Petri Based Recurrent Fuzzy Neural Control for SY-II Remote Operated Vehicle

Huang Hai, Wan Lei, Zhang Guocheng, Pang Yongjie
Key Laboratory of Science and Technology for National Defense of Autonomous Underwater Vehicle
Harbin Engineering University
Harbin, China

***Abstract*-Recurrent fuzzy neural network is widely applied in many areas because it combines the advantages of low level learning and high level reasoning. In considering the complicated factors and requirements for the Remote Operated Vehicle(ROV) control, petri network has been introduced to design a dynamic controller for underwater robot. It intends to reduce the computation burdens during network parameters learning. The gradient descent method has been used for online training. In order to guarantee its convergence, we have used the discrete Lyapunov function to determine its learning rate. The tank experiments have proved that the controller can adjust control quantity to reduce caculation and present strong advantages in the ROV robustness control.**

Keywords-Remote Operated Vehicle; Petri Network(PN); Recurrent Fuzzy Neural Network(RFNN)

I. INTRODUCTION

With aboundunt power supply, ROV can exchange a great deal of information with its surface work station. It can be utilized broadly in marine exploration and resource exploitation[1,2]. To realize object detection and oceania task, accurate motion control must be fundamental. However, it is very difficult to construct a precise dynamic model for 6-DOF ROV underwater motion, not only because the sea environment is complicated and unknown, but the ROV itself is the strongly nonlinear and coupling[3,4].

In the combination with S surface control and fuzzy concept, reference[5] not only improved the control effectiveness but simplified the controller design. But this method is based on proportion and integration priciple, it does not possess the capacity of self-learning. The adjustment of its controller parameters depends completely on researcher's experience. In consideration with strong nonlinearity and coupling characters of underwater robots, Neural Network (NN) is good at self-learning to adapt to environment change[6,7]. However vibration may prone to happen, because the NN learning is usually lagging out. In compare with Back Propagation (BP) Neural Network, Recurrent Neural Network (RNN) can deal with dynamic change much better[8,9] . On the other hand, Fuzzy Neural Network (FNN) is widely applied in the control of Underwater robots aboard, because it combines low level learning with high level reasoning[10,11]. Apparently, taking dynamic feedback as memory element, RFNN possesses advantages of the two, but with complicated calculation[12].

Recently, Petri-Network(PN) has become an effective tool to model, control, analyse and optimize. In order to reduce calculation and improve the effectiveness, we introduce PN in the designation of RFNN to solve the ROV control problems such as strong nonlinearity and unknow oceania environment .

II. SY-II ROV AND ITS CONTROL STRUCTURE

SY-II ROV (Figure 1) is an open frame underwater robot. It is equipped with a depth guage, magnetic compass, two main thrusters, two side thrusters and two vertical thrusters. If we establish absolute reference frame $E-\xi\eta\zeta$ and relative reference frame G-xyz and neglect roll, 5-DOF motion model can be obtained as:

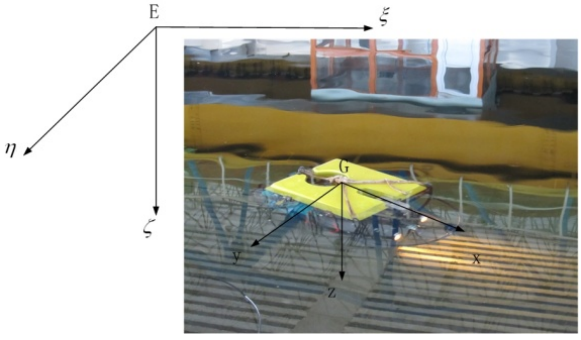

Figure 1. SY-II ROV

$$\begin{cases} m\left[\left(\dot{u}-vr+wq\right)-x_G(q^2+r^2)+y_G\dot{r}+z_G\dot{q}\right]=X \\ m\left[\left(\dot{v}+ur\right)-y_Gr^2+z_Gqr+x_G\dot{r}\right]=Y \\ m\left[\left(\dot{w}-uq\right)-z_Gq^2-x_G\dot{q}+y_Grq\right]=Z \\ I_y\dot{q}+m\cdot\left[z_G\cdot\left(\dot{u}+wq-vr\right)-x_G\cdot\left(\dot{w}-uq\right)\right]=M \\ I_z\dot{r}+m\cdot\left[x_G\cdot\left(\dot{v}+ur\right)-y_G\cdot\left(\dot{u}-vr\right)\right]=N \end{cases}$$

where m is the mass of robot; x_G , y_G , z_G are the gravity centers of the robot ; I_x , I_y , I_z are the moments of inertia in associated with axes Gx , Gy , Gz respectively; u , v , w , q , r are velocities and angular velocities of the 5 DOF respectively; X , Y , Z , M , N are torques of 5-DOF respectively. Thus the hydrodynamic model is obtained:

$$M(\vec{q})\ddot{\vec{q}}+C(\dot{\vec{q}})\dot{\vec{q}}+D(\dot{\vec{q}})\dot{\vec{q}}+g(\vec{x})=\vec{\tau}$$
$$\dot{\vec{x}}=Q\dot{\vec{q}} \quad (1)$$

where $\dot{\vec{q}}=[u \quad v \quad w \quad q \quad r]^T$, M is the 5×5 inertia matrix, $\vec{x}=[x \quad y \quad z \quad \theta \quad \psi]^T$ is the 5×1 position vector according to absolute reference frame. $C(\dot{\vec{q}})$ is the 5×1 vector of centrifugal and Coriolis terms, $D(\dot{\vec{q}})$ is resistance matrix, $g(\vec{x})$ is gravity, $\vec{\tau}$ is the input force vector of thrusters and torques from the controller.

Figure 2. illustrates SY-II's control structure block diagram. The desired forces/torques can be computed by the controller according to the objective values and the the sensors feedback. The depth is obtained from pressure sensor, and SY-II pitch, roll, and heading are measured by the magnetic sensor. The controller in the diagram is PN based RFNN.

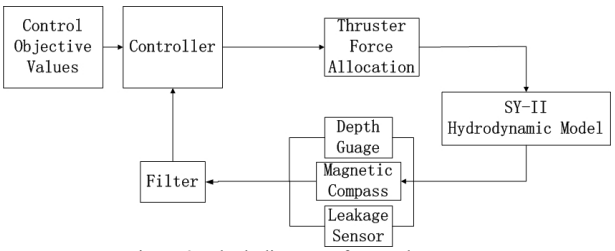

Figure 2. Block diagram of control structure

III. CONTROLLER DESIGN

A.The Controller

Figure.3 is the 5-layer block diagram of PN based RFNN. It includes input layer, membership layer, petri layer, rule layer and output layer. The feedback connection is realized through reccurent feedback at membership layer. The propagation fuction in each layer are issued as follows.

1). The first layer is input layer. Its output node is expressed as: $y_i^{(1)}=x_i^{(1)}$, $i=1,2,3...n_i$ (2)

where $x_i^{(1)}$ and $y_i^{(1)}$ are the input and output of the ith node in this layer, n_i is the number of input variables.

2). The second layer is membership layer. Each node in this layer acts as a membership function. The input of this layer is expressed as:

$$x_i^{(2)}(n)=y_i^{(1)}(n)+\eta_{ij}(n-1)\theta_{ij} \quad (3)$$

where θ_{ij} denotes the weight of the self-feedback loop, n is the iteration times, $\eta_{ij}(n-1)$ represents the last time output of 2nd layer. The output of 2nd layer is defined with Gaussian membership functions as:

$$y_{ij}^{(2)}=\exp\left(-\frac{(x_i^{(2)}-m_{ij})^2}{(\sigma_{ij})^2}\right), j=1,2,\ldots,n_j \quad (4)$$

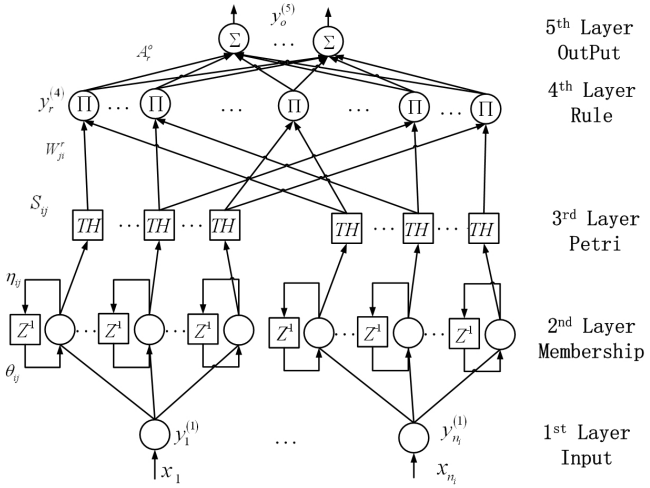

Figure 3. Peri-based RFNN controller

where m_{ij} and σ_{ij} are the mean and standard deviation of the Gaussian function respectively. n_j is the number of the linguistic variables associated with each input.

3). The third layer is Petri layer whose purpose is to produce threshold by using competition laws to determine the training requirements:

$$S_{ij}=\begin{cases}1 & y_{ij}^{(2)}\geq TH\\ 0 & y_{ij}^{(2)}<TH\end{cases} \quad (5)$$

where S_{ij} is the switch and TH is a dynamic threshold value in corresponding with errors to be determined in III B.

4). The 4th layer is rule layer, which multiplies the input variables to produce output:

$$y_r^{(4)}=\begin{cases}0 & TH=0\\ \prod_{i=1}^{n_i}W_{ji}^r y_{ij}^{(2)} & TH=1\end{cases} \quad (6)$$

where W_{ji}^4 is the weights between the 3rd layer and the 4th layer, $r=1,2,\ldots,n_r$ is the total rule numbers.

5. The 5th layer is output layer:

$$y_o^{(5)}=\sum_{r=1}^{n_r}A_r^o y_r^{(4)} \quad (7)$$

where A_r^o is the output action strength with respect to the rth rule.

B. Strategy for Online Learning

The core of learning strategy is to obtain a gradient vector so that each element can be defined as the derivative of an energy function. SY-II ROV can be controlled with auto-depth and auto-heading because it is equipped with depth guage and magnetic sensor. This paper will deduce the algorithm from depth and heading control, motion control of other DOFs can be obtained similarly.

First of all, we define the energy function E as:

$$E=\frac{1}{2}\left[\left(h_d-h\right)^2+\left(\dot{h}_d-\dot{h}\right)^2+\left(\psi_d-\psi\right)^2+\left(\dot{\psi}_d-\dot{\psi}\right)^2\right]$$

$$=\frac{1}{2}\left(e_{dh}^2+\dot{e}_{dh}^2+e_{d\psi}^2+\dot{e}_{d\psi}^2\right) \qquad 8 \qquad (\ )$$

where h_d is desired diving depth, ψ_d is desired heading, h is the real diving depth, ψ is the real heading, $e_{dh}=h_d-h$ is current depth error, $e_{d\psi}=\psi_d-\psi$ is current heading error, $\dot{h}_d$, $\dot{h}$, $\dot{\psi}$, $\dot{\psi}_d$, $\dot{e}_h$ and $\dot{e}_\psi$ are the derivative of h_d, h, ψ, ψ_d, e_{dh} and $e_{d\psi}$. TH of equation (5) will be defined through the following equation:

$$TH=\frac{\partial\exp\left(-\delta E\right)}{1+\exp\left(-\delta E\right)} \tag{9}$$

where ∂ and δ are positive constants to coordinate TH which will be larger if the error is smaller.

In the output layer the error term to be propagated is:

$$\beta_o=-\left(\frac{\partial E}{\partial e_{dh}}\frac{\partial e_{dh}}{\partial h}\frac{\partial h}{\partial y_o^{(5)}}+\frac{\partial E}{\partial e_{d\psi}}\frac{\partial e_{d\psi}}{\partial \psi}\frac{\partial \psi}{\partial y_o^{(5)}}+\frac{\partial E}{\partial \dot{e}_{dh}}\frac{\partial \dot{e}_{dh}}{\partial \dot{h}}\frac{\partial \dot{h}}{\partial y_o^{(5)}}+\frac{\partial E}{\partial \dot{e}_{d\psi}}\frac{\partial \dot{e}_{d\psi}}{\partial \dot{\psi}}\frac{\partial \dot{\psi}}{\partial y_o^{(5)}}\right)$$

$$=e_{dh}\frac{\partial h}{\partial y_o^{(5)}}+e_{d\psi}\frac{\partial \psi}{\partial y_o^{(5)}}+\dot{e}_{dh}\frac{\partial \dot{h}}{\partial y_o^{(5)}}+\dot{e}_{d\psi}\frac{\partial \dot{\psi}}{\partial y_o^{(5)}}$$

The weight of output layer is updated by the equation (10):

$$\Delta A_k^o=-\mu_\omega\frac{\partial E}{\partial \omega_k^o}=\mu_\omega\left[-\frac{\partial E}{\partial y_o^{(5)}}\right]\left(\frac{\partial y_o^{(5)}}{\partial \omega_k^o}\right)=\mu_\omega\beta_o A_r^o \tag{10}$$

where μ_ω is the learning-rate parameter of the connecting weights. The propagating error term can be calculated as:

$$\beta^{(4)}=-\frac{\partial E}{\partial A_r^o}=\left[-\frac{\partial E}{\partial y_o^{(5)}}\right]\left(\frac{\partial E}{\partial A_r^o}\right)=\begin{cases}\beta_o A_k^o, & A_r^o\neq 0\\ 0, & A_r^o=0\end{cases}$$

In the Petri layer, the error term is:

$$\beta_r^{(3)}=\frac{\partial E}{\partial r_{ij}}$$

$$=\left[\frac{\partial E}{\partial y_o^{(5)}}\frac{\partial y_o^{(5)}}{\partial A_r^o}\right]\left(\frac{\partial A_r^o}{\partial y_{ij}^{(2)}r_{ij}}\frac{\partial y_{ij}^{(2)}r_{ij}}{\partial r_{ij}}\right)$$

$$=\begin{cases}\sum_r \beta_r^{(4)}y_r^{(4)}, & S_{ij}=1\\ 0, & S_{ij}=0\end{cases}$$

where $r_{ij}=-\frac{\left(x_i^{(2)}-m_{ij}\right)^2}{\left(\sigma_{ij}\right)^2}$. Thus he update laws for m_{ij}, σ_{ij} and θ_{ij} are:

$$\Delta m_{ij}=-\mu_m\frac{\partial E}{\partial m_{ij}}$$

$$=\mu_m\left[-\frac{\partial E}{\partial y_o^{(5)}}\frac{\partial y_o^{(5)}}{\partial r_{ij}}\frac{\partial r_{ij}}{\partial m_{ij}}\right]=\mu_m\beta_r^{(3)}\frac{2\left(x_i^{(2)}-m_{ij}\right)^2}{\left(\sigma_{ij}\right)^2}y_{ij}^{(2)}\left(n-1\right)$$

$$\Delta\sigma_{ij}=-\mu_\sigma\frac{\partial E}{\partial \sigma_{ij}}=\mu_\sigma\beta_r^{(3)}\frac{2\left(x_i^{(2)}-m_{ij}\right)^2}{\left(\sigma_{ij}\right)^3} \tag{11}$$

$$\Delta\theta_{ij}=-\mu_\theta\frac{\partial E}{\partial \theta_{ij}}=-\mu_\theta\beta_r^{(3)}\frac{2\left(x_i^{(2)}-m_{ij}\right)}{\left(\sigma_{ij}\right)^2}y_{ij}^{(2)}\left(n-1\right)$$

where μ_m and μ_σ are the learning-rate parameters for the mean and the standard deviation of the Gaussian function respectively, μ_θ is the learning-rate parameter for the self-feedback loop.

$A_k^o\left(n\right)$, $m_{ij}\left(n\right)$, $\sigma_{ij}\left(n\right)$ and $\theta_{ij}\left(n\right)$ are updated as:

$$\left(\bullet\right)\left(n+1\right)=\left(\bullet\right)\left(n\right)+\Delta\left(\bullet\right)\left(n\right) \tag{12}$$

In order to improve learning speed, h, ψ, $\dot{h}$ and $\dot{\psi}$ can be approximated by their sign functions sgn().

$$\frac{\partial\left(*\right)}{\partial y_o^{(5)}}\cong\operatorname{sgn}\left(\frac{\left(*\right)\left(n\right)-\left(*\right)\left(n-1\right)}{y_o^{(5)}\left(n\right)-y_o^{(5)}\left(n-1\right)}\right) \tag{13}$$

where $\left(*\right)$ represents h, ψ, $\dot{h}$ and $\dot{\psi}$.

C. Analysis for Convergence

Values selection for the learning-rate parameters has significant effects on the network performance. In the following, learning-rate parameters are to be analyzed in associated with the convergence of network.

The discrete-type of (8) is:

$$E(n+1)=E(n)+\Delta E(n)$$

$$=E(n)+\sum_{k=1}^{n_r}\left[\frac{\partial E(n)}{\partial A_k^o}\Delta A_k^o\right]+\sum_{i=1}^{n_i}\sum_{j=1}^{n_j}\left[\partial E(n)\left(\frac{\Delta m_{ij}}{\partial m_{ij}}+\frac{\Delta\sigma_{ij}}{\partial\sigma_{ij}}+\frac{\Delta\theta_{ij}}{\partial\theta_{ij}}\right)\right]$$

$$=\frac{1}{4}E(n)-\mu_\omega\sum_{k=1}^{n_r}\left[\frac{\partial E(n)}{\partial y_o^{(5)}}\frac{\partial y_o^{(5)}}{\partial A_k^o}\right]^2$$

$$+\frac{1}{4}E(n)-\mu_m\sum_{i=1}^{n_i}\sum_{j=1}^{n_j}\left[\frac{\partial E(n)}{\partial x_i}\frac{\partial x_i}{\partial y_o^{(5)}}\frac{\partial y_o^{(5)}}{\partial m_{ij}}\right]^2$$

$$+\frac{1}{4}E(n)-\mu_\sigma\sum_{i=1}^{n_i}\sum_{j=1}^{n_j}\left[\frac{\partial E(n)}{\partial x_i}\frac{\partial x_i}{\partial y_o^{(5)}}\frac{\partial y_o^{(5)}}{\partial \sigma_{ij}}\right]^2$$

$$+\frac{1}{4}E(n)-\mu_\alpha\sum_{i=1}^{n_i}\sum_{j=1}^{n_j}\left[\frac{\partial E(n)}{\partial x_i}\frac{\partial x_i}{\partial y_o^{(5)}}\frac{\partial y_o^{(5)}}{\partial \theta_{ij}}\right]^2 \tag{14}$$

Thus μ_ω, μ_m, μ_σ and μ_θ are set as:

$$\mu_\omega=\frac{E(n)}{4\left[\sum_{k=1}^{n_y}\left[\frac{\partial E(n)}{\partial y_o^{(5)}}\frac{\partial y_o^{(5)}}{\partial \omega_k^o}\right]^2+\varepsilon\right]},$$

$$\mu_m=\frac{E(n)}{4\left[\sum_{i=1}^{n_i}\sum_{j=1}^{n_j}\left[\frac{\partial E(n)}{\partial x_i}\frac{\partial x_i}{\partial y_o^{(5)}}\frac{\partial y_o^{(5)}}{\partial m_{ij}}\right]^2+\varepsilon\right]}, \tag{15}$$

$$\mu_\sigma=\frac{E(n)}{4\left[\sum_{i=1}^{n_i}\sum_{j=1}^{n_j}\left[\frac{\partial E(n)}{\partial x_i}\frac{\partial x_i}{\partial y_o^{(5)}}\frac{\partial y_o^{(5)}}{\partial \sigma_{ij}}\right]^2+\varepsilon\right]} \quad \text{and}$$

$$\mu_\theta=\frac{E(n)}{4\left[\sum_{i=1}^{n_i}\sum_{j=1}^{n_j}\left[\frac{\partial E(n)}{\partial x_i}\frac{\partial x_i}{\partial y_o^{(5)}}\frac{\partial y_o^{(5)}}{\partial \theta_{ij}}\right]^2+\varepsilon\right]}$$

where ε is a positive constant .Thus (14) can be written as:

$$E(n+1)\approx\varepsilon(\mu_\omega+\mu_m+\mu_\sigma+\mu_\theta)$$

$$<\frac{E(n)}{4}+\frac{E(n)}{4}+\frac{E(n)}{4}+\frac{E(n)}{4}=E(n) \tag{16}$$

According to (8) and (16), the proposed PN based RFNN controller is convergence.

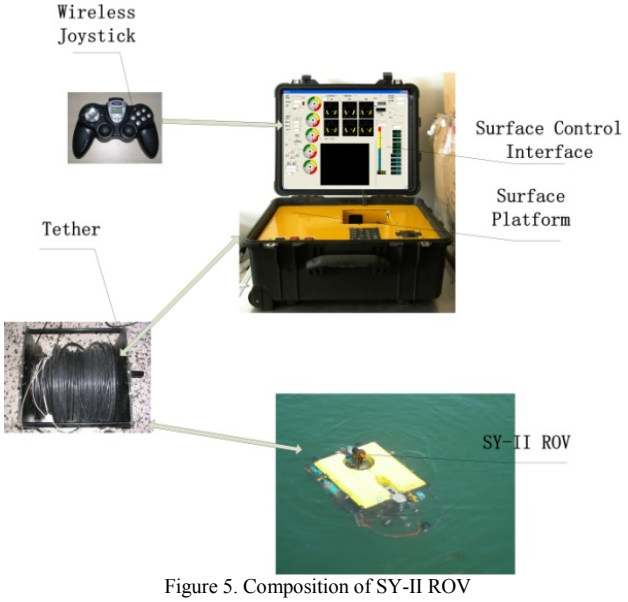

Figure 5. Composition of SY-II ROV

IV. EXPERIMENTS

In order to verify the PN based RFNN controller, experiments has been made in a 50×30×10 meters pond of the Key Laboratory of Science and Technology for National Defense of Autonomous Underwater Vehicle, Harbin Engineering University.

Figure.5 illustrates SY-II ROV composition. It can not only be manipulated through remote control joystick, but be auto-controlled through surface platform. Communication and power are transmitted through tether between surface platform and underwater vehicle.

In these experiments, we have made comparisons between PN based RFNN and S surface control[13,14](which is widely applied in underwater vehicles control). For SY-II, its hard tether provides strong non-linear and random disturbance against auto-control. Particularly being different at different heading, the tether force causes S surface heading auto-control very difficult. However, PN based RFNN controller can response disturbance very quickly, reduce control errors and display strong robustness.

V. SUMMARIES

In order to reduce the computation burdens during recurrent fuzzy neural network parameters learning, petri net has been introduced for the design of dynamic controller of underwater robot. In the online learning strategy, the gradient descent method has been used for online training.

Published by Atlantis Press, Paris, France.

In order to guarantee its convergence, discrete Lyapunov function has been used to determine its learning rate. The tank experiments have proved that the controller can improve the computation efficiency, reduce control errors, vibration and overshoot, and display strong robustness in the underwater robotic control.

ACKNOWLEDGMENT

This project is supported by National High Technology Research and Development Program of China (2008AA092301); Postdoctral Science Foundation of Chinese Hei Long-jing province(323630295)

REFERENCES

[1] Adam J.A. Probing Beneath the Sea [J]. IEEE spectrum,1985,22(4): 55~64.

[2] Xu Yu-Ru , Xiao Kun. Technology Development of Autonomous Ocean Vehicle. Acta Automatica Sinica. 33(5): 2007: 518~521.

[3] Li Ye, Zhang Lei, Wan Lei, Liang Xiao. Optimization of S-surface Controller for Autonomous Underwater Vehicle with Immune-genetic algorithm. Journal of Harbin Institute of Technology (New Series). 2008, 15(3):404~410

[4] B. Buckham, F.R. Driscoll and M. Nahon, "Development of a Finite Element Cable Model for Use in Low-Tension Dynamics Simulation," Journal of Applied Mechanics, (71), 2004: 476~485.

[5] F.R. Driscoll, R.G. Lueck, M. Nahon. Development and Validation of a Lumped-mass Dynamics Model of a Deep-sea ROV System. Applied Ocean Research, (22), 2000: 169~182

[6] Tang Xu-Dong, Pang Yong-Jie, Li Ye. Process Neural Control Algorithm for Autonomous Underwater Vehicle. Control Theory and Applications. 26(4): 420~424

[7] Tang Xudong, Pang Yongjie, Li Ye, Qing, Zaibai. A Fuzzy Neural Networks Controller of Underwater Vehicles based on Ant Colony Algorithm. Proceedings of the 27th Chinese Control Conference. Kunming, Yunnan, China. 2008.7: 637~641

[8] R. J. Wai, C. M. Lin, and Y. F. Peng, "Adaptive Hybrid Control for Linear Piezoelectric Ceramic Motor Drive Using Diagonal Recurrent CMAC network," IEEE Trans. Neural Netw., 2004, 15(6) : 1491~1506.

[9] S. J. Yoo, Y. H. Choi, and J. B. Park, "Generalized Predictive Control Based on Self-recurrent Wavelet Neural Network for Stable Path Tracking of Mobile Robots: Adaptive learning rates approach," IEEE Trans. Circuit Syst., 2006, 53(6): 1381~1394.

[10] F. J. Lin, R. J. Wai, and C. C. Lee, "Fuzzy Neural Network Position Controller for Ultrasonic Motor Drive Using Push–pull DC–DC Converter," in Proc. Inst. Elect. Eng.—Control Theory Appl., 1999, 146(1) : 99~107.

[11] C. Ye, N. H. C. Yung, and D. Wang, "A Fuzzy Controller With Supervised Learning Assisted Reinforcement Learning Algorithm for Obstacle Avoidance," IEEE Trans. Syst., Man, Cybern. B, Cybern., 2003, 33 (1): 17-27.

[12] R. J. Wai and C. C. Chu, "Motion Control of Linear Induction Motor via Petri Fuzzy-neural-network," IEEE Trans. Ind. Electron., 2007, 54 (1): 281~295.

[13] Tang Xu-Dong, Pang Yong-Jie, Zhang He, Lu Guo-Chun. Improved PSO-based S Plane Controller for Motion Control of Underwater Vehicle. 2008 Chinese Control and Decision Conference, Yantai, Shandong, China. 2008.7 : 2843~2848

[14] Wang Jian-Guo, Wu Gong-Xing, Wan Lei, Sun Yu-Shan, Wang Li-Rong. Controller Design of Underwater Robots Based on Generalized S-plane. Electric Machines and Control. 2009, 13(1): 144~148

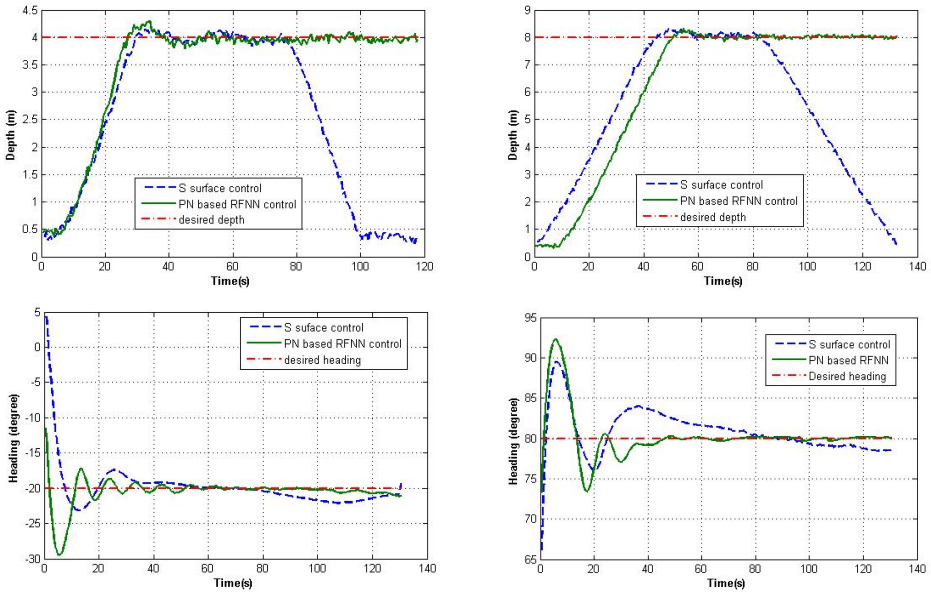

Figure 6. Experimental results and comparisons