ICIC Express Letters
Volume **13**, Number **11**, November **2019**

ICIC International ©2019 ISSN 1881-803X
pp. 1079–1086

A CONVOLUTION NEURAL NETWORK BASED OBSTACLE AVOIDING METHOD FOR UNMANNED UNDERWATER VEHICLE

Changjian Lin, Hongjian Wang*, Mingyu Fu, Jianya Yuan and Dan Yu

College of Automation
Harbin Engineering University
No. 145, Nantong Street, Nangang District, Harbin 150001, P. R. China
18243088393@163.com; { fumingyu; cocomomo }@hrbeu.edu.cn; 2414450031@qq.com
*Corresponding author: cctime99@163.com

Received May 2019; accepted August 2019

Abstract. *The development of the deep neural network has generated new possibilities for the problem of Unmanned Underwater Vehicle (UUV) autonomous obstacle avoidance. This paper proposes a Convolution Neural Network (CNN) based obstacle avoiding method for UUV using multi-beam Forward Looking Sonar (FLS) in unknown environments. The proposed method is used to learn to recognize and to avoid obstacles end-to-end. In order to avoid the frequent fluctuation of the output instruction, the outputs of algorithm are fed back to the input again in chronological order. Extensive experiments are designed to test the conciseness, generation and effectiveness of the proposed method. And the simulation results show that in addition to conciseness and effectiveness, the proposed CNN based autonomous obstacle avoidance method's enormous generation and adaptation make it possible for UUV to avoid obstacles in complex and dynamic environments, even if the training set is generated in simple and static environments.*
Keywords: Autonomous obstacle avoidance, Convolution neural network, Unmanned underwater vehicle, Real time

1. **Introduction.** The progress of human research, development and utilization of the oceans has never ceased. The emergence of UUV not only brings benefits to human resources, but also provides new possibilities for ocean exploration. UUV usually works in dangerous or distant environments with the features of personnel inaccessibility, that people cannot or do not want to go to. Therefore, improving the autonomous ability of UUV including autonomous perception, autonomous planning and capacity for independent behavior is the important trend of its development. Unmanned systems were not able to learn new behavior or acquire new knowledge simply by repeating their actions according to pre-programmed design [1]. However, with the development of artificial intelligence, the ability to learn autonomously of unmanned systems is no longer a fantasy.

Some researches on autonomous learning of unmanned systems are actively being carried out [2]. At present, autonomous learning methods mainly include Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL) methods. Reinforcement learning is an algorithm designed to simulate the process of human beings learning from the environment. By introducing the traditional RL method into the UUV real-time collision avoidance system, the UUV can interact with the seabed environment continuously, so as to gain experience and improve ability of collision avoidance. However, slow learning speed and the curse of dimensionality hinder the application of RL in UUV obstacle avoidance planning. In order to accelerate the learning speed, Kawano and Ura [8] proposed a two level RL algorithm which refers position and velocity of UUV respectively. The fast RL algorithm has the ability to solve the curse of dimensionality caused by the complex dynamics of UUV, and complete a learning process within an acceptable time

DOI: 10.24507/icicel.13.11.1079

range. In order to achieve a high learning speed, Sun [9] introduced prior knowledge to Q-learning, which urges UUV to perform desired actions instead of randomly selected actions. To cope with the curse of dimensionality, the neural network is applied to the RL algorithm, and the strong nonlinear processing ability of the neural network method is used to improve the generalization ability of the algorithm. Ran [10] introduced the abstract thought to solve the curse of dimensionality and built a MAXQ learning algorithm based on a three level structure for UUV path planning. The learning task of each level is assigned according to the detected information of environment, the current state of UUV and the information transmitted by the adjacent levels. Among them, the root task is the path planning for UUV; the next layer contains two sub-tasks: obstacle avoidance and trend to the target; and the third level is the basic action execution level.

Traditional RL is limited to the small and discrete action space and sample space. However, UUV real-time obstacle avoidance often has a large state space and continuous action space. The problem of underwater obstacle avoidance requires the input of high-dimensional environmental information which is difficult to deal with by traditional RL method. DRL combines the high dimensional input of deep learning with RL. Dooraki and Lee [11] presented a memory-based DRL algorithm for robots autonomous exploration in unknown environments. The algorithm has long-term and short-term memory, so that robot can distinguish between similar states and learn from its own experiences. Cheng and Zhang [12] proposed a concise DRL obstacle avoidance algorithm for the underactuated unmanned marine vessel. The proposed concise DRL method solves the problem of usability caused by the complicated control law in the traditional analytic approach. [13] presented deep Q network autonomous navigation and obstacle avoidance of self-driving cars. [14] is concerned with an algorithm that utilizes DRL to learn exploration knowledge over office blueprints. Wang et al. [15] achieved mobile robots obstacle avoidance by a behavior learning algorithm based on deep belief networks.

However, there are still the following challenges in obstacle avoidance planning based on DRL: low sample utilization, reward functions that are difficult to design, and poor adaptive ability. In this paper, we develop an end-to-end obstacle avoidance algorithm based on CNN for UUV. This means that there is no extra processing to extract environment feature. In order to avoid repeated and redundant actions, short-term memory of historical output is introduced to the algorithm. Extensive experiments are used to evaluate the performance of the proposed algorithm. Due to the strong ability of learning and feature extraction, the proposed method can quickly extract and learn the knowledge in the samples, so as to realize obstacle avoidance. And the proposed method is capable for avoiding obstacles in more complex environments than the sample set, so that its strong adaptive ability and generalization is proved.

Besides the last part of conclusion and development of research, this paper can be divided into several parts as follows. In Section 2, the simulation models of UUV and FLS are introduced briefly. The structure of CNN for obstacle avoidance is described in Section 3. Section 4 expounds the framework of CNN based obstacle avoidance method in detail. And the experiments and analysis are given in Section 5.

2. **Simulation Model.**

2.1. **Simulation model of UUV.**

In attitude control of UUV, the pitch adjustment is more difficult than yaw adjustment, so the obstacle avoidance planning is usually achieved by yaw adjustment. In this paper, UUV horizontal obstacles avoidance is considered and a 2-dimensional reference frame is established as Figure 1. Establish North East-fixed reference frame NOE as global reference frame and body-fixed reference frame $x_b o_b y_b$ as the local reference frame. And the origin of body reference frame o_b coincides with the center of gravity of UUV. x, y, ψ are positions and heading of UUV in global reference

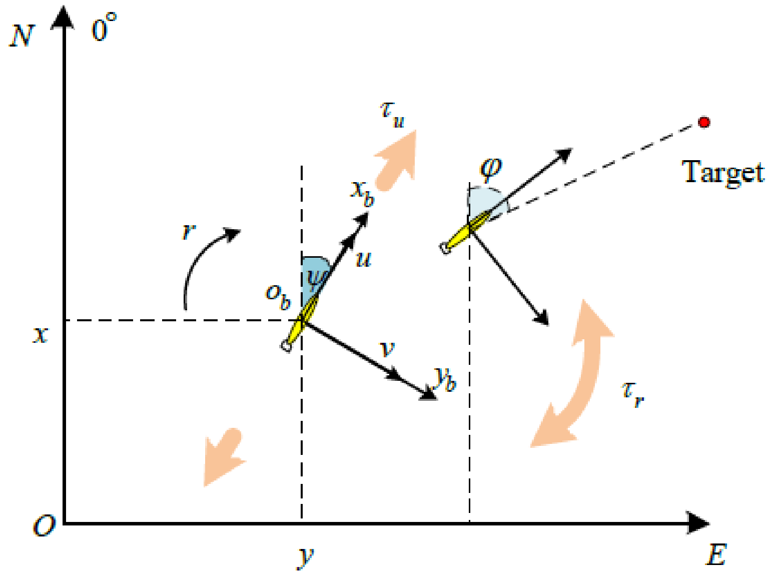


Figure 1. Global and local reference frames

frame, u, v, r denote the velocity about surge, sway and yaw of UUV in local reference frame, φ is the relative angle of UUV with the target, τ denotes moment of force.

The 3 degrees-of freedom control model is built for UUV. The kinematic and dynamic of UUV can be described as:

$$\begin{cases} \dot{x} = u\cos(\psi) - v\sin(\psi) \\ \dot{y} = u\sin(\psi) + v\cos(\psi) \\ \dot{\psi} = r \end{cases} \quad \begin{cases} \dot{u}_r = \left(-d_{11}u_r + \tau_u\right)/m_{11} \\ \dot{v}_r = \left(Am_{33} - Bm_{23}\right)/\left(m_{22}m_{33} - m_{23}^2\right) \\ \dot{r} = \left(Bm_{22} - Am_{23}\right)/\left(m_{22}m_{33} - m_{23}^2\right) \end{cases} \tag{1}$$

where

$$A = -d_{22}v_r + (d_{23} - u_r c_{23})r \quad B = (d_{32} - u_r c_{32})v_r - d_{33}r + \tau_r \tag{2}$$

$$M = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & -Y_{\dot{r}} \\ 0 & -Y & I_z - N_{\dot{r}} \end{bmatrix} \quad D = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix} \tag{3}$$

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

where X, Y are forces in x, y direction respectively, M is inertia matrix of UUV, N is moment of force in ψ direction, C denotes Coriolis-centripetal matrix, D signifies hydrodynamic damping matrix, and m is the mass of UUV.

2.2. **Simulation model of FLS.** A 2-dimensional simulation model of FLS is built to detect environmental information around UUV. The maximum range of detection is 120 m from the sonar head. Utilizing 80 dynamically focused receive beams spaced at 1.5°, the FLS model measures a 120° swath coverage area. The detection data of every beam at time t is stored in a vector $\mathbf{S}_t = \left[s_t^0, s_t^1, \ldots, s_t^{79}\right]$, s_t^i donates the shortest distance between FLS and the obstacle within the detection range measured by beam i at time t.

3. **Convolution Neural Network.** CNN is a kind of special neural network for processing data that has a known grid-like topology [16], and is famous for its powerful ability of feature extraction. The input of our algorithm can be viewed as a 2-D grid sampling at a fixed time interval. The algorithm structure is shown in Figure 2. A CNN with $80 \times T$ inputs is used for extracting feature where T is the time steps for historical observations.

The filters size of Conv1-4 is set as 3×3. Then two fully connected layers are used to process the outputs of convolutional layers for generating yaw and velocity. $\Delta\psi$, v are the feedback of yaw and velocity respectively. The forward propagation of this network is as follows.

Step 1: Fill in the edges of the original input to get the input tensor a^1.

Step 2: Initialize the weights W and biases b of all filters in hidden layers.

Step 3: For $l = 2$ to 5, $a^l = pool\left(ReLU\left(z^l\right)\right) = ReLU\left(a^{l-1} * W^l + b^l\right)$. Here a^l is the output tensor of lth layer, $pool(\cdot)$ denotes the process of shrinking tensors according to pooling operations, $ReLU(\cdot)$ denotes the rectified linear unit, and $*$ denotes the convolution operation.

Step 4: For full connected layer and output layer, $a^l = \sigma\left(z^l\right) = \sigma\left(W^l a^{l-1} + b^l\right)$.

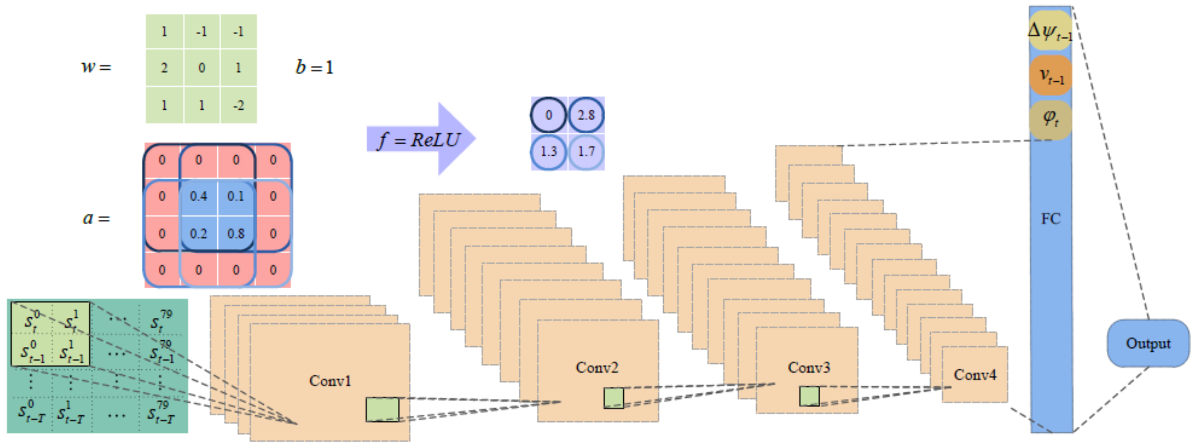


FIGURE 2. Network structure of the CNN for obstacle avoiding

The network constantly adjusts the weights according to the errors to reach convergence. The back propagation of this network is as follows.

Step 1: Calculate the gradient error of the output layer δ^L according to the loss function.

Step 2: Calculate the gradient error of hidden layers. For $l = L-1$ to 2, determine the type of lth hidden layer. If full-connected layer, then $\delta^l = \left(W^{l+1}\right)^T \delta^{l+1} \odot \sigma'\left(z^l\right)$.

If convolutional layer, then $\delta^l = upsample\left(\delta^{l+1}\right) \odot \sigma\left(z^l\right) * rot180\left(W^{l+1}\right) \odot \sigma\left(z^l\right)$. $upsample(\cdot)$ denotes the operation of magnifying the matrix and redistributing the elements, $rot180(\cdot)$ means rotating 180°.

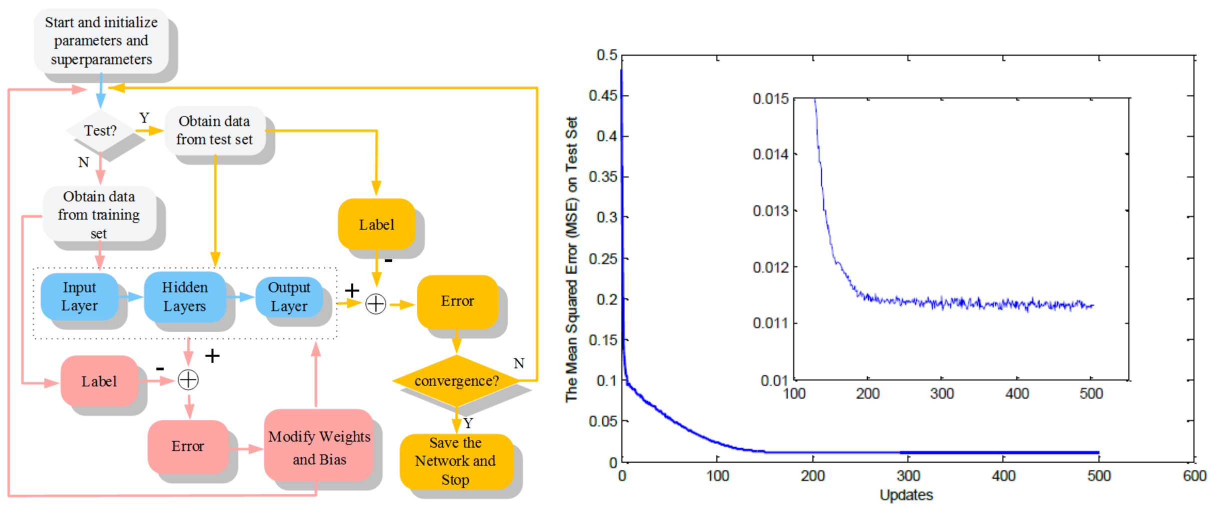
Step 3: Update all weights and bias.

4. Obstacle Avoidance Planning.

4.1. Training and learning. In this paper, a dataset including 1,000,000 training samples and 1,000 test samples is used for network training. Every learning environment contains several obstacles of random size and position. And Min-Max normalization is used to simplify input data.

To overcome the problem of overfitting, dropout with 0.6 keep probability is used in training. Mean Squared Error (MSE) is used as the loss function. The network minimizes the loss function by the Adam optimizer. The weights are updated using the mini-batch gradient descent, batch size is 5,000, and the maximum number of iterations is 10,000. Test the learning effect of CNN based obstacle avoidance model on the test set per 20 iterations. The training process is shown in Figure 3(a).

The convergence process of MSE on test set is shown in Figure 3(b). As shown in the figure, MSE converges rapidly in the early stage of training. And with the progress of training, the convergence speed of MSE gradually slows down, and finally converges to a close vicinity to zero. The error converging to zero means that the algorithm is fully fitted



(a) (b)

Figure 3. Training process (a) and the MSE (b) of CNN based obstacle avoidance network

to the training set data, which will lead to over-fitting, so that the algorithm can only deal with the same or similar situations as the training set. In order to give consideration to learning ability, exploration ability and generalization ability of the obstacle avoidance planning algorithm, the error is expected to converge to a close vicinity to zero, not zero so that the algorithm can avoid obstacles in complex and dynamic environments after training in sample and static environments.

4.2. **The structure of obstacle avoidance method.** The structure of CNN based obstacle avoidance method is shown in Figure 4. The fully trained network outputs commands according to the detected information of FLS, the relative position of UUV with target obtained by motion and attitude sensors and output feedback of network. Then the motion controller navigates UUV to avoid obstacles according to the commands output by obstacle avoidance method.

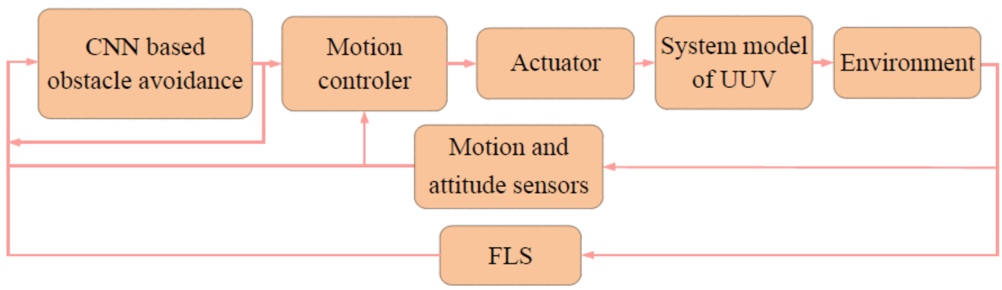


Figure 4. Framework of CNN based obstacle avoidance method

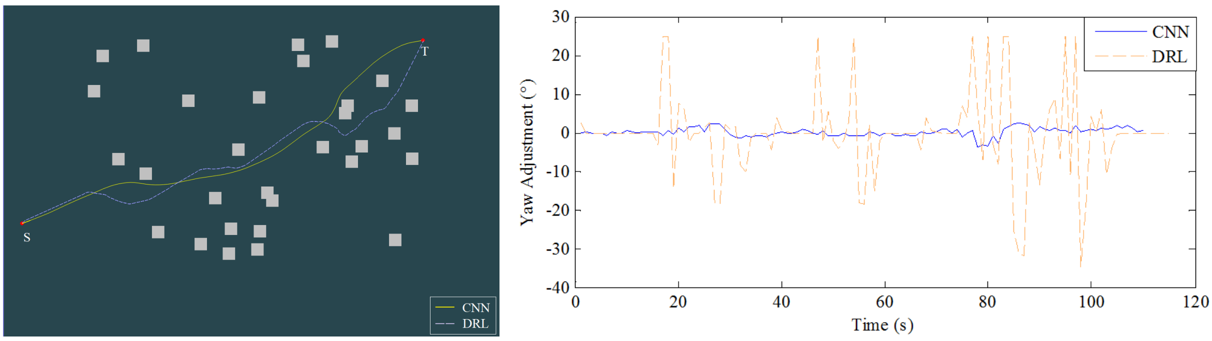
5. **Experiments and Results.** In order to evaluate the CNN based obstacle avoidance method comprehensively, statistical experiment, designed experiment in static environment and dynamic environment are conducted. In this section, UUV navigates in maps of size 800 m $\times$ 1200 m at a uniform velocity 8 kn, the frequency of FLS is set as 1 Hz. There are 4 convolution layers and 2 full connected layers in the obstacle avoidance network, and the time step T is set to 10.

5.1. **Statistical experiment.** To quantitatively analyze the performance of the proposed method in obstacle avoidance, 300 maps with different number of obstacles are generated randomly in this section. And the performance of the CNN based obstacle avoidance method in obstacle avoidance success, time-consumption and path cost is counted in Table 1. Note that the number of obstacles in learning environments is set as 30, and the obstacles are set as rectangles with side lengths greater than 20 and less than 30. As the statistical results show that the CNN based obstacle avoidance method had a good performance even in environments more complex than learning environments.

TABLE 1. The performance of the proposed method in statistical experiment

Number of obstacles	**Obstacle avoidance success**	**Average time-consumption**	**Average path cost**
30	100%	142.66 ms	1006.5 m
50	99%	143.52 ms	1029.33 m
80	95%	145.36 ms	1051.17 m

5.2. **Designed experiment in static environment.** Obstacles are set as rectangles of discrete distribution in learning environments. In order to test the learning outcome, static environment 1 is designed, and the simulation results of CNN and DRL based obstacle avoidance methods are shown in Figure 5. In static environment 1, compared with DRL the track planned by CNN is smoother and no frequent fluctuation. It can be seen from the yaw adjustment curves of UUV shown in Figure 5(b) that the output instruction of obstacle avoidance based on CNN is more consistent with the motion characteristics of UUV than that of DRL.

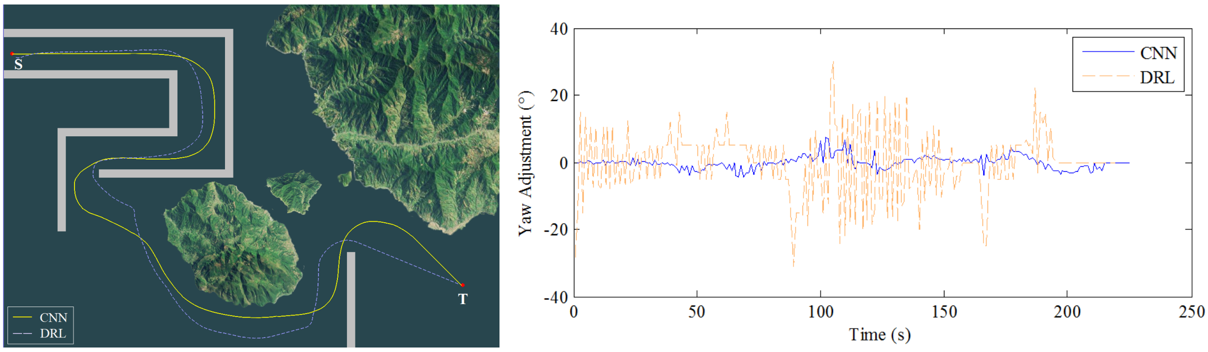


(a) (b)

FIGURE 5. Track (a) and yaw adjustment (b) of UUV in static environment 1 (The simulation results of CNN are represented by solid lines and the dotted lines represent that of DRL.)

A complex static environment 2 is used to test the obstacles avoidance planning performance of the proposed method. There are large and irregular obstacles distributed in the simulation environment 2. The simulation results are shown in Figure 6. For the track planned by CNN, UUV adjusted the heading slightly and successfully avoided obstacles and reached the target point. As can be seen from Figure 6(b), UUV is guided by DRL yaws sharply and swings frequently, which not only consumes much energy but also reduces the service life of the propulsion system.

5.3. **Designed experiment in dynamic environment.** This section designs a dynamic environment to verify the adaptability of the proposed method. The simulation result is shown in Figure 7. There are 30 dynamic obstacles (blocks with arrows) and 30 static



(a) (b)

Figure 6. Track (a) and yaw adjustment (b) of UUV in static environment 2 (The simulation results of CNN are represented by solid lines and the dotted lines represent that of DRL.)

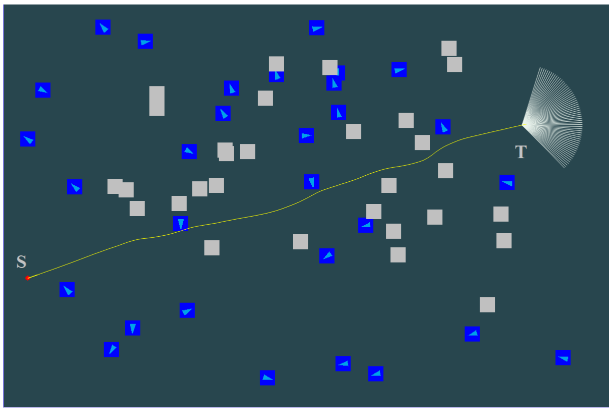


Figure 7. Simulation result in dynamic environment

obstacles in the environment. The direction of moving obstacles is indicated by arrows, and the velocity of obstacles is set as 8 kn. The simulation result shows that the proposed CNN based obstacle avoidance method is able to guide UUV to avoid obstacles in complex and dynamic environment even if the network was trained in simple and static environments.

6. **Conclusions.** This paper proposed a CNN based online obstacle avoidance method for UUV. The method recognizes obstacles and guides UUV to avoid obstacles according to real-time sensors information. This means that there is no extra processing to extract environment feature. In order to avoid repeated and redundant actions of UUV during navigation, short-term memory of historical output of the algorithm is added in this paper. The stability and instantaneity of the method were proved by statistical experiment. The obstacle avoidance experiments in different scenarios proved the learning ability and adaptability of the proposed algorithm that UUV is able to avoid obstacles in complex and dynamic environments, even if the network was trained in simple and static environments. In future works, obstacle avoidance strategy and obstacle state estimation should be introduced into our algorithm to improve the ability of obstacle avoidance and autonomous motion planning of UUV.

Acknowledgment. This research work is supported by Natural Science Foundation of China (No. 61633008, No. 51609046) and Natural Science Foundation of Heilongjiang Province under Grant F2015035.

REFERENCES

[1] K. Kim, D. Kim and J. Lee, Deep learning based on smooth driving for autonomous navigation, *Proc. of 2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, Auckland, New Zealand, pp.616-621, 2018.

[2] H. Kawano, Method for applying reinforcement learning to motion planning and control of under-actuated underwater vehicle in unknown non-uniform seaflow, *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.996-1002, 2005.

[3] C. Zhou, F. Li, W. Cao, C. Wang and Y. Wu, Design and implementation of a novel obstacle avoidance scheme based on combination of CNN-based deep learning method and liDAR-based image processing approach, *Journal of Intelligent & Fuzzy Systems*, vol.35, no.2, pp.1695-1705, 2018.

[4] Z. Wang and L. Hu, Industrial manipulator path planning based on deep Q-learning, *Control and Instruments in Chemical Industry*, vol.45, no.2, pp.141-145, 2018.

[5] L. Tai, S. Li and M. Liu, A deep-network solution towards model-less obstacle avoidance, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016.

[6] N. T. Luy, Reinforecement learning-based optimal tracking control for wheeled mobile robot, *Proc. of 2012 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER 2012)*, pp.371-376, 2012.

[7] H. M. La, R. S. Lim, W. Sheng and J. Chen, Cooperative flocking and learning in multi-robot systems for predator avoidance, *Proc. of 2013 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER 2013)*, pp.337-342, 2013.

[8] H. Kawano and T. Ura, Fast reinforcement learning algorithm for motion planning of non-holonomic autonomous underwater vehicle in disturbance, *Proc. of 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, pp.903-908, 2002.

[9] K. Sun, *Research on Obstacle Avoidance for AUV Based on Reinforcement Learning*, Master Thesis, Shenyang Jianzhu University, 2016.

[10] X. Ran, *Research on AUV Path Planning Method Based on Hierarchical Reinforcement Learning*, Master Thesis, Harbin Engineering University, 2017.

[11] A. R. Dooraki and D. Lee, An end-to-end deep reinforcement learning-based intelligent agent capable of autonomous exploration in unknown environments, *Sensors*, vol.18, no.10, pp.1-17, 2018.

[12] Y. Cheng and W. Zhang, Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels, *Neurocomputing*, vol.272, pp.63-73, 2018.

[13] A. R. Fayjie, S. Hossain, D. Oualid and D. Lee, Driverless car: Autonomous driving using deep reinforcement learning in urban environment, *Proc. of the 15th International Conference on Ubiquitous Robots (UR)*, Hawaii, USA, pp.896-901, 2018.

[14] D. Zhu, T. Li, D. Ho, C. Wang and M. Q. Meng, Deep reinforcement learning supervised autonomous exploration in office environments, *Proc. of 2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, pp.7548-7555, 2018.

[15] J. Wang, R. Yan, H. Tang and F. Sun, Automatic object searching and behavior learning for mobile robots in unstructured environment by deep belief networks, *IEEE Trans. Cognitive and Developmental Systems (Early Access)*, 2018.

[16] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, MIT Press, 2016.