Fault-Tolerant Speed Control of a Biomimetic Multijoint Robotic Fish

Yueqi Yang[1,2], Zhengxing Wu[1], Junzhi Yu[1]
[1] State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
{yangyueqi2016, zhengxing.wu, junzhi.yu}@ia.ac.cn

***Abstract*—Fault tolerance is critical to the real applications of underwater robots in complex underwater environments. In this paper, a biomimetic robotic fish with a stuck tail joint is used as the control object, and the fault-tolerant control method is proposed to control the faulty robotic fish to reach the specified speed. Specifically, based on the yaw fault-tolerant control, we firstly analyze the coupling problem between the control parameters of the robotic fish and design the feedback controller based on the central pattern generator model. In order to solve the problem that the feedback controller converges slowly, a multi-layer perceptron-based feedforward controller is proposed. Simulation experiments are designed to test the performance and effectiveness of the control system. Simulation results reveal that the proposed fault-tolerant control method is able to realize the speed control of faulty robotic fish while maintaining the yaw state stability.**

***Index Terms*—Fault-tolerant control, underwater robots, robotic fish, speed control, multi-layer perceptron.**

I. INTRODUCTION

With the rapid development of robotics, biomimetics have attracted a great number of researchers' attention. By studying and analyzing the shape and motion of organisms in nature, inspirations can be provided to optimize the performance of artificial systems. In the underwater bionics field, the analysis of fish's high efficiency and superior swimming performance is one of active research topics nowadays. Therefore, with the increasing demand for maneuverability and efficiency of autonomous underwater vehicles (AUV), biomimetic robotic fish has received extensive attention. As a new type of underwater robot that mimics fish has high mobility and high flexibility, robotic fish can be applied to more complex underwater environments [1],[2]. In addition, the robotic fish also has the advantages of low noise, harmless to the underwater ecosystem, and so on.

At present, researchers have built a variety of robotic fish, and carried out a lot of work on the swimming performance, mechanical structure, and motion control of robotic fish [3]–[5]. However, due to the complexity of robotic fish system, how to improve its stability is still a problem at this stage. Specifically, the robotic fish works in the complex underwater environment, and factors such as undercurrents, fish, reefs, and aquatic plants increase the risk of damage to the robotic fish, which undoubtedly tests the reliability of the robotic fish system. On the other hand, the robotic fish mimics the highly maneuvering behavior of fish. For example, by analyzing the high-speed C-type starting characteristics of fish, Liu *et al.* designed the equation of motion for the C-type steering of robotic fish [6]. Su *et al.* proposed a C-type start control method based on the dynamic trajectory method to increase the peak steering speed in the plane of robotic fish to 670°/s [7]. Conte *et al.* simulated the S-type start mode of the pike [8]. These highly mobile motion modes also enhance the load on the robotic drive and increase the rate of failure. In addition, unlike the traditional AUV, due to the coordination of multiple joints, the movement of the robot fish depends on the stability of the robotic fish system, and the failure of any joint of the robot fish can seriously affect its athletic performance. The stability and robustness of the robotic fish system are important to guarantee practical applications of robotic fish.

To tackle the problem that the fault is easy to occur, a conventional method is to design a fault-tolerant control method to make the system achieve satisfactory performance even if the fault occurs [9]–[11]. However, most of the existing underwater fault-tolerant control study is aimed at the traditional propeller-driven underwater vehicle, mainly focusing on two aspects: fault-tolerant control for sensors on AUV and for motion actuators. Researches on fault-tolerant control methods for multi-link robotic fish are still missing. The strong coupling between the control mechanisms of the robotic fish makes it very sensitive to faults. The impact of faults on the robot fish is mainly reflected in the swimming direction and speed. In particular, the swimming speed of the robot fish directly affects the efficiency of its mission completion. Therefore, it is of great practical significance to study the speed fault-tolerant control method of robotic fish.

Considering the characteristics of the underwater motion of the robotic fish, there is a great delay between the change of the control signal and the stability of the motion. It is difficult to obtain the desired effect by relying solely on feedback control. Therefore, designing control methods based on dynamic model is one of commonly methods. According to the dynamic model, the swimming states of the robotic fish can be obtained from the control parameters. Even so, it is still a very difficult problem to obtain control parameters through the swimming states due to the nonlinearity and time dependence of the dynamic model. Multilayer perceptron, as a data-driven function fitting method, can effectively correspond to the correspondence between input and output of data sets [12]. Nevertheless, there are still many difficulties in the use of multilayer perceptrons, such as overfitting, data continuity requirements, and so on. Fitting the relationship between the target swimming states and the control parameters by the multi-layer perceptron is an effective way to solve the current problem.

The main objective of this paper is to optimize the swimming speed of the multijoint robotic fish whose tail joint is stuck, and to make the robotic fish still have satisfactory swimming performance in the fault state. The innovations and contributions of this paper are mainly divided into the following two aspects. First, in order to keep the robotic fish in a stable direction at the fault state, a fault-tolerant control method based on dynamic model and central pattern generator (CPG) is used to correct the direction of swimming. On this basis, we analyze the coupling relationship between speed control parameters and direction control parameters, and create a feedback controller to maintain the robot fish stable. Second, considering the large delay characteristics of speed regulation, this paper calculates multiple sets of data from control parameters, fault states to swimming speeds through the dynamics model. By using those data sets as a training set, a multi-perceptor is trained. The perceptron uses the desired speed and fault status as inputs and the control parameters as outputs. The speed is controlled by using the model obtained by the perceptron as the open-loop controller. Finally, the simulation experiment is given to compare the effects of different fault-tolerant control methods and the feasibility of the proposed method. Different from previous work, the focus of this paper is on the speed correction of faulty robot fish. By analyzing its motion pattern, a speed fault-tolerant control method for robotic fish is first proposed. The method we propose further improves the robustness and stability of the robotic fish system.

The rest of this paper is organized as follows: The mechanical structure and motion control of the robotic fish is introduced in Section II. Section III introduces the direction fault-tolerance control and describes the feedback controller for speed based on the CPG model, and then analysis are given by the simulation results. The feedforward control method based on multi-layer perceptron is introduced in detail in Section IV, followed by simulation experimentsFinally, Section V provides conclusions and future work.

II. Mechanical Design And Motion Control

In the previous work of our laboratory, a multijoint robotic fish was developed. The mechanical structure of the robotic fish mimics the design of *Esox Lucius*, as is shown in Fig. 1. The self-propelled body of the robotic fish consists of four tail joints and a caudal fin. The tail joints are driven by servomotors. Through the cooperation of the tail joints and the caudal fin, the robotic fish can reach a peak steering speed of 670°/s and an average steering speed of 490°/s. In addition, a pair of two-degree-of-freedom pectoral fins were designed to meet the requirement of three-dimensional motion. The fault-tolerant control method studied in this paper is aimed at two-dimensional space plane motion, so the pectoral fin will be fixed to the initial state. The control unit of the robotic fish is mounted in the head. When the tail joint is stuck, the swimming speed and yaw of the robotic fish are seriously affected. In order to detect data such as the yaw of the robotic fish, an inertial measurement unit (IMU) is installed in the control unit. The fault-tolerant control method proposed in this paper uses this robot fish as the controlled object.

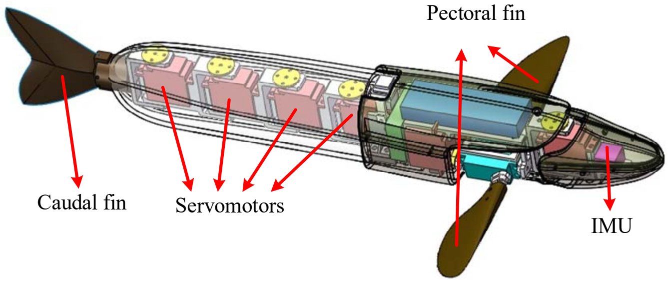


Fig. 1. Mechanical structure of the robotic fish.

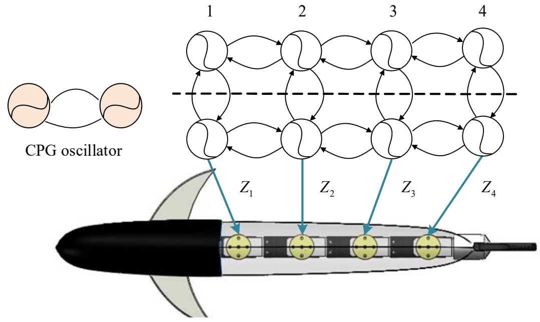


Fig. 2. Topological structure of the CPG model.

The fish body wave fitting method and the CPG model method are two commonly used robotic fish motion control strategies [13]. Inspired by the bio-CPG network [14], the CPG model generates a rhythm control signals through neuron oscillators. Compared with the fish body wave fitting method, the CPG model method has the advantages of smooth state switching, strong stability, and so on. Therefore, the robotic fish used in this paper adopts the CPG as the motion control method. Specifically, a CPG model with offset and phase difference is used [15], and its structure is shown in Fig. 2. The state equation of the CPG model is shown as follows:

$$
\begin{cases}
\dot{x}_i = -\omega_i\left(y_i - b_i\right) + x_i\left[r_i^2 - x_i^2 - \left(y_i - b_i\right)^2\right] \\
\quad +h_1\left[x_{i-1}\cos\psi_i + \left(y_{i-1} - b_{i-1}\right)\sin\psi_i\right] \\
\dot{y}_i = \omega_i x_i + \left(y_i - b_i\right)\left[r_i^2 - x_i^2 - \left(y_i - b_i\right)^2\right] \\
\quad +h_2\left[x_{i+1}\sin\psi_i + \left(y_{i+1} - b_{i+1}\right)\cos\psi_i\right] \\
z_i = c_i y_i
\end{cases}
\tag{1}
$$

where i indicates the i th oscillator ($i = 1,\ldots,4$); x_i and y_i represent the state variables of the i th oscillator; ψ_i denotes the phase difference of adjacent neurons; r_i and ω_i represents the amplitude and angular frequency of the i th oscillator, respectively; b_i denotes the bias of the i th oscillator; h_1 and h_2 are the coupling coefficients between adjacent neurons; c_i is a constant coefficient; and z_i represents the output signal of the i th oscillator.

The motion state of the robotic fish can be controlled by adjusting the CPG parameters. For example, by adjusting b_i, the yaw direction of the robotic fish can be changed; ω_i affects the swimming speed of the robotic fish; a reasonable configuration of ψ_i can make the robotic fish swim towards the opposite direction. As mentioned before, when the tail joint is stuck, the swimming direction and speed of the robotic fish will be seriously affected. Therefore, this paper focuses on b and ω_i, and the other parameters are reasonably configured [15]. In addition, for stability and simplicity, we set $\omega_1=\omega_2=\omega_3=\omega_4=\omega$ and $b_1=b_2=b_3=b_4=b$.

III. Speed Control of Faulty Robot Fish Based on CPG Model

If the yaw direction of the robotic fish is incorrect, it makes no sense to optimize the swimming speed. Therefore, to fulfil the task of speed optimization, the impact of the fault on yaw must first be dealt with. In our previous work, a yaw fault-tolerant control method based on CPG model and dynamic model was proposed, and its control block diagram is shown in Fig. 3.

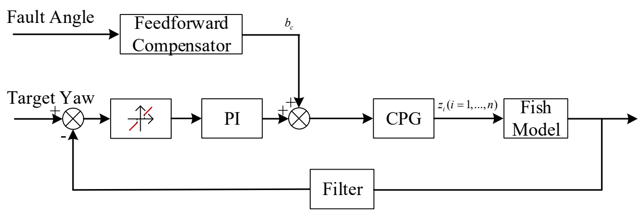


Fig. 3. Block diagram of the yaw fault-tolerant control system.

The proposed method is mainly divided into two parts: a feedback controller and a feedforward compensator. The feedback controller uses the CPG offset b as the control variable and the filtered yaw collected by the IMU as feedback. By reasonably setting the controller parameters and introducing the threshold, the yaw control of the faulty robot fish is finally realized. The mathematical expression of the feedback controller is described below.

$$\begin{cases} b(t)=K_{bP}\,e(t)+K_{bI}\int_0^t e_b(u)\,\mathrm{d}u \\ e_b(t)=\begin{cases}\hat{\gamma}-f\gamma(t), & \left|\hat{\gamma}-f\gamma(t)\right|\geq T_\gamma \\ 0, & \left|\hat{\gamma}-f\gamma(t)\right|<T_\gamma\end{cases}\end{cases} \tag{2}$$

where t denotes the moment, $b(t)$ denotes the bias of CPG at time t, $f\gamma(t)$ is the filtering result of the yaw angle at time t, $\hat{\gamma}$ is the target yaw, $e_b(t)$ denotes the PI controller input at time t, T_γ represents the threshold, and K_{bP} and K_{bI} are the feedback gains.

In order to optimize the convergence time of the yaw fault-tolerant control method, we design a feedforward compensator based on the dynamic model, its mathematical expression is described as follows:

$$b_c=\arg\min_{b_0}\left\{\left\|\int_{t=0}^{T}\int_{u=0}^{t}\Gamma_6\left(u,\beta,b_0,f\right)\mathrm{d}u\mathrm{d}t\right\|\right\} \tag{3}$$

where b_c is the compensation offset; f and β represent the fault location and the fault angle, respectively; Γ_6 denotes the torque in the vertical direction of the head; $T=2\pi/\omega$ is the period; and u is a variable.

The feedforward compensator calculates the compensation offset through the optimization function, which accelerates the convergence speed of the yaw fault-tolerant control method. Fig. 4 shows the experimental results of yaw fault-tolerant control. As can be seen, the yaw fault-tolerant control method can effectively reduce the impact of faults on the yaw of the robot fish. In addition, as mentioned in Section 2, the swimming speed of the robotic fish can be controlled by ω. Therefore, on the basis of the yaw fault-tolerant control, it is feasible to control the robotic fish with ω as the parameter to reach the target swimming speed of the faulty robotic fish. It is worth mentioning that this article focuses on v_x which is the speed of the head of the robotic fish.

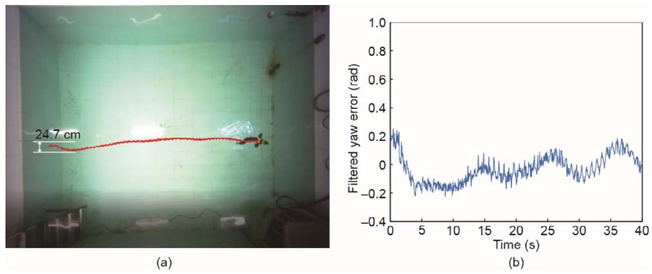
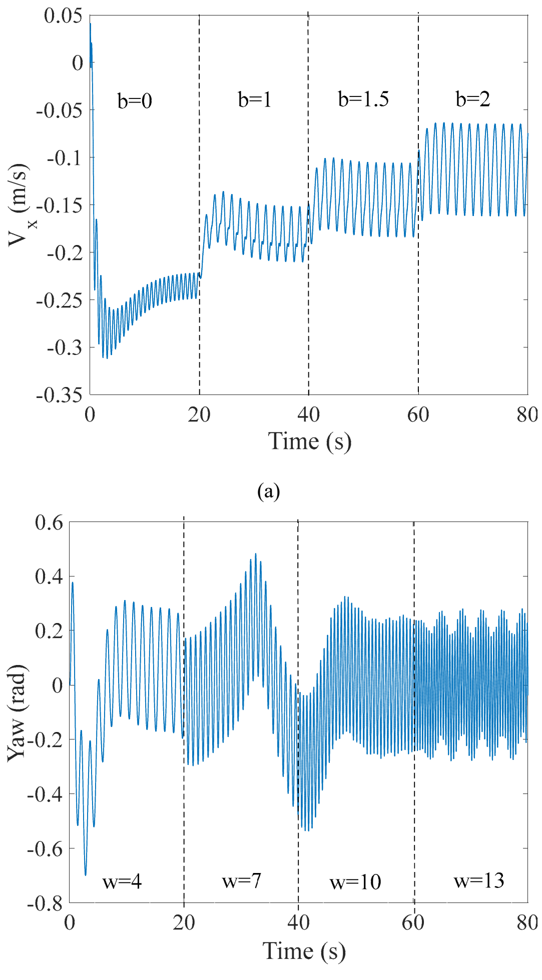


Fig. 4. Experimental results of yaw fault-tolerant control. (a) Swimming path; (b) Filtered yaw error.

Considering that there may be coupling between CPG parameters, and coupling can seriously affect the performance of multivariable feedback control systems. Therefore, we designed a simulation experiment to analyze the effect of b_i on v_x and ω on the yaw. In our previous work, a data-driven dynamic model was built for a similar robotic fish [16]. In the simulation experiment, the dynamic model is applied to replace the robotic fish. The results are shown in Fig. 5. Fig. 5 (a) shows the relationship of b_i and v_x, in this simulation, we set ω=4. It can be seen that the change in b causes the swimming speed to change. Fig. 5 (b) shows the effect of on the yaw of the robotic fish with fault-tolerant control. The faulty condition is set to β=30° and f=3 for all simulation in this paper. Although ω has some effect on yaw, this effect will be corrected under the influence of yaw fault-tolerant control. It can be seen from the data of the 40 s to 60 s that when the yaw direction is stable, ω hardly affects yaw. In addition, there is a delay between the change from ω and the stability of v_x state.



(b)

Fig. 5. Results of coupling analysis simulation experiment. (a) Relationship between b and v_x; (b) Relationship between ω and yaw.

Based on the above analysis, a proportional integral (PI) feedback controller with stable judgment switch is designed as follows:

$$\begin{cases} \omega(t) = K_{\omega P} e_{\omega P}(t) + K_{\omega I} \int_0^t e_{\omega I}(u)\,\mathrm{d}u \\ e_{\omega P}(t) = \begin{cases} \hat{v}_x - fv_x(t), & \textit{stability conditions} \\ \textit{hold}, & \textit{others} \end{cases} \\ e_{\omega I}(t) = \begin{cases} \hat{v}_x - fv_x(t), & \textit{stability conditions} \\ 0, & \textit{others} \end{cases} \end{cases} \tag{4}$$

where $\hat{v}_x$ and $fv_x(t)$ denote the target speed the filtering result of the speed at time t, respectively; $K_{\omega P}$ and $K_{\omega I}$ are the feedback gains; $e_{\omega P}(t)$ and $e_{\omega I}(t)$ are the PI controller input at time t; *hold* means to keep the previous value; and stability conditions are defined as all of the following conditions are met at the same time:

$$\begin{cases} \Delta(\hat{v}_x - fv_x(t)) < T_v \\ \Delta(b) < T_b \end{cases} \tag{5}$$

where T_v and T_b denotes the thresholds of v_x and b, respectively; $\Delta(f(t))$ means the average change of $\bullet$ over a previous period, the calculation method of $\Delta(f(t))$ is detailed in Eq. 6.

$$\Delta(f(t)) = \frac{1}{n-1} \sum_{j=1}^{n-1} \left| \int_{t-\frac{2\pi(n-j)}{\omega(t)}}^{t-\frac{2\pi(n-j-1)}{\omega(t)}} f(u)du - \int_{t-\frac{2\pi(n-j+1)}{\omega(t)}}^{t-\frac{2\pi(n-j)}{\omega(t)}} f(u)du \right| \tag{6}$$

where n denotes the number of period included in sampling time, and is set as n=8 in this paper.

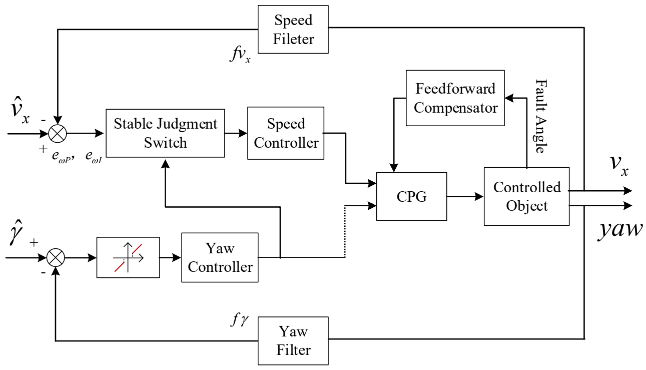
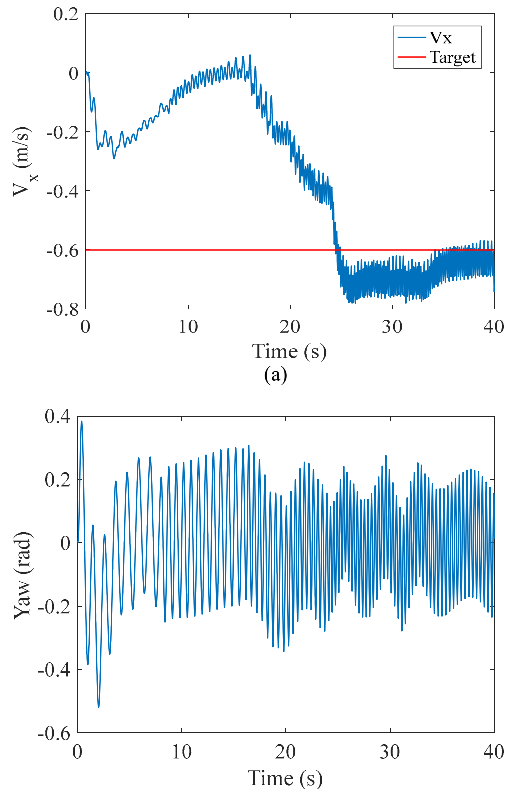


Fig. 6. Block diagram of the fault-tolerant control method with speed feedback.



(b)

Fig. 7. Simulation result of the fault-tolerant control system with speed feedback. (a) Speed; (b) Yaw angle.

The stability judgment switch ensures that when ω changes, b and speed output of the object are in a stable state. The fault-tolerant control block diagram with speed feedback is shown in Fig. 6. Further, Fig. 7 illustrates the simulation output of the control system presented above. As can be observed, the fault-tolerant control method with speed feedback can effectively control the faulty robotic fish to reach the target speed and maintain yaw stability. However, the adjustment time of speed exceeds 30 s. This is mainly because the stability judgment switch greatly extends the speed control period. In order to ensure system stability, speed control will only be implemented with speed and yaw stability.

IV. Multilayer Perceptron based Speed Control for Faulty Robotic Fish

Cooperating with the feedforward controller is one of the common methods to speed up the convergence of the feedback controller. When the model, environment, and other control parameters are fixed, the relationship between ω and v_x is determined. According to the data test, the relationship between the expected offset of the faulty robotic fish and the angle of failure is largely monotonous. Therefore, the desired angular frequency ω_d can be expressed as a function of the angle of failure, the location of the fault, and the target speed, as is shown in Eq. 7.

$$\omega_d = g(\beta, f, \hat{v}_x) \tag{7}$$

Due to the nonlinearity and time-varying of the robotic fish system, it is difficult to solve the correspondence *g*. However, the dynamic model of the robotic fish makes it easy to obtain the swimming state through the control parameters [16]. Therefore, it is feasible to collect multiple sets of data through the dynamic model and fit *g* through a multilayer perceptron.

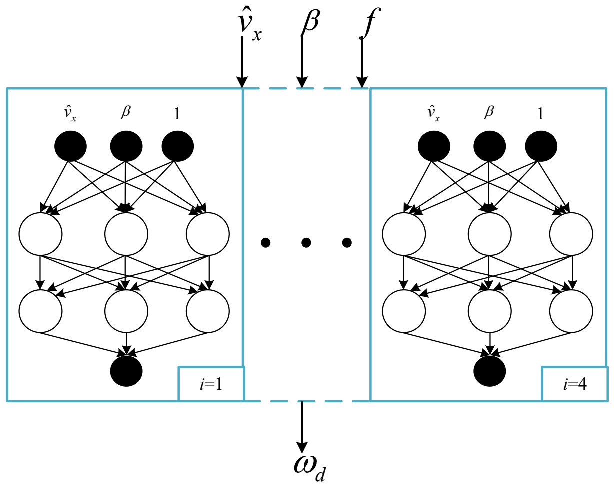


Fig. 8. Topology structure of multi-layer perceptrons.

The multilayer perceptron defaults its input and output are continuous but $f = 1,...,4$ is discrete. As a matter of fact, the change of the position of the faulty joint is equivalent to changing the robotic fish model. Therefore, it is more reasonable to train four multi-layer perceptrons according to different faulty joint positions. Topology structure of multi-layer perceptrons is shown in Fig. 8. The four perceptrons have the same topology, and the input layer consists of $\hat{v}_x$, β and a constant offset. Each perceptron has two hidden layers, each containing three neurons. The output of the perceptron is ω_d. Using the trained multi-layer perceptron as a feedforward controller, the system block diagram of the whole system is shown in Fig. 9.

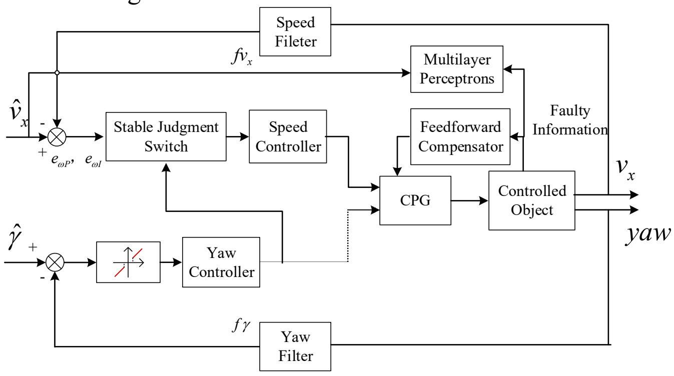


Fig. 9. Block diagram of the whole fault-tolerant control method.

Fig. 10 shows the simulation results of the fault-tolerant control with multi-layer perceptorns only. In the simulation environment, the multi-layer perceptron accurately obtains ω_b from the input. It should be noted that the parameters used in the simulation did not appear in the training set of perceptrons. Compared with the feedback control method proposed in Section III, the feedforward controller has the advantages of correspondingly faster and less impact on yaw. However, since the dynamic model is established under ideal conditions, and the actual underwater environment is complicated, the robot fish is easily disturbed. Therefore, it is difficult for the open-loop controller to achieve the effect in the simulation experiment. The simulation result of the complete fault-tolerant control method is shown in Fig. 11. The convergence time of the system is shortened to 10 s, and the yaw angle is stable. Physical experiments are underway.

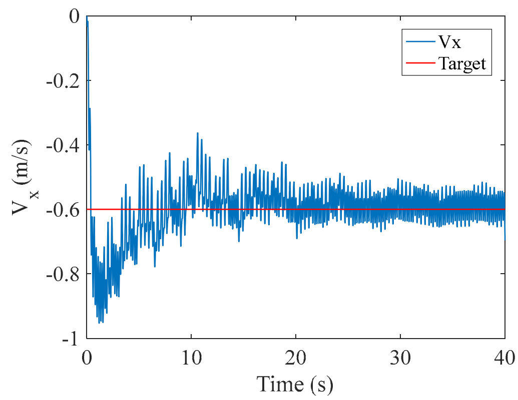


Fig. 10. Simulation output of fault-tolerant control with multi-layer perceptorns only.

V. Conclusions and Future Work

In this paper, we have proposed a fault-tolerant control method for controlling the swimming speed of faulty robotic fish. On the basis of yaw fault-tolerant control, the feedback controller based on CPG model is designed by analyzing the coupling relationship between control parameters. In order to further optimize the performance of speed control, we use a multi-layer perceptron-based feedforward controller by using the data obtained from the dynamic model as a training set. Simulation experiments demonstrate that the multi-layer perceptron can accurately fit the relationship between control parameters and target speed. The convergence time of the fault-tolerant control method with feedforward controller added has been significantly shortened.

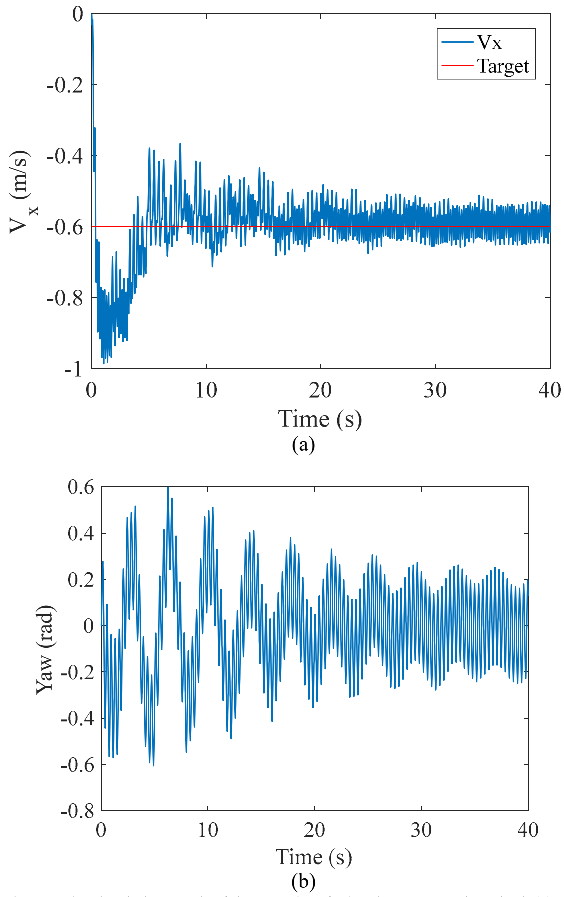


Fig. 11. The simulation result of the complete fault-tolerant control method. (a) Speed; (b) Yaw angle.

In the future, we plan to extend the fault-tolerant control method to three-dimensional space. We will also intend to invest in troubleshooting.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grants 61633017, 61603388, and 61725305, and in part by the Key Research and Development and Transformation Project of Qinghai Province under Grant 2017-GX-103.

REFERENCES

[1] C. Zhou, Research on modeling, control and cooperation of a marsupial biomimetic robotic fish, [dissertation]. Beijing: Institute of Automation, Chinese Academy of Sciences, 2008.

[2] J. Yu, Research on Control and Coordination of Multiple Bio-mimetic Robot Fishes, [dissertation], Beijing: Institute of Automation, Chinese Academy of Sciences, 2003.

[3] J. Yu, S. Chen, Z. Wu, and W. Wang, “On a miniature free-swimming robotic fish with multiple sensors,” *Int. J. Adv. Robot. Syst.*, pp. 13–65, 2016.

[4] J. Yu, M. Tan, S. Wang, and E. Chen, “Development of a biomimetic robotic fish and its control algorithm,” *IEEE Trans. Syst. Man. Cybern. B. Cybern.*, vol. 34, no. 4, pp. 1798–1810, 2004.

[5] J. Liang, T. Wang, and L. Wen, “Development of a two-joint robotic fish for real-world exploration,” *J. Field. Robot.*, vol. 28, no. 1, pp. 70–79, 2011.

[6] C. Zhou and K. Low, “Design and locomotion control of a biomimetic underwater vehicle with fin propulsion,” *IEEE/ASME Trans. Mechatronics*, vol. 17, no. 1, pp. 25–35, 2012.

[7] J. Liu and H. Hu, “Mimicry of sharp turning behaviours in a robotic fish,” in *Pro. 2005 IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, Apr. 2005, pp. 3318–3323.

[8] J. Conte, Y. Modarres-Sadeghi, M. N. Watts, F. S. Hover, M.S. Triantafyllou, “A fast-starting mechanical fish that accelerates at 40 ms−2,” *Bioinspir. Biomim.*, vol. 5, no. 3, p. 035004, 2010.

[9] Z. Luo, J. Shang, and Z. Zhang, “Innovative design of six wheeled space exploration robot using module combination,” in *Proc. 19th Int. Conf. Mechatronics Mach. Vis. Practice*, Auckland, New Zealand, New York, Nov. 2012, pp. 460–465.

[10] K. Chen, M. Kamezaki, and T. Katano, “Fundamental development of a virtual reality simulator for four-arm disaster rescue robot OCTOPUS,” in *Proc. 2016 IEEE Int. Conf. Adv. Intell. Mechatronics*, Banff, AB, Canada, Canada, Jul. 2016, pp. 421–426.

[11] R. Abbaspou, “Design and implementation of multi-sensor based autonomous minesweeping robot,” in *Proc. 2010 Int. Congr. Ultra Modern Telecommun. Control Syst. Workshops*, Moscow, Russia, New York, 2010, pp. 443–449.

[12] H. E. Yuan-Tao and W. Deng, “The compare between three training algorighms of multilayer perceptorns,” *J. Suzhou Univ.*, 2008.

[13] D. A. McCrea and I. A. Rybak, “Organization of mammalian locomotor rhythm and pattern generation,” *Brain Res. Rev.*, vol. 57, no. 1, pp. 134–146, 2008.

[14] J. E. Rubin, N. A. Shevtsova, G. B. Ermentrout, J. C. Smith, and I. A. Rybak, “Multiple rhythmic states in a model of the respiratory central pattern generator,” *J. Neurophysiol.*, vol. 101, no. 4, pp. 2146–2165, 2009.

[15] J. Yu, Z. Wu, M. Wang, and M. Tan, “CPG network optimization for a biomimetic robotic fish via PSO,” *IEEE Trans. Neural. Netw. Learn. Syst.*, vol. 27, no. 9, pp. 1962–1968, 2016.

[16] J. Yu, J. Yuan, Z. Wu, and M. Tan, “Data-driven dynamic modeling for a swimming robotic fish,” *IEEE Trans. Ind. Electron.*, vol. 63, no. 6, pp. 5632–5640, 2016.