems, body-fixed reference frame ($G - xyz$) and earth-fixed reference frame ($E - \xi\eta\zeta$) [28]. AUV translational and rotational motions are described in six degrees of freedom (6DOF) as follows:

$$\begin{aligned} \eta &= [\eta_1^T \ \eta_2^T]^T, \quad \eta_1 = [\eta \ \xi \ \zeta]^T, \quad \eta_2 = [\phi \ \theta \ \psi]^T \\ v &= [v_1^T \ v_2^T]^T, \quad v_1 = [u \ v \ w]^T, \quad v_2 = [p \ q \ r]^T \\ \tau &= [\tau_1^T \ \tau_2^T]^T, \quad \tau_1 = [X \ Y \ Z]^T, \quad \tau_2 = [K \ M \ N]^T \end{aligned} \quad (1)$$

where η_1 and η_2 refer to the position and orientation of the AUV with respect to the earth-fixed reference frame, v denotes the translational and rotational speeds with respect to the body-fixed reference frame, τ_1 and τ_2 refer to the external forces and moments with respect to the body-fixed reference frame. A diagram of the AUV coordinate system is shown in Figure 1.

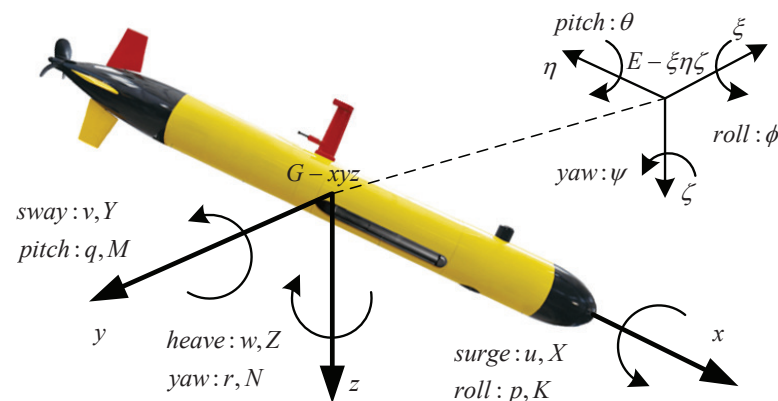


Figure 1. AUV body-fixed and earth-fixed coordinate systems.

Translational velocities $[u \ v \ w]$ are converted from linear velocities $[\dot{\xi} \ \dot{\eta} \ \dot{\zeta}]$ by T_1 :

$$T_1 = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (2)$$

Then,

$$\begin{bmatrix} \ddot{\xi} \\ \dot{\eta} \\ \ddot{\zeta} \end{bmatrix} = T_1 \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (3)$$

The angular rates $[\dot{\phi} \ \dot{\theta} \ \dot{\psi}]$ are converted to the rotational velocities $[p \ q \ r]$ by T_2 :

$$T_2 = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta \end{bmatrix} \quad (4)$$

Thus,

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = T_2 \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (5)$$

The general motion is described in $(G - xyz)$ by Equation (6), where the first three equations describe the translation and the last three describe rotation:

$$\begin{cases} m[(\dot{u} - vr + wq) - x_G(q^2 + r^2) + y_G(pq - \dot{r}) + z_G(pr + \dot{q})] = \sum X \\ m[(\dot{v} - wp + ur) - y_G(r^2 + p^2) + z_G(qr - \dot{p}) + x_G(qp + \dot{r})] = \sum Y \\ m[(\dot{w} - uq + vq) - z_G(p^2 + q^2) + x_G(rp - \dot{q}) + y_G(rq + \dot{p})] = \sum Z \\ I_x \dot{p} + (I_z - I_y)qr + m[y_G(\dot{w} + pv - qu) - z_G(\dot{v} + ru - pw)] = \sum K \\ I_y \dot{q} + (I_x - I_z)rp + m[z_G(\dot{u} + wq - vr) - x_G(\dot{w} + pv - uq)] = \sum M \\ I_z \dot{r} + (I_y - I_x)pq + m[x_G(\dot{r} + ur - pw) - y_G(\dot{u} + qw - vr)] = \sum N \end{cases} \quad (6)$$

where:

m : AUV mass.

x_G, y_G, z_G : the position of centre gravity of the AUV.

I_x, I_y, I_z : the moment of inertia of the AUV.

u, v, w : velocities along the x-axis, y-axis, and z-axis of the AUV.

p, q, r : roll angular velocity, pitch angular velocity and yaw angular velocity.

$\dot{u}, \dot{v}, \dot{w}, \dot{p}, \dot{q}, \dot{r}$: linear acceleration and angular acceleration.

$\sum X, \sum Y, \sum Z, \sum K, \sum M, \sum N$: external force and moment.

2.2. The Dynamical Principles AUV Model Identification

The dynamic model mathematically describes the essential law of the interaction between the AUV and the environment, which can well reflect the state transition of the AUV under the action of force. This paper will realize the AUV model identification based on the dynamic model. The external force (moment) exerted on the AUV mainly includes gravity and buoyancy, hydrodynamic force, rudder force, thrust force, etc. [29]. The force analysis and model identification principle are detailed below.

2.2.1. The Static Force

The static force of the AUV is generated by the gravity P and buoyancy B . The center of buoyancy and center of gravity coordinates in the body-fixed frame are $(0, 0, 0)$ and $(0, 0, z_c)$. The component of the static force in the earth-fixed frame is $(0, 0, P - B)$, which can be obtained by converting it to the motion coordinate system by Equation (7):

$$\begin{cases} X = -(P - B) \sin(\theta) \\ Y = -(P - B) \cos \theta \sin \phi \\ Z = -(P - B) \cos \theta \cos \phi \\ K = -ph \cos \theta \sin \phi \\ M = -ph \sin \theta \\ N = 0 \end{cases} \quad (7)$$

where p is the underwater full displacement of AUV and h represents the depth of AUV.

2.2.2. Hydrodynamic Force

The hydrodynamic forces of AUV are usually divided into inertial hydrodynamic forces and viscous hydrodynamic forces, and the interaction between the two is ignored. In infinitely deep, wide and still water, the hydrodynamic forces of the AUV depend only on its motion and is a function of motion parameters $u, v, w, p, q, r, \dot{u}, \dot{v}, \dot{w}, \dot{p}, \dot{q}, \dot{r}$. According to the idea of Taylor expansion, the hydrodynamic forces (X_H, Y_H, Z_H) and the moments (K_H, M_H, N_H) are expanded to obtain the expression of the hydrodynamic forces:

$$\begin{aligned}
 X_H &= [X_{qq}q^2 + X_{rr}r^2 + X_{rp}rp] + [X_{\dot{u}}\dot{u} + X_{vr}vr + X_{wq}wq] + [X_{u|u}|u| + X_{vv}v^2 + X_{ww}w^2] \\
 Y_H &= [Y_{\dot{r}}\dot{r} + Y_{qr}qr + Y_{\dot{p}}\dot{p} + Y_{pq}pq + Y_{p|p}|p|] + [Y_{\dot{v}}\dot{v} + Y_{vq}vq + Y_{wr}wr + Y_{wp}wp] \\
 &\quad + [Y_{r|r}|\frac{v}{|v|}(v^2 + w^2)^{1/2}|r| + Y_{pu}up] + [Y_0u^2 + Y_vuv + Y_{v|v}|v|(v^2 + w^2)^{1/2}] + Y_{vw}vw \\
 Z_H &= [Z_{\dot{q}}\dot{q} + Z_{rr}r^2 + Z_{pp}p^2 + Z_{rp}rp] + [Z_{\dot{w}}\dot{w} + Z_{vr}vr + Z_{vp}vp] + [Z_{qu}uq + Z_{w|w}|w|(v^2 + w^2)^{1/2}|q|] \\
 &\quad + [Z_0u^2 + Z_wuw + Z_{w|w}|w|(v^2 + w^2)^{1/2}] + [Z_{|w|}|u|w| + Z_{ww}|w|(v^2 + w^2)^{1/2}] + Z_{vv}v^2 \\
 K_H &= [K_{\dot{p}}\dot{p} + K_{\dot{r}}\dot{r} + K_{qr}qr + K_{pq}pq + K_{p|p}|p|] + [K_{pu}up + K_{ru}ur + K_{\dot{v}}\dot{v}] + [K_{vq}vq + K_{wp}wp + K_{wr}wr] \\
 &\quad + [K_0u^2 + K_vuv + K_{v|v}|v|(v^2 + w^2)^{1/2}] + K_{vw}vw \\
 M_H &= [M_{\dot{q}}\dot{q} + M_{rr}r^2 + M_{q|q}|q|q| + M_{pp}p^2 + M_{rp}rp] + [M_{\dot{w}}\dot{w} + M_{vr}vr + M_{vp}vp] \\
 &\quad + [M_{qu}uq + M_{w|w}|q|(v^2 + w^2)^{1/2}] + [M_0u^2 + M_wuw + M_{w|w}|w|(v^2 + w^2)^{1/2}] + [M_{|w|}|u|w| + M_{ww}|w|(v^2 + w^2)^{1/2}] + M_{vv}v^2 \\
 N_H &= [N_{\dot{r}}\dot{r} + N_{qr}qr + N_{r|r}|\frac{v}{|v|}r| + N_{\dot{p}}\dot{p} + N_{pq}pq] + [N_{\dot{v}}\dot{v} + N_{vr}vr + N_{vq}vq + N_{wp}wp] \\
 &\quad + [N_{ru}ur + N_{v|r}r|(v^2 + w^2)^{1/2} + N_{pu}up] + [N_0u^2 + N_vuv + N_{v|v}|v|(v^2 + w^2)^{1/2}] + N_{vw}vw
 \end{aligned} \tag{8}$$

2.2.3. Thrust

The thrust generated by the propeller is calculated as follows:

$$X_T = (1 - t)\rho n^2 D^4 K_T \tag{9}$$

where r : the propeller rotational speed; D : the propeller diameter; t : thrust derating factor; ρ : the water density; and K_T : dimensionless thrust coefficient. K_T is a function related to the advance ratio $J = \frac{u(1-w)}{nD}$, which can be approximated as:

$$K_T = k_0 + k_1 J + k_2 J^2 \tag{10}$$

where k_0, k_1, k_2 are a constant coefficients.

$n = \frac{u(1-w)}{DJ}$ can be obtained from the advance ratio formula, and the relevant variables are substituted into Equation (9), the functional relationship between thrust and speed can be obtained with the following equation:

$$X_T = \frac{1}{2}\rho L^2 u^2 (a_T + b_T + c_T) \tag{11}$$

where L is the body length, $a_T = \mu k_2$, $b_T = \mu k_1/J$, $c_T = \mu k_0/J^2$, $\mu = 2(1-t)(1-w)^2 D^2/L^2$.

2.2.4. Rudder Force

This paper discusses the underactuated underwater robot, whose rudder force comes from a pair of horizontal and vertical rudders mounted on its tail. When the rudder moves at speed V and angle of attack α , it will be subjected to two parts of force: lift force perpendicular to the direction of the water flow, and the resistance along the direction of the water flow, the calculation formula is as follows:

$$\begin{aligned} L &= \frac{1}{2} C_L \rho A_R V^2 \\ D &= \frac{1}{2} C_D \rho A_R V^2 \end{aligned} \quad (12)$$

where C_L is the lift coefficient, C_D is the drag coefficient, and A_R is the cross-sectional area of the rudder.

2.2.5. The Principle of Identification

$F_{else} = [X_{else}, Y_{else}, Z_{else}, K_{else}, M_{else}, N_{else}]^T$ is used to represent static force, thrust force, rudder force, and disturbance force. Then, using the superposition principle, we can obtain the AUV force expression:

$$F = F_H + F_{else} \quad (13)$$

Bring Equation (13) into Equation (6), and simplify the equation of motion, we can obtain:

$$E \ddot{X} = F_{vis} + F_{else} \quad (14)$$

where:

$$\begin{aligned} X &= [u, v, w, p, q, r]^T \\ E &= \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 & 0 & mz_G & -my_G \\ 0 & m - Y_{\dot{v}} & 0 & mz_g & 0 & mx_G - Y_{\dot{r}} \\ 0 & 0 & m - Z_{\dot{w}} & my_G & -mx_G - Z_{\dot{q}} & 0 \\ 0 & -mz_G & my_G & I_x - K_{\dot{p}} & 0 & 0 \\ mz_G & 0 & -mx_G - M_{\dot{w}} & 0 & I_y - M_{\dot{q}} & 0 \\ -my_G & mx_G - N_{\dot{v}} & 0 & 0 & 0 & I_z - N_{\dot{r}} \end{bmatrix} \end{aligned} \quad (16)$$

$F_{vis} = [X_{vis}, Y_{vis}, Z_{vis}, K_{vis}, M_{vis}, N_{vis}]^T$ is non-inertial hydrodynamic force.

It can be seen from the above analysis that the acceleration of the AUV is recorded as the combined action of the non-inertial hydrodynamic force and other forces except the hydrodynamic force, denoted as $\ddot{X} = f(u, v, w, p, q, r, n, \delta_r, \delta_s)$. Further, the acceleration results from the non-inertial hydrodynamic force $F_{vis} = f_H(u, v, w, p, q, r)$, the thrust force $F_T = f_T(u, n)$, and the rudder force $F_{L/D} = f_{\delta}(u, \delta_r, \delta_s)$.

Based on the above analysis of the AUV model, the AUV states are denoted as $X_t = [x, y, z, \phi, \theta, \psi, u, v, w, p, q, r]$, then the state changes can be expressed as $\Delta X_t = f_1(\phi, \theta, \psi, u, v, w, p, q, r, n, \delta_r, \delta_s)$. The historical data of AUV implies the causal relationship of its dynamic model and has the characteristics of a hidden Markov model, which can be used to build an AUV data-driven model.

3. AUV Model Identification Method

AUV system identification is essentially a mathematical modeling method. Its primary purpose is to build its mathematical model, which can be used in many aspects, such as controller involvement, system prediction, and system simulation. We can see from the above that the AUV system is a complex and uncertain, highly nonlinear system, and it is not easy to obtain an accurate dynamic model. We aim at this problem by adopting a data-driven model identification method based on LSTM neural network. Further, the Q method is used to optimize its hyperparameters to improve the learning efficiency of the LSTM neural network.

3.1. Fundamentals of System Identification

Identification based on a neural network means that the neural network is directly used to learn the mapping relationship between input and output. The learning criterion minimizes the error between the network's output and the system's actual output. From the above, the goal of learning is to minimize the objective function of error [30], which is as follows:

$$E = \frac{1}{2}(y(t) - y_n(t))^2 \quad (17)$$

where, $y_n(t)$ is the output of the neural network at time t , and $y(t)$ is the actual output of the system at time t . Neural networks can fit any function with arbitrary precision. In principle, the desired output will be obtained as long as there is enough training data and input.

Since the environment is full of time-series information, the information before and after them is related to a certain extent. For example, AUV's position and attitude data are all data sequences that change with time. Additionally, the LSTM [31,32] is good at processing this kind of data information. Its basic structure is shown in Figure 2.

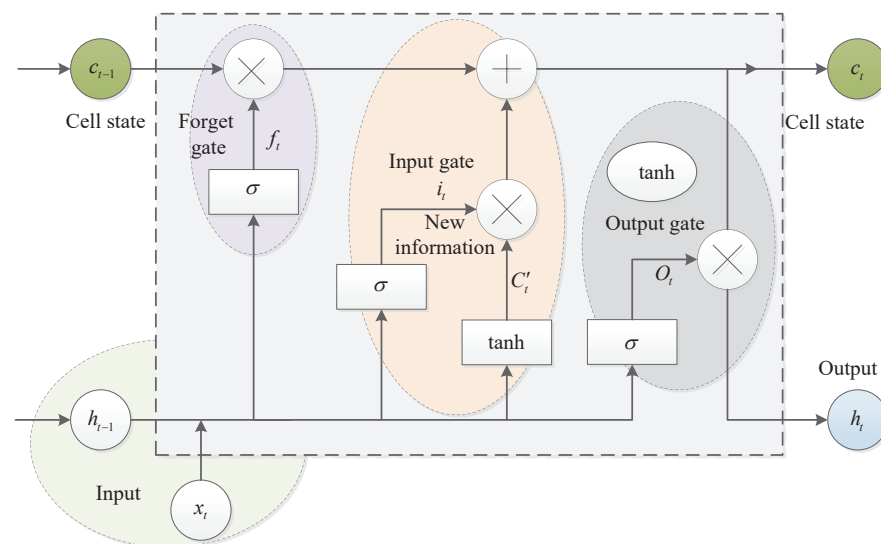


Figure 2. The structure of the LSTM neural network.

The network structure introduces a cell state which contains all the information at the last moment. When new information is encountered, a series of operations will be taken to choose between the old and new information. Coupled with the introduced “memory-forgetting” mechanism, the processing of long-term series data can be realized. The structure mainly includes input, forget, and output gates. The input are h_{t-1} and x_t , the output is h_t , and the cell states are c_{t-1} and c_t .

The forget gate controls the time dependence and effects of previous inputs and determines which states are remembered or forgotten. The output of the forget gate is:

$$o_f = \sigma(w_f x_t + u_f h_{t-1}) c_{t-1} \quad (18)$$

The input gate is also called the selection memory stage. It is to decide the degree of consideration for the current moment. The calculations for each part are as follows:

$$\begin{cases} i_t = \sigma(w_i x_t + u_i h_{t-1}) \\ c'_t = \tanh(w_c x_t + u_c h_{t-1}) \\ c_t = f_t c_{t-1} + i_t c'_t \end{cases} \quad (19)$$

The output gate determines the final output information. The computational procedure is summarized as follows.

$$\begin{cases} O_t = \sigma(w_o x_t + u_o h_{t-1}) \\ h_t = O_t \tanh(C_t) \end{cases} \quad (20)$$

The above is the basic principle of the neural network LSTM, which uses the gated state to selectively memorize the input information to meet memory needs, while forgetting the long-term sequence information.

3.2. Identification Principle and Process

After the actual test of AUV, we can obtain the dataset required for model identification. After removing the invalid data, an input and output model is established according to the navigation control instruction information of the AUV and its posture information. According to Equation (14), we can obtain:

$$\dot{x} = \frac{dx}{dt} = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ E_1 F_1 \\ E_2 F_2 \\ E_3 F_3 \\ x_{10} \\ x_{11} \\ x_{12} \\ E_4 F_4 \\ E_5 F_5 \\ E_6 F_6 \end{bmatrix} \quad (21)$$

where, $\begin{cases} x_1 = x, x_2 = y, x_3 = z, x_4 = u, x_5 = v, x_6 = w, x_7 = \phi, x_8 = \theta, x_9 = \psi, \\ x_{10} = p, x_{11} = q, x_{12} = r \\ F = F(\phi, \theta, \psi, u, v, w, p, q, r, n, \delta_r, \delta_s) \end{cases}$. The subscripts of the matrices E_n and F_n represent the row and column of the matrix, respectively. Next, the learned AUV dynamic model can be expressed as:

$$P(x'_t, u'_t) : (x'_t, u'_t) \rightarrow (y'_{t+1} - y'_t) = \Delta y'_t \quad (22)$$

where u'_t is comprised of the thruster command n_t and rudder angle commands $(\delta_{r_t}, \delta_{s_t})$.

The structure of the LSTM-based AUV model is shown in Figure 3. The input elements of the input layer is $x'_{inp_t} = [x'_t, u'_t]$, and the output of neural network is $x'_{out_t} = [\Delta y'_t]$. The attitude and speed information of the next moment can be obtained by using the control instruction, attitude and speed information of 20 sets of time-series data. The learned model can be optimized according to the set loss function. More details of network architectures are described below.

To avoid inconsistencies due to different relative scale sizes of different features, we normalize the data as follows:

$$f : x \rightarrow x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (23)$$

where $x, x' \in R$, $x_{\min} = \min(x)$, $x_{\max} = \max(x)$.

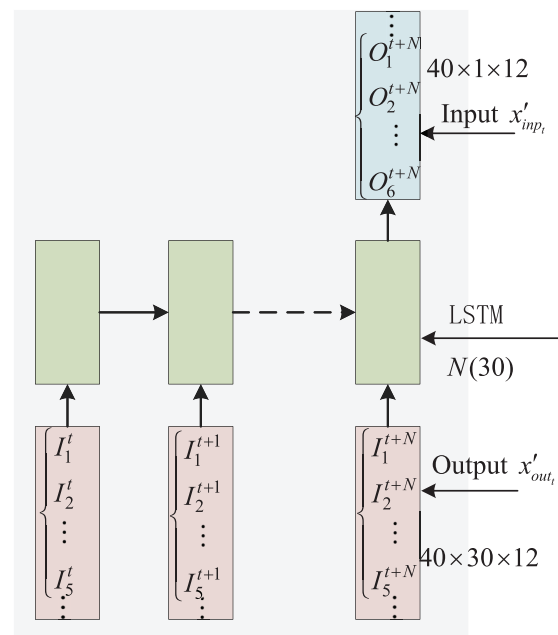


Figure 3. One-step AUV dynamic model based on LSTM.

Model evaluation criteria are mainly used to evaluate the accuracy of the recognition model. In this study, the mean square error between the output of the neural network and the actual output is used as the evaluation index, as follows:

$$MSE = \frac{1}{N} \sum_{n=1}^N [d(n) - y(n)]^2 \quad (24)$$

where $d(n)$ is the output of the neural network, $y(n)$ is the system's actual output, and N is the number of datasets calculated at one time.

For the system identification of the AUV in this study, the model is trained by minimizing the error $M = \frac{1}{|D|} \sum_{(I, \Delta y') \in D} \frac{1}{2} \|(\Delta y'_t - O)\|^2$, where $O = P(I)$, the training dataset D consists of input–output training pairs $I : (x', u') \rightarrow O : (\Delta y)$. The multi-step error is adopted to test the effectiveness of the learned dynamic model, as shown in Equation (25).

$$M^H = \frac{1}{|D_H|} \sum_{(O', u'_t, \Delta y') \in D_H} \frac{1}{H} \sum_{h=1}^H \frac{1}{2} \|(\Delta y - P((O'_{4-12}, u'_t)))\|^2 \quad (25)$$

where O'_{4-12} represents the desired input data $(\phi, \theta, \psi, u, v, w, p, q, r)$ from the first four columns of the output data O' obtained in the previous step. Additionally, the training dataset D_H consists of input–output data pairs $I : (O'_{4-12}, u') \rightarrow O : (\Delta y)$. To sum up, the learned multi-step model can update the AUV state $(x, y, z, \phi, \theta, \psi, u, v, w, p, q, r)$ in a cyclic manner using only action instructions u' .

3.3. Hyperparameter Optimization for Identification Algorithm

3.3.1. MDP Modeling of LSTM Hyperparameter Optimization Problems

The performance of the LSTM algorithm is highly dependent on hyperparameters. Moreover, different tasks often require different hyperparameter configurations. To achieve high-precision identification of the AUV system, we adopt reinforcement learning to optimize the hyperparameter configuration of the LSTM network.

Reinforcement learning is one of the many categories of machine learning methods in which the best/suboptimal strategy is determined by interacting with dynamic environments [33]. A Markov Decision Process (MDP) includes five elements: $M = \langle S, A, R, T, \gamma \rangle$: where:

S : the set of possible states.

A : the set of actions generated by the policy.

R : reward model.

T : dynamics model, the probability of reaching the next state with the current state and action.

γ : discount factor (between 0 and 1).

Figure 4 shows the basic structure of reinforcement learning. In this way, the agent learns how to map states to actions. At time t , the agent receives state s_t and produces action a_t , then transitions to the next state s_{t+1} and obtains reward r_t . The process does not stop until the final condition is reached. To improve the optimization efficiency, the hyperparameter optimization of the LSTM neural network is regarded as a reinforcement learning problem with discrete state space and discrete action space. Then, the above elements are designed for this problem.

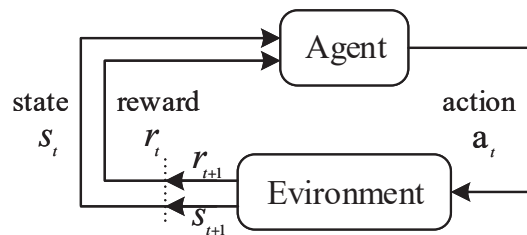


Figure 4. The basic structure of reinforcement learning.

Action space: The hyperparameters to be optimized and the candidate values of each hyperparameter are determined according to the LSTM network model structure. We concentrate on the number of neurons N_{n1} and N_{n2} in the hidden layers, batch size N_b and time step N_{ts} in this study, while other settings are determined empirically. Therefore, the hyperparameters to be optimized and their candidate values are shown in Table 2.

Table 2. The optimized hyperparameters and variation ranges.

Hyperparameter	Variation Range
The number of neurons (1)	(50, 100, 150, 200)
The number of neurons (2)	(50, 100, 150, 200)
Batch size	(8, 16, 32, 64)
Time step	(5, 10, 15, 20)

State space: In this paper, the current hyperparameter configuration a of the network is taken as the state s at time t . Then, the state space S is the same as the action space A , that is, $(N_{n1}, N_{n2}, N_b, N_{ts})$.

Reward: It can be seen from the previous analysis that the error of identification decreases with a decreasing output error. So we define the immediate reward as the root mean square error of test sets with Equation (26).

$$r = -\frac{1}{N_t} \sum_{n=1}^{N_t} [d_t(n) - y_t(n)]^2 \quad (26)$$

where N_t is the number of datasets, d_t , and y_t present the output of network and the real output on the testing set, respectively.

3.3.2. Hyperparameter Optimization Method

The hyperparameter optimization problem of the LSTM neural network is defined as a reinforcement learning problem with discrete state space and action space. Reinforcement learning methods for value function or policy approximation through neural networks, such as DQN, are also suitable for solving reinforcement learning problems with discrete action spaces. However, this kind of algorithm needs much learning to ensure convergence,

and it is not suitable for these methods when the computing resources are very limited. Thus, the Q-Learning algorithm is selected in this paper to solve the problem.

Q-Learning is a temporal difference algorithm designed to solve the reinforcement learning problem. The optimal action policy $\pi^* : s_t \rightarrow a_t$ can be obtained by maximizing the action value Q function $Q(s_t, a_t)$, which reflects the long-term impact of an action. The Q function is updated according to Equation (27):

$$Q(s_t, a_t) = (1 - \alpha)Q(s_t, a_t) + \alpha \left[r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) \right] \quad (27)$$

where $0 < \alpha < 1$ is a learning rate, $0 < \gamma < 1$ is the discount factor.

The structure of the LSTM hyperparameter optimization method based on Q-Learning is shown in Figure 5. The hyperparameter optimization process is as follows: first, the Q-value table is initialized with zeros, and the initial hyperparameter configurations (that is, initial state s_0) are randomly selected in the action space. Then, the action a_t is chosen according to the ϵ -greedy selection rule. Next, we perform the new action a_t and the system acquires a new state s_{t+1} and reward r_{t+1} . Finally, the Q-value table is updated according to the formula. The round ends when the termination conditions are met.

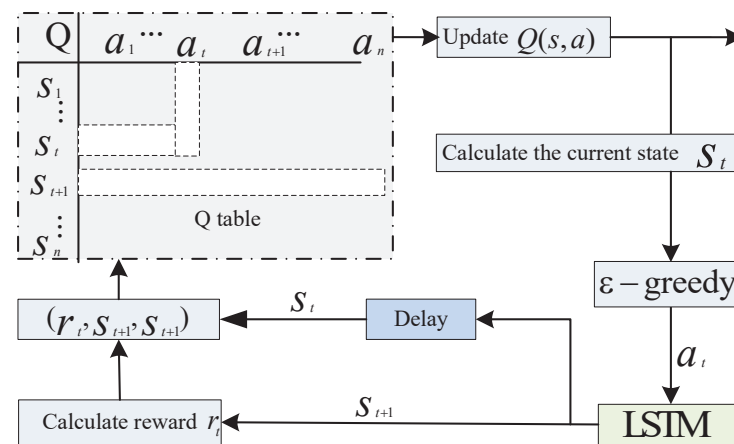


Figure 5. The Structure of hyperparameter optimization method based on Q-Learning.

To improve the optimization efficiency of hyperparameters, we improve the above methods. In each round of learning, in addition to randomly selecting the initial state in the first round, the agent selects the current best hyperparameter configuration as the initial state to start optimization. It can effectively avoid the ineffective exploration of the reinforcement learning agent between the poor-performing hyperparameter combinations.

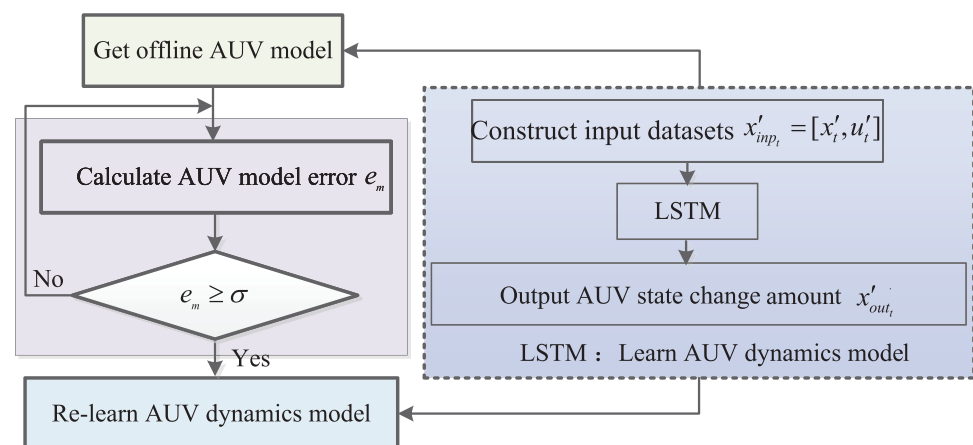
3.4. Identification Algorithm

After introducing the model identification method based on LSTM and the hyperparameter optimization method based on improved Q-learning, we can obtain the complete process and framework of the method in this paper. The algorithm flow of hyperparameter optimization for the LSTM method of AUV model identification based on Q-Learning is shown in Table 3.

Table 3. AUV Model Identification Algorithm.

AUV Model Identification Algorithm
Given a training datasets, including input and output datasets
do for episode in 1 to count
Network initialization: Initialize neural network hyperparameters
do for t in 1 to T
Choose optimized hyperparameters
Calculate network output
Calculate the error between the true output and the output of the network
Update neural network
Update Q table
end for
end for

Figure 6 shows a detailed system model identification block diagram. Offline models can be obtained by offline training through real historical data. During the actual sailing, the system will regularly check the accuracy of the learned dynamics model. If the error is greater than σ pre-determined empirically and does not meet the requirements, the model will be retrained based on new real-time data.

**Figure 6.** Block diagram of AUV model identification.

4. Results

In order to verify the effectiveness of the proposed method, numerous experiments have been performed. The experiments were divided into two parts based on simulation data (see Section 4.1) and real data (see Section 4.2). Simulations were run on the model described in Section 2. The real data were acquired by the Sailfish AUV.

4.1. Results on Simulation Data

First, the validity of AUV system identification based on the LSTM neural network is verified. A set of hyperparameters is randomly set for the LSTM neural network, and a system identification experiment is carried out for the above AUV simulation system. In total, 4990 input and output data pairs are used for model identification, of which 4000 sets of data are used as training sets to train the LSTM neural network. The other 990 sets of data are used as validation sets to test the recognition effect of the model. The number of training is set to 100.

The change curve of the loss function during training is shown in Figure 7. As shown in the figure, the loss curve converges during training, and the value of the loss function decreases with iterations. From the perspective of the convergence of the loss curve, the application of the LSTM neural network can effectively realize the identification of the AUV system.

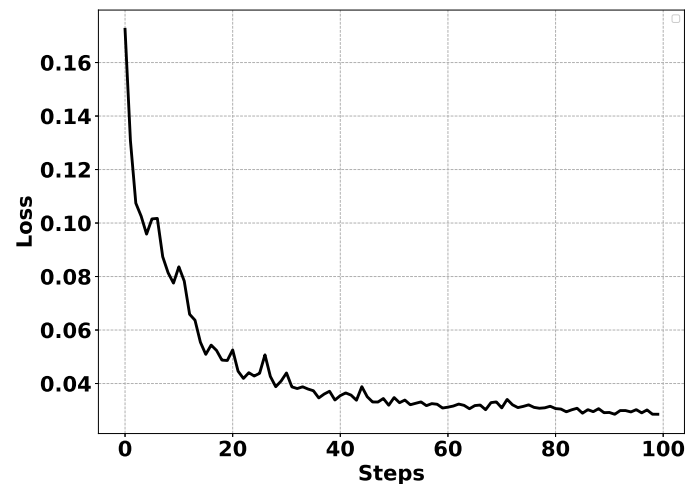


Figure 7. The result of loss before optimization.

After 100 times of training, the validation dataset is brought into the network model to test. After calculation, the squared sum of the output error is 8.490711, and the mean and variance of the absolute value of the error are 0.1761263 and 0.0145471, respectively. We can see that the output of the LSTM network model can fit the actual output of the system after 100 times of training. However, the error between the network output and the real output is large. The fitting curve between the network output and the real output will be shown in the comparison results of different methods later.

The large output fitting error of the above LSTM neural network model is because the hyperparameter settings of the model are not suitable. The identification accuracy is greatly affected by the network hyperparameter settings. Inappropriate hyperparameter settings may even lead to non-convergence. In order to achieve high-precision system identification, a reinforcement learning algorithm is used to optimize the selection of the hyperparameters of the above neural network.

Data were divided into training and test sets during each test, yielding a training set of 4000 and a test set of 990. The proposed method was performed for 100 episodes. To demonstrate the optimization performance of the method, we compare the results of five different stages (proc1-0th episode, proc2-25th episode, proc3-50th episode, proc4-75th episode, and proc5-100th episode).

We can see from the above results that the five optimized LSTM neural network models can make the loss curve converge quickly, shown in Figure 8. It can be also noticed that the convergence speed increases as the episode increases, and this method has prominent optimization characteristics.

The statistical results of the output errors of the five groups of LSTM neural network models on the validation set are recorded in Table 4. In order to make the trend of MSE more apparent, it is enlarged 1000 times and displayed in Figure 9. As can be seen from the graph, the MAE, MSE and RMSE of the error of making predictions on the validation set decrease as the number of episodes increases. The above results also demonstrate the performance of the method from a statistical point of view.

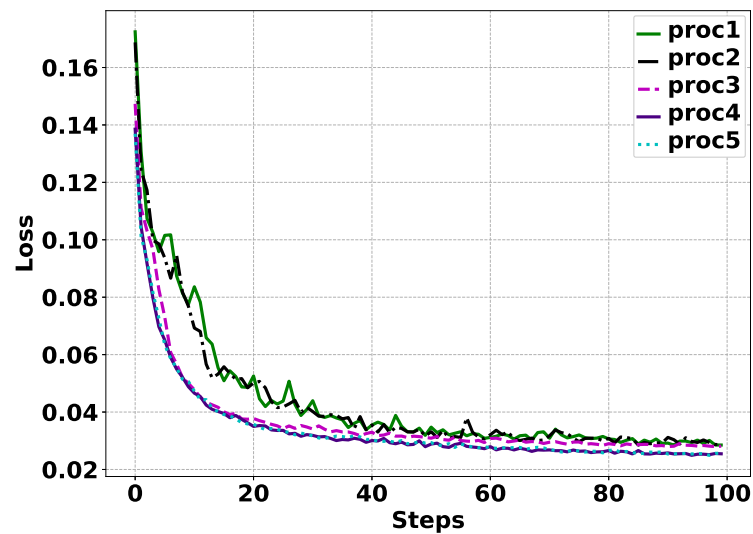


Figure 8. Loss results at different stages.

Table 4. Results of LSTM neural network hyperparameter optimization.

Episode	Hyperparameter Combination	Mean Absolute Error	Mean Squared Error	Root Mean Square Error
0	(50, 50, 8, 10)	0.001784	7.167763×10^{-6}	0.002677
25	(100, 100, 32, 15)	0.001693	7.037789×10^{-6}	0.002652
50	(100, 150, 8, 20)	0.001503	6.737789×10^{-6}	0.002302
75	(150, 50, 32, 15)	0.001304	6.397751×10^{-6}	0.002129
100	(50, 100, 8, 10)	0.001277	4.396948×10^{-6}	0.002096

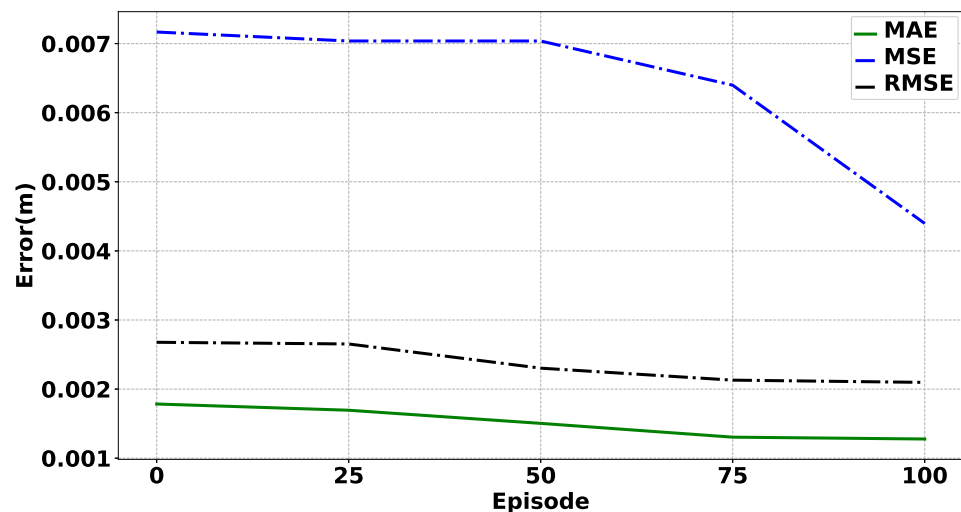


Figure 9. Error results at different stages.

In order to further illustrate the advantages of this method, we compared the effect of the method before and after optimization with the commonly used DR algorithm. We compare the prediction results on the validation set optimized after 0 episodes and 100 episodes with the results predicted by the DR algorithm. The fitting curves of the network output and the real output of the AUV's position, linear velocity, angle, and

angular velocity are shown in Figures 10–13, respectively. It can be seen from the figure that the optimized LSTM model makes the output of the 12 variables of the identification model have better agreement with the actual system output. In order to more clearly reflect the effectiveness of the method proposed in this paper, the MAE, MSE, and RMSE indicators of the prediction error are shown in Table 5. Compared with the unoptimized approach and DR, our method shows superior performance, with less MAE (28.42%, 29.13%), MSE (38.66%, 38.96%), and RMSE (21.70%, 25.81%). It can be seen that the deviation between the output of our method and the actual system is smaller than that of the other two methods.

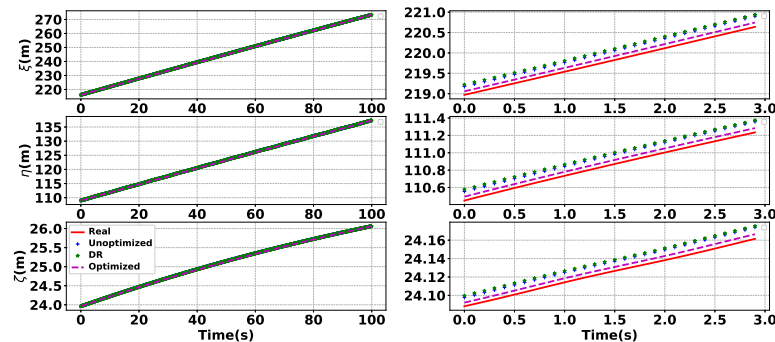


Figure 10. Identification result of AUV position.

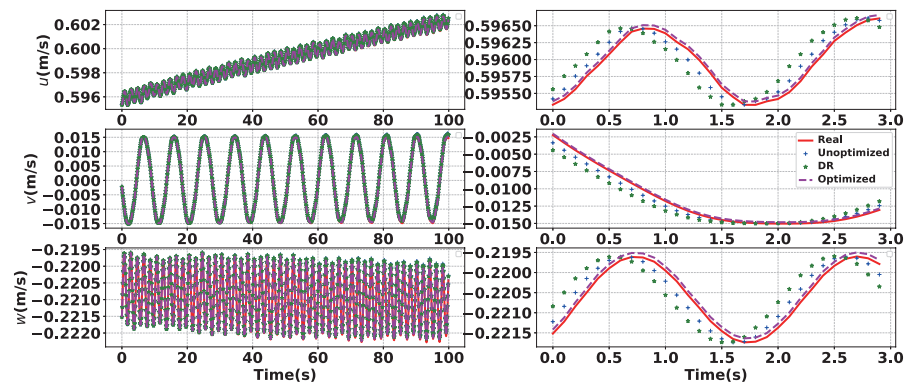


Figure 11. Identification result of AUV velocity.

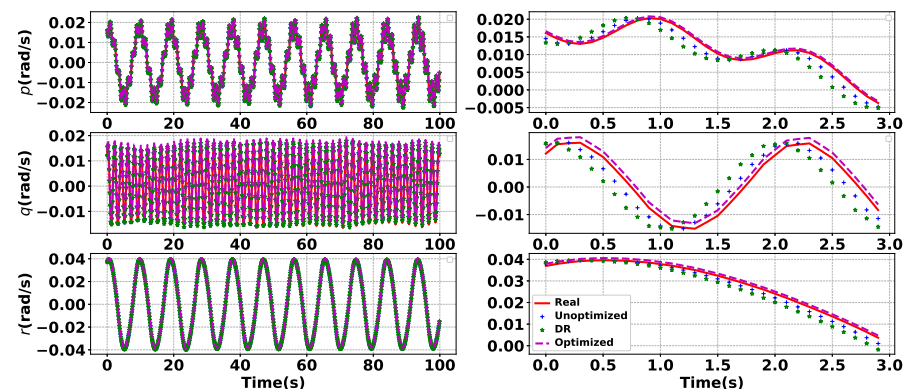


Figure 12. Identification result of AUV angular velocity.

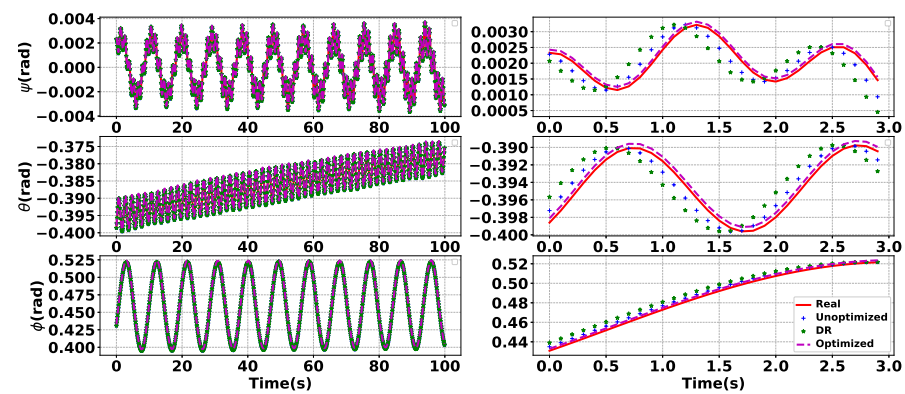


Figure 13. Identification result of AUV attitude angle.

Table 5. Error comparison of different methods.

Method	MAE	MSE	RMSE
Unoptimized	0.001784	7.167763×10^{-6}	0.002677
DR	0.001802	7.203546×10^{-6}	0.002825
Optimized method	0.001277	4.396948×10^{-6}	0.002096

4.2. Results on Real Data

The AUV dataset is required for training and validating the LSTM neural network model identification method. Therefore, the experiments of data collection should be carried out first. The experiments were carried out on the Sailfish AUV, as shown in Figure 14.

The actual data includes the sensor's noise in the acquisition process, and it is more difficult to obtain an accurate model than the simulation. Therefore, the method performed 200 episodes. The experimental dataset is divided into training and testing sets, yielding a training set of 4000 and a test set of 990. From the loss curve of the LSTM neural network in the training process, this method has prominent learning characteristics, as shown in Figure 15.



Figure 14. Sailfish AUV platform during the experiment.

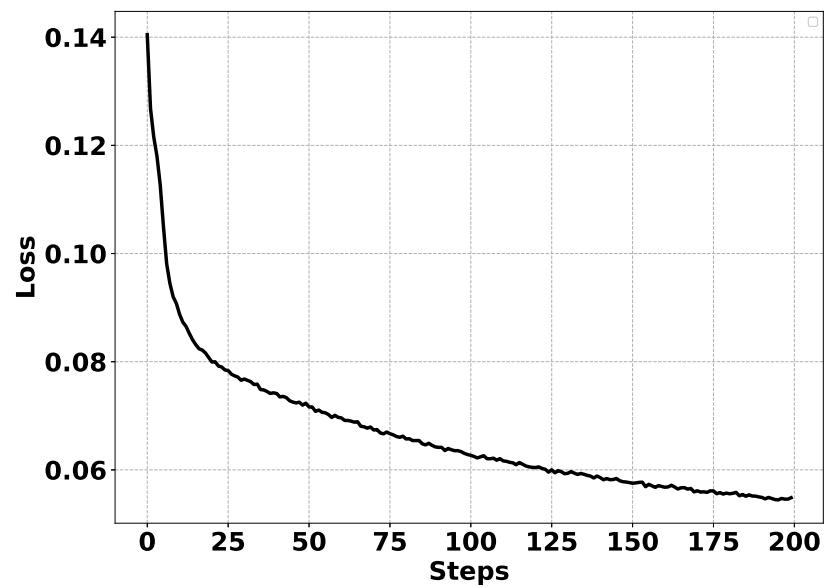


Figure 15. The training loss curve of the proposed LSTM neural network.

The fitting curve between the network output and the real output is shown in Figures 16–19. The calculated MSE of the output error is 550.830181, and the MAE and RMSE values are 6.250734 and 23.469771, respectively. We can see that the LSTM neural network obtained by optimization can fit the system's actual output and achieve high-precision recognition of AUVs. In order to illustrate the superiority of the method, it is compared with the commonly used dead reckoning (DR) method. The results are shown in Table 6, the proposed method provided 64.90% higher MAE, 64.20% higher MSE, and 37.76% higher RMSE than the DR method.

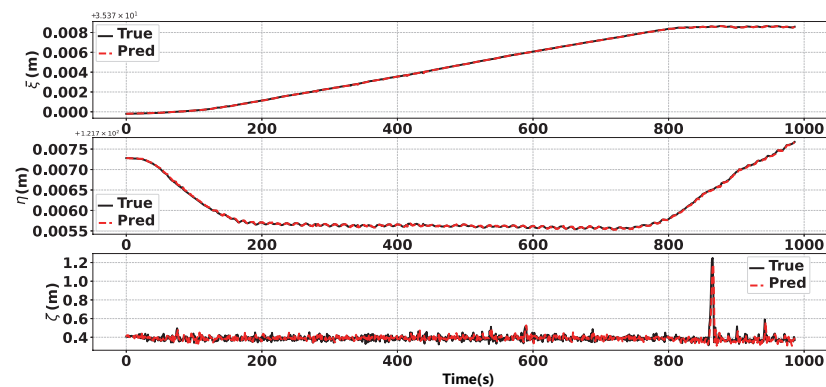


Figure 16. Identification result of AUV position.

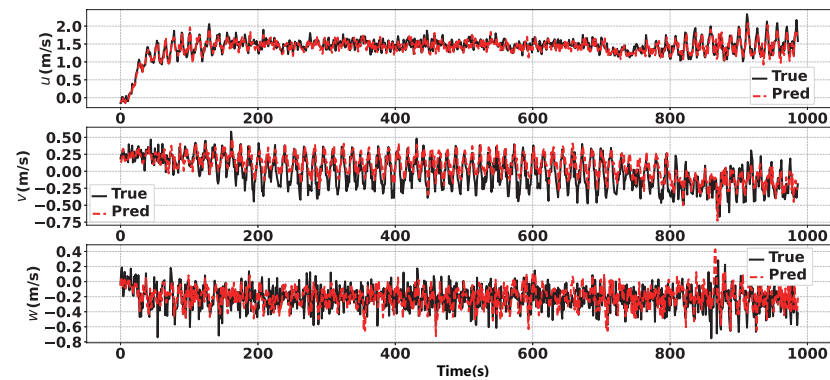


Figure 17. Identification result of AUV velocity.

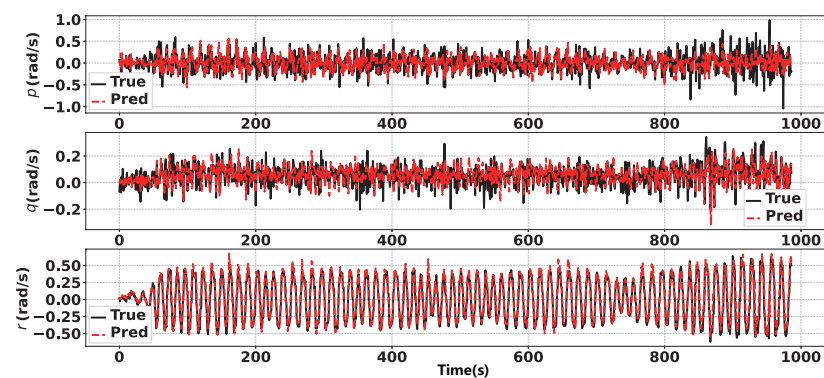


Figure 18. Identification result of AUV angular velocity.

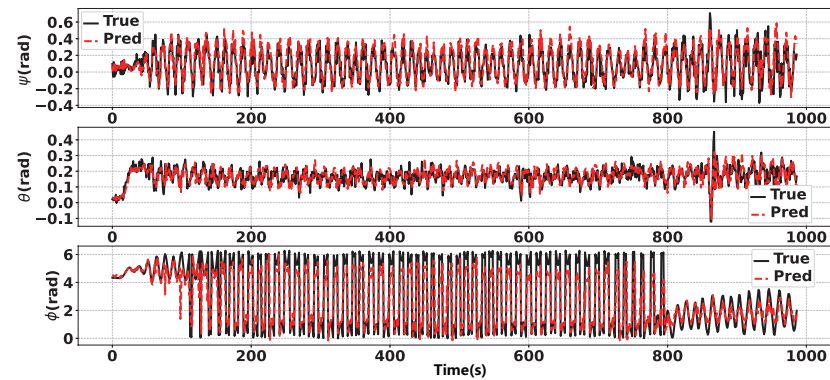


Figure 19. Identification result of AUV attitude angle.

Table 6. Error comparison of different methods.

Method	MAE	MSE	RMSE
Proposed method	6.250734	206.830181	23.469771
DR	17.808035	550.071027	37.705304

Due to the small prediction bias for the AUV motion state, It turns out, yet again, that the proposed method has high predictive power. Therefore, the proposed identification method is of great significance to the actual navigation control of AUV.

5. Conclusions

Aiming at the identification problem of the AUV system, this paper adopts a neural network hyperparameter optimization method based on Q-Learning. This method has been experimentally verified, and the conclusions can be summarized as follows:

1. The LSTM framework has the characteristics of natural Markovization, which can model time series data with high precision. It is found that the historical data of AUV implies the causal relationship of its dynamic model and has the characteristics of a hidden Markov model. The experimental results also show that the adopted method can predict the AUV model well.

2. Optimally selecting hyperparameters can significantly improve the efficiency of LSTMs in specific tasks. It is concluded that the improved Q-learning method can make the LSTM neural network realize the high-precision identification of the AUV system.

3. The offline training in the system model identification framework can reduce online learning time and ensure the security of the initial online use. The online learning model can also ensure its validity.

4. The proposed method has high model identification accuracy and has certain application prospects. We can apply this method to fault diagnosis of AUV, design of the model-based controller, and other aspects.

However, the hyperparameter optimization method only considers recognition accuracy. Our method can potentially be improved in the convergence speed. Moreover, the performance of the LSTM and the improved Q method used in this paper still has certain limitations. We will improve the identification method and optimization method later.

Author Contributions: Conceptualization, D.W.; methodology, D.W.; software, D.W.; validation, D.W.; formal analysis, B.H.; investigation, D.W.; resources, Y.S.; data curation, Y.S.; writing—original draft preparation, D.W.; writing—review and editing, D.W.; visualization, D.W.; supervision, J.W.; project administration, B.H.; funding acquisition, P.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China (Project No.2016YFC0301400), the Fundamental Research Funds for the Central Universities (Project No.201961005), the National Natural Science Foundation of China (Project No.51379198), the National Natural Science Foundation of China (under grant No.51809246), and the National Natural Science Foundation of Shandong Province (under grant No.ZR2018QF003).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fang, Y.; Huang, Z.; Pu, J.; Zhang, J. AUV position tracking and trajectory control based on fast-deployed deep reinforcement learning method. *Ocean Eng.* **2022**, *245*, 110452. [\[CrossRef\]](#)
2. Praczyk, T. Using Neuro—Evolutionary Techniques to Tune Odometric Navigational System of Small Biomimetic Autonomous Underwater Vehicle—Preliminary Report. *J. Intell. Robot. Syst.* **2020**, *100*, 363–376. [\[CrossRef\]](#)
3. Yuan, C.; Licht, S.; He, H. Formation Learning Control of Multiple Autonomous Underwater Vehicles With Heterogeneous Nonlinear Uncertain Dynamics. *IEEE Trans. Cybern.* **2018**, *48*, 2920–2934. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Qiao, L.; Zhang, W. Adaptive Second-Order Fast Nonsingular Terminal Sliding Mode Tracking Control for Fully Actuated Autonomous Underwater Vehicles. *IEEE J. Ocean. Eng.* **2019**, *44*, 363–385. [\[CrossRef\]](#)
5. Min, F.; Pan, G.; Xu, X. Modeling of Autonomous Underwater Vehicles with Multi-Propellers Based on Maximum Likelihood Method. *J. Mar. Sci. Eng.* **2020**, *8*, 407. [\[CrossRef\]](#)
6. Deng, F.; Levi, C.; Yin, H.; Duan, M. Identification of an Autonomous Underwater Vehicle hydrodynamic model using three Kalman filters. *Ocean Eng.* **2021**, *229*, 108962. [\[CrossRef\]](#)
7. Wu, B.; Han, X.; Hui, N. System Identification and Controller Design of a Novel Autonomous Underwater Vehicle. *Machines* **2021**, *9*, 109. [\[CrossRef\]](#)
8. Wang, D.; He, B.; Shen, Y.; Li, G.; Chen, G. A Modified ALOS Method of Path Tracking for AUVs with Reinforcement Learning Accelerated by Dynamic Data-Driven AUV Model. *J. Intell. Robot. Syst.* **2022**, *104*, 1–23. [\[CrossRef\]](#)
9. Bresciani, M.; Costanzi, R.; Manzari, V.; Peralta, G.; Terracciano, D.S.; Caiti, A. Dynamic parameters identification for a longitudinal model of an AUV exploiting experimental data. In Proceedings of the Global Oceans 2020: Singapore—U.S. Gulf Coast, Biloxi, MS, USA, 5–30 October 2020; pp. 1–6.

10. Jiang, C.m.; Wan, L.; Sun, Y.S. Design of motion control system of pipeline detection AUV. *J. Cent. South Univ.* **2017**, *24*, 637–646. [\[CrossRef\]](#)
11. Wang, J.G.; Jiang, C.M.; Sun, Y.S.; He, B.; Li, J.Q. Neural network identification of underwater vehicle by hybrid learning algorithm. *Zhongnan Daxue Xuebao (Ziran Kexue Ban)/J. Cent. South Univ. (Sci. Technol.)* **2011**, *42*, 427–431.
12. Muñoz Palacios, F.; Cervantes Rojas, J.S.; Valdovinos, J.; Sandre Hernandez, O.; Salazar, S.; Romero, H. Dynamic Neural Network-Based Adaptive Tracking Control for an Autonomous Underwater Vehicle Subject to Modeling and Parametric Uncertainties. *Appl. Sci.* **2021**, *11*, 2797. [\[CrossRef\]](#)
13. Kim, D.; Park, M.; Park, Y.L. Probabilistic Modeling and Bayesian Filtering for Improved State Estimation for Soft Robots. *IEEE Trans. Robot.* **2021**, *37*, 1728–1741. [\[CrossRef\]](#)
14. Zhang, T.; Zheng, X.Q.; Liu, M.X. Multiscale attention-based LSTM for ship motion prediction. *Ocean Eng.* **2021**, *230*, 109066. [\[CrossRef\]](#)
15. Shahi, T.B.; Shrestha, A.; Neupane, A.; Guo, W. Stock Price Forecasting with Deep Learning: A Comparative Study. *Mathematics* **2020**, *8*, 1441. [\[CrossRef\]](#)
16. Yam, L.; Yan, Y.; Jiang, J. Vibration-based damage detection for composite structures using wavelet transform and neural network identification. *Compos. Struct.* **2003**, *60*, 403–412. [\[CrossRef\]](#)
17. Dahunsi, O.; Pedro, J. Neural Network-Based Identification and Approximate Predictive Control of a Servo-Hydraulic Vehicle Suspension System. *Eng. Lett.* **2010**, *18*, 357.
18. Liu, J.; Du, J. Composite learning tracking control for underactuated autonomous underwater vehicle with unknown dynamics and disturbances in three-dimension space. *Appl. Ocean Res.* **2021**, *112*, 102686. [\[CrossRef\]](#)
19. Dong, X.; Shen, J.; Wang, W.; Shao, L.; Ling, H.; Porikli, F. Dynamical Hyperparameter Optimization via Deep Reinforcement Learning in Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 1515–1529. [\[CrossRef\]](#)
20. Mercangöz, M.; Cortinovis, A.; Schönborn, S. Autonomous Process Model Identification using Recurrent Neural Networks and Hyperparameter Optimization. *IFAC-PapersOnLine* **2020**, *53*, 11614–11619. [\[CrossRef\]](#)
21. Sena, M.; Erkilinc, M.S.; Dippon, T.; Shariati, B.; Emmerich, R.; Fischer, J.K.; Freund, R. Bayesian Optimization for Nonlinear System Identification and Pre-Distortion in Cognitive Transmitters. *J. Light. Technol.* **2021**, *39*, 5008–5020. [\[CrossRef\]](#)
22. Baker, B.; Gupta, O.; Naik, N.; Raskar, R. Designing Neural Network Architectures using Reinforcement Learning. *arXiv* **2016**, arXiv:1611.02167.
23. Chen, S.; Wu, J.; Liu, X. EMORL: Effective multi-objective reinforcement learning method for hyperparameter optimization. *Eng. Appl. Artif. Intell.* **2021**, *104*, 104315. [\[CrossRef\]](#)
24. Liu, X.; Wu, J.; Chen, S. A context-based meta-reinforcement learning approach to efficient hyperparameter optimization. *Neurocomputing* **2022**, *478*, 89–103. [\[CrossRef\]](#)
25. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M.A. Playing Atari with Deep Reinforcement Learning. *arXiv* **2013**, arXiv:1312.5602.
26. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347.
27. Watkins, C.; Dayan, P. Technical Note: Q-Learning. *Mach. Learn.* **1992**, *8*, 279–292. [\[CrossRef\]](#)
28. Nouri, N.M.; Valadi, M.; Asgharian, J. Optimal input design for hydrodynamic derivatives estimation of nonlinear dynamic model of AUV. *Nonlinear Dyn.* **2018**, *92*, 139–151. [\[CrossRef\]](#)
29. Prestero, T. Verification of a Six-Degree of Freedom Simulation Model for the REMUS Autonomous Underwater Vehicle. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2011.
30. Rashid, T.; Hassan, M.; Mohammadi, M.; Fraser, K. Improvement of Variant Adaptable LSTM Trained With Metaheuristic Algorithms for Healthcare Analysis. In *Research Anthology on Artificial Intelligence Applications in Security*; Information Resources Management Association: Hershey, PA, USA, 2021; pp. 1031–1051.
31. Rashid, T.A.; Fattah, P.; Awla, D.K. Using Accuracy Measure for Improving the Training of LSTM with Metaheuristic Algorithms. *Procedia Comput. Sci.* **2018**, *140*, 324–333. [\[CrossRef\]](#)
32. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [\[CrossRef\]](#)
33. Sutton, R.; Barto, A. Reinforcement Learning: An Introduction. *IEEE Trans. Neural Netw.* **1998**, *9*, 1054. [\[CrossRef\]](#)