Ocean Engineering 140 (2017) 453–465

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

On intelligent risk analysis and critical decision of underwater robotic vehicle

Xianbo Xiang[a,*], Caoyang Yu[a], Qin Zhang[b]

[a] *School of Naval Architecture and Ocean Engineering, Huazhong University of Science and Technology, 1037, Luoyu Road, Wuhan 430074, China*
[b] *State Key Lab of Digital Manufacturing, Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China*

ARTICLE INFO

Keywords:
Underwater robotic vehicle (URV)
Mamdani fuzzy neural network (MFNN)
Risk analysis
Critical decision
Onboard safety

ABSTRACT

The marine community has witnessed a remarkable growth of underwater robotic vehicles (URVs) for undersea exploration and exploitation in recent decades. Yet, it is critical to intelligently diagnose the fault and evaluate the risk of the onboard system, and render critical decision to ensure the safety of the URV with high-value assets. In this paper, a dedicated two-layer fault treatment system including risk analysis subsystem and intelligent decision subsystem is proposed to enhance the onboard safety of the URV. First, a hierarchical fault tree model of the URV is built by integrating the state information of sensors, actuators and running status. Second, in the risk analysis subsystem, the onboard system risk is analyzed based on the adaptive learning and fuzzy inference capabilities of the Mamdani fuzzy neural network (MFNN). Third, in the safety decision subsystem, the risk level of the URV is evaluated by adopting the maximum membership and threshold principles, which enables the intelligent decision to take critical operation and ensure the safety of the URV. Finally, the proposed fault treatment system is validated by numerical simulation and hardware in loop test. Experimental results demonstrate the feasibility and efficiency of the intelligent fault treatment system for the URV.

1. Introduction

Although the vast ocean is rich in biological organisms, chemicals, mineral deposits, natural gas and petroleum resources (Blidberg et al., 1991), the oceanic environment is quite harsh and hostile such that human beings are not capable of diving into deep sea or staying underwater for long time intervention. As an emerging marine vehicle without human being onboard, underwater robotic vehicle (URV) with persistent autonomy plays an important role in undersea activities (Roberts and Sutton, 2006), and it is widely used for accomplishing underwater missions, such as offshore oil/gas exploration and exploitation (Xiang et al., 2015a), oceanographic survey (Choyekh et al., 2015), deepsea sampling (Xiang et al., 2015b), environmental assessment (Arshad, 2009), subsea cable/pipeline inspection (Xiang et al., 2016a), and underwater intervention (F. Zhang et al., 2015).

Yet, when the URV navigates in the complex and harsh undersea environment, one fault of the onboard system might threaten the safety of the URV if there are no risk analysis and emergent decision functions onboard. In the case of minor system fault, the URV cannot successfully accomplish underwater tasks. However, a serious fault will results in a terrible catastrophe that the URV is missing or cannot be recovered. The loss accidents of underwater vehicle KAIKO and Nereus brought the underwater robotic community huge losses in history (Momma et al., 2004; Showstack, 2014)). Hence, as a paramount tool for replacing human being to explore and exploit undersea resources, autonomy and safety are the most critical requirements for the URV. With the continuous improvements of URV's functions, the complexity of the onboard system is also increasing and its safety attracts more and more attentions, which demands the development of fault diagnosis for thrusters, decision making and fault-tolerant control in order to guarantee the safety of the vehicle (Lazakis et al., 2010, 2016; Zhang, 2012; Chu and Zhang, 2014).

In fact, onboard sensors and thrusters carried by the URV are necessary for underwater activities but they are also the main causes of faults that at least destroy the successful mission accomplishment, and worst of all, pose a huge threat to the safety of the URV (Zhu and Sun, 2013). According to the concept of device and system, the faults of the URV in this paper are classified into two categories:

(1) Device-level fault: This mainly refers to single device fault, for instance, thruster failure, or a certain type of device faults resulted by the hostile oceanic environment or internal unknown faulty factors.

(2) System-level fault: This specifically refers to the comprehensive

---

[*] Corresponding author.
E-mail addresses: xbxiang@hust.edu.cn (X. Xiang), ycyhust@hust.edu.cn (C. Yu), qin.zhang@hust.edu.cn (Q. Zhang).

http://dx.doi.org/10.1016/j.oceaneng.2017.06.020
Received 1 January 2017; Received in revised form 24 March 2017; Accepted 7 June 2017

fault as a consequence of sensor fault, thruster fault, communication interruption, over-depth, timeout and other faulty aspects in the device-level. Hence, it has a significant impact on the URV system and onboard safety.

In recent decades, some intelligent methods have been adopted to diagnose the device fault of the URV, especially to single onboard thruster with a partial fault, multi-state fault or continuous state fault. For example, fault diagnosis of thrusters was accomplished by evaluating any significant change in the behavior of the vehicle via the extended Kalman filter estimator onboard (Alessandri et al., 1999). Liu et al. proposed the fault diagnosis method based on cerebellar model articulation controller (CMAC) neural network to obtain online learning for the underwater thrusters in Liu et al. (2012). Zhu et al. achieved information fusion and fault diagnosis based on credit assignment-based fuzzy-CMAC neural network in the OUTLAND underwater vehicle (Zhu and Sun, 2013). An improved fractal feature integrated with wavelet decomposition identification method was proposed to overcome the identification error and its inconsistency (Liu et al., 2016). Furthermore, thruster fault detection, isolation and identification in autonomous underwater vehicle were completed based on grey qualitative simulation (M.-J. Zhang et al., 2015).

In addition, some researchers embarked on fault diagnosis for multiple actuators by detecting the state of each one in order to enable fault-tolerant control. Tarun et al. considered that the thruster of ODIN underwater vehicle has a fault if the different between the desired output voltage and the actual one is larger than a predefined threshold. Subsequently, the weighted pseudo-inverse method was used to reallocate the thruster forces and achieve the fault-tolerant control function (Podder and Sarkar, 2001). Sarkar et al. presented a redundancy resolution scheme that considers the presence of an excess number of thrusters along with any thruster fault and determines the reference thruster forces to produce the desired motion (Sarkar et al., 2002)). Edin et al. proposed a two-layer framework consisting of fault diagnosis subsystem and fault accommodation subsystem. The former monitored the state of each thruster by integration of self-organizing maps and fuzzy logic clustering methods. The latter used information provided by the fault diagnosis subsystem to perform control reallocation (Omerdic and Roberts, 2004). In addition, fuzzy logic, sliding mode and neural network algorithms were widely applied to fault-tolerant control of the URV (Akmal et al., 2012; Wang et al., 2015).

Above research activities only consider the thruster fault in the device-level, and few fault diagnosis of onboard URV system are completed by considering both thruster and sensor faults. Takai et al. adopt the method of artificial neural network to build the dynamics model of Twin-Burger underwater vehicle, and used model matching part to detect the output value of actual sensors. Yet, it was assumed that no more than one kind of sensor or thruster fault occurred at any time (Takai and Ura, 1999). Zhang et al. proposed a fuzzy weighted support vector domain method that can effectively solve the problem of low classification accuracy in the process of multi-fault pattern classification owing to the effect of sparse density and uneven distribution of fault samples. However, this method only diagnoses if there occurs a fault of the thrusters or sensor, and the risk level of the whole URV system can not be determined (Zhang et al., 2014). In addition, Richard et al. adopted the Livingstone 2 (L2) diagnosis engine on Autosub6000 vehicle without taking into account interactions among components of the onboard system (Dearden and Ernits, 2013).

Hence, it can be concluded that various intelligent methods have been developed and applied to diagnose the device-level fault of the URV in recent research activities. However, multi-class faults of the URV system, such as thruster fault, sensor fault, communication interruption and other aspects, are not considered as an integrated faulty system to carry out risk analysis. It means that comprehensive fault analysis in the system-level is not taken into account. In addition, although the fault-tolerant control can be designed after fault diagnosis in the previous research, it is not able to deal with the serious system-level fault which will heavily threaten the safety of the URV. In this sense, besides classic onboard fault analysis, making intelligent decision and taking emergency operation should be taken into account as well to ensure the safety of the URV. Compared with the classic fault-tolerant control, emergency operation is at the expense of the work capacity of the URV to some extent, but it reduces the risk of the loss and guarantees the safety of the URV.

Motivated by above considerations, this paper takes into account all the state information of actuators, sensors and the running status of the URV to build fault tree model. Subsequently, a two-layer fault treatment system including risk analysis subsystem and intelligent decision subsystem is proposed to evaluate the system risk of the URV, and then render critical to ensure the safety of the URV by taking emergency operations. In this paper, the main contributions are fourfold as follows:

(1) The hierarchical fault tree model of the URV is built by integrating all potential fault sources in the device-level, including actuator fault, sensor fault, communication interruption, over-depth, power supply fault, water leakage and operation timeout.

(2) The risk analysis subsystem based on the Mamdani fuzzy neural network (MFNN) method is proposed to identify the system risk of the URV from device-level faults. The MFNN method is adopted to make full use of the adaptive learning capability of neural network by learning a small amount of samples offline, and then completes the online rapid fuzzy inference to diagnose the system fault.

(3) The proposed intelligent decision subsystem applies the maximum membership and threshold principles to evaluate the system risk of the URV, and then synthesizes expert knowledge rules to render intelligent decision and take critical operations, in order to ensure the safety of the URV.

(4) Numerical simulations in MATLAB and experiments of hardware in loop test on the VxWorks real-time operating system (RTOS) validate the feasibility and efficiency of the proposed two-layer fault treatment system for the URV.

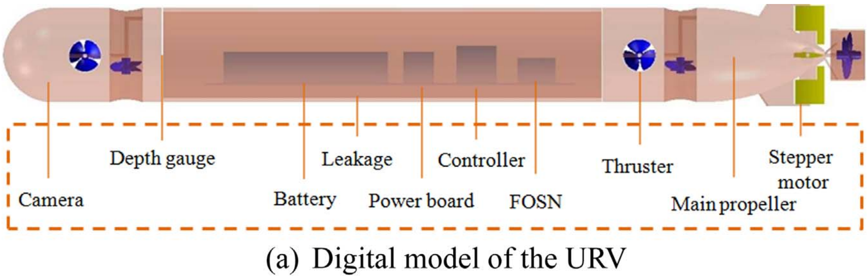
The rest of this paper is organized as follows: Section 2 briefly present the URV prototype. Section 3 illustrates the principle of risk analysis based on the MFNN method and the process of intelligent decision. Numerical simulation results are presented in Section 4. The structure of hardware in loop test platform, algorithm implementation on the PC/104, and the experimental results are given in Section 5. Conclusions and future work are summarized in Section 6.

2. Prototype overview

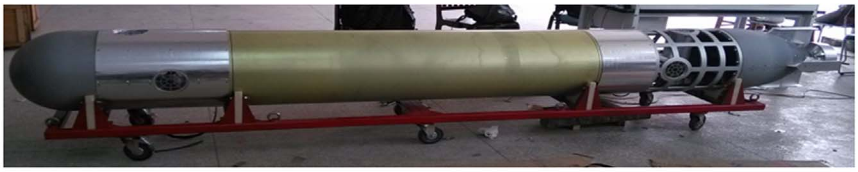
This section describes a torpedo-shaped URV prototype developed at the lab of autonomous robotic marine system (ARMS), which aims at performing environmental survey and surveillance task in medium range shallow water. In addition, the section depicts the hardware and software architecture of the URV, which inspires us to build fault tree analysis model of the whole system in the next section.

2.1. URV prototype

Fig. 1 presents the model of a torpedo-shaped URV, which is divided into the bow, middle and stern parts. The bow part is built as a frame structure, where a lateral thruster and a vertical thruster are accommodated for the horizontal and vertical motion, respectively. There is also a network camera for underwater observation and a ballast device for emergent recovery. The middle part of the URV is a cylindrical pressure hull containing a lithium-ion rechargeable battery, an embedded onboard controller, and some measurement and navigation devices. The onboard controller stack is composed of a PC/104 CPU board card, two multi-functional analog and digital I/O board cards, and two serial communication board cards. The measurement devices include a depth gauge and two water leakage probes. The navigation devices include a FOSN and a Mini-AHRS that measure the



(a) Digital model of the URV



(b) Prototype of the URV

Fig. 1. The URV developed at the ARMS lab. (a) Digital model of the URV, (b) Prototype of the URV.

position and attitude of the URV. The stern part is also built as a frame structure, which accommodates one thruster for the horizontal motion and another one for the vertical motion, a main propeller driven by the DC motor, and four rudders driven by stepper motors.

This URV measures approximately 5.34 m long and 0.62 m wide, and it weighs approximately 380 kg in air and −0.21 kg in water. In addition, it is designed to have a maximum cruising speed of 6 knots. The rest specifications are shown in Table 1.

2.2. Hardware architecture

The overall hardware architecture of the URV is illustrated in Fig. 2. The underwater onboard control system based on the embedded PC/104 controller communicates with the surface monitoring station via fiber, fiber transceiver and ethernet switch. A MSP-8 integrated board acquires data from the depth gauge, leakage sensor, Mini-AHRS and FOSN. The main propeller, four vertical/lateral thrusters and two pairs of rudders are driven by the MSP-4 board, ADT300 board and MSP-8 board via RS-485, DAC and RS232 interface, respectively. The management unit of the Lithium-ion battery sends the status information to the MSP-4 board via RS485. The power allocation board is controlled via a RS-232 serial port of the MSP-8 board to switch on/off the power supply of all the devices (Li et al., 2005). Once the fault treatment system detects a serious fault, the onboard controller will make critical decision to release the ballast via a DIO channel on the ADT652 board in order guarantee the safety of the URV.

2.3. Software architecture

The overall software architecture of the URV consists of the surface monitoring system and the underwater onboard system composed of five units, including network communication unit, motion control unit, sensor perception unit, actuator unit and fault treatment unit, as shown in Fig. 3. According to the operation mission assigned from the surface monitoring system and the real time position/orientation information collected by the sensor perception unit, the motion control unit runs online control algorithm and then sends control commands to the actuator unit in order to accomplish underwater operation mission. Meanwhile, the network communication unit transmits the real time posture and running status information of the URV from underwater onboard system to the surface monitoring system. In addition, the fault treatment unit will be alerted once a system fault occurs.

This paper focuses on working principle and algorithm of fault treatment unit including risk analysis subsystem and intelligent decision subsystem. In fact, this fault treatment unit works closely with other four onboard units as shown in Fig. 3, where the red line represents the fault flow and the black line denotes the information and control flow. In fact, the fault treatment system completes the acquisition and judgment of the following fault status in terms of multi-task operations.

(1) Judge whether or not the PC/104 can regularly receive the feedback information from the depth gauge, Mini-AHRS and FOSN, and then acquire and analyze the feedback information of the battery unit and leakage sensor;

(2) Judge whether or not the PC/104 can receive the acknowledgement information from four vertical/lateral thrusters and two pairs of rudders, and then acquire and analyze the feedback information of the main propeller;

(3) Detect and judge the communication status through the feedback information of *read* function of the onboard codes;

(4) Calculate the ratio of the running time to the set time, and the present depth to the predefined depth threshold.

After finishing fault detection of the URV onboard system, risk analysis can be carried out by adopting the MFNN method, and then the output information and critical decision result by applying the maximum membership and threshold principles will be sent to the motion control unit in order to take an emergent operation. For example, if the decision result indicates that a serious system fault has occurred, the ballast will be released by the motion control unit to drive the URV to surface up rapidly in order to guarantee its safety.

Table 1
Specification of the URV prototype.

Item	Specification
Depth rating	300 m

Propulsion	Thrusters: 4×Technodyne Model 280;
	Steering rudders: 4×HSM20147A;
	Main propeller: 1×840W

Sensor	INS: FOSN & Mini-AHRS; Depth: MB300; Leakage: ST003; Camera: OE14

Controller	PC/104(ISA); CF(4G); ADT300(D/A,DIO);
	ADT652(A/D);MSP-4 & 8(COM)

3. Risk analysis and critical decision

As shown in Fig. 4, the fault tree model of the underwater onboard system for the URV is firstly established based on the hierarchy of fault and its causality in order to complete the judgment of device-level fault.

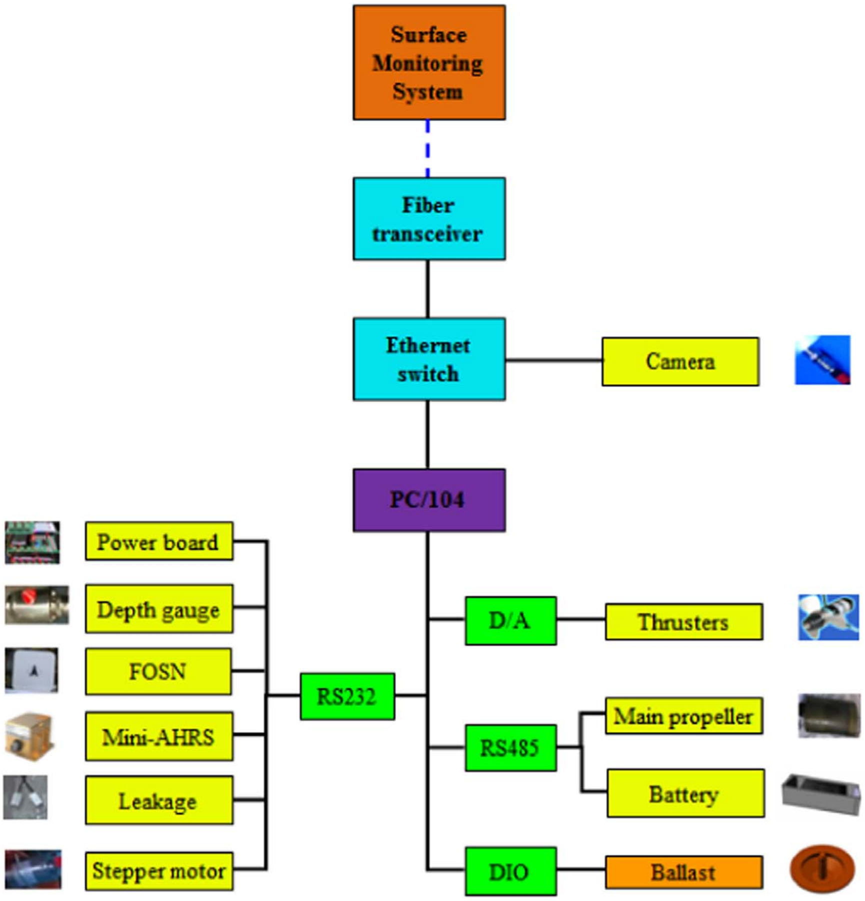


Fig. 2. Hardware architecture of the URV.

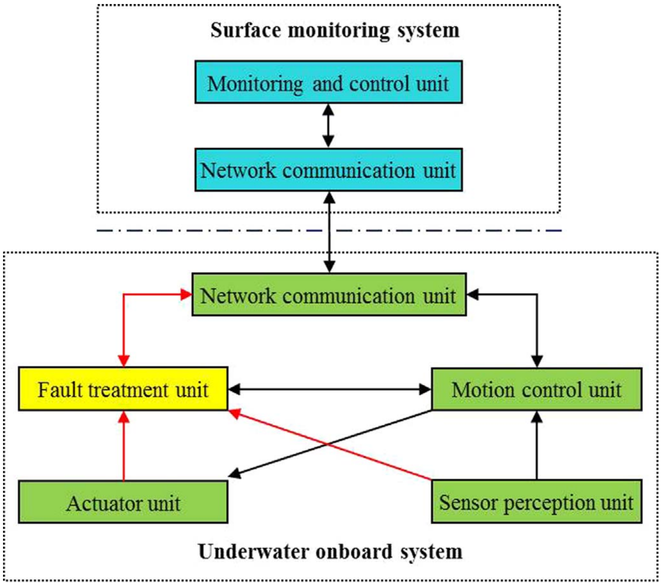

Surface monitoring system
Monitoring and control unit
Network communication unit
Network communication unit
Fault treatment unit
Motion control unit
Actuator unit
Sensor perception unit
Underwater onboard system

Fig. 3. Software architecture of the URV.

Subsequently, the real-time device-level fault information is sent to risk analysis subsystem based on the MFNN model in order to diagnose the system risk of the URV. Finally, the intelligent decision subsystem applies the maximum membership and threshold principles to evaluate the system risk, and then makes critical decision and takes the corresponding emergent operation to ensure the safety of the URV.

3.1. System fault tree model

Fault tree analysis (FTA) is a top-down deductive failure analysis in which an undesired state of a system is analyzed using Boolean logic to combine a set of lower-level events (Lee et al., 1985). This analysis method is mainly used in the fields of safety engineering and reliability engineering to understand how systems can fail, to identify the best ways to reduce risk or to determine event rates of a faulty accident or a particular system level failure. FTA often takes the most serious fault of the system as the target of fault analysis, which is called the top event, and then searches all the necessary or sufficient reasons resulting in the top event as the intermediate event, until the search reaches the individual fault sources, which are called basic events. This logic relationship is illustrated as a fault tree in Fig. 5, which consists of the top event, undeveloped event, intermediate events, basic events and the logic OR gates (Cheng et al., 2014).

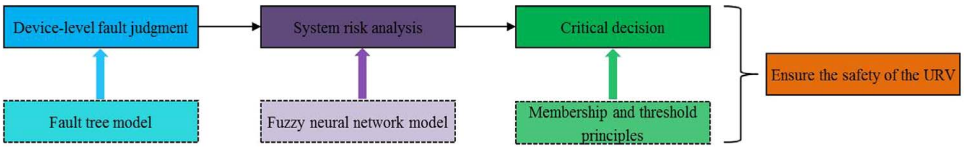

Device-level fault judgment
System risk analysis
Critical decision
Fault tree model
Fuzzy neural network model
Membership and threshold principles
Ensure the safety of the URV

Fig. 4. System architecture of enhanced onboard safety of the URV.

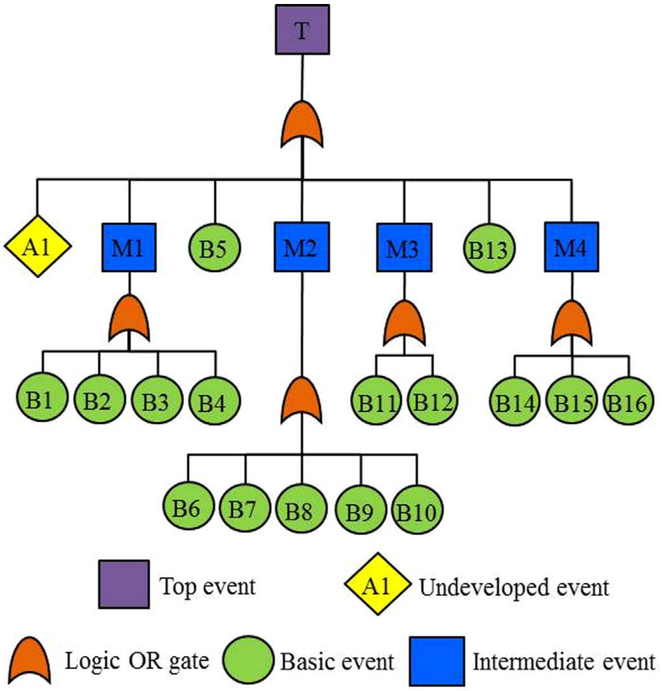


Fig. 5. Topology of URV fault tree.

Table 2
Events of the fault tree.

	Top event		Undeveloped event
T	Fault alarm	A1	Communication interruption
	Intermediate event		
M1	Sensor fault	M2	Power supply fault
M3	Leakage alarm	M4	Actuator fault
	Basic event		
B1	Depth gauge fault	B2	Mini-AHRS fault
B3	FOSN fault	B4	ST003 fault
B5	Over-depth alarm	B6	SOC alarm
B7	Undervoltage alarm	B8	Over temp. alarm
B9	Low temperature alarm	B10	Current alarm
B11	Bow leakage	B12	Stern leakage
B13	Timeout alarm	B14	Propeller fault
B15	Rudder fault	B16	Thruster fault

As listed in Table 2, the fault events include communication interruption between the surface monitoring system and the underwater onboard system, sensor fault, over-depth, power supply fault, water leakage, operation timeout, and actuator fault. In the fault tree model of the URV, the fault alarm is the top event and the communication interruption is the undeveloped event. Sensor fault, power supply fault, water leakage and actuator fault are the intermediate events. Depth gauge fault and other fifteen faults are the basic events. All the basic events constitute the intermediate events via logic OR gates, and then all the intermediate events render the top event by logic OR gates.

In the fault treatment system, the principle of judging operation timeout and over-depth is to calculate the ratio of the present value to the set threshold, and the power supply fault and water leakage are detected from the real-time feedback information. In addition, the rest faults are identified based on whether or not the PC/104 can receive the updated data from external devices or analyzed through the communication data packet.

3.2. Risk analysis based on MFNN

This section firstly explains why to choose the Mamdani fuzzy neural network (MFNN) method for risk analysis subsystem, and then describes the implementation steps of the MFNN method. Finally, the critical decision process is presented in details.

3.2.1. Why to choose MFNN

In this paper, for all kinds of faults, the fuzzy partition of the membership function is set to 3, represented by positive small (PS), positive middle (PM) and positive big (PB), respectively. The fault status corresponding to different fuzzy ratings and output range of the original detection information are shown in Table 3. Obviously, it needs to define up to 3^7 rules in the fuzzy expert system without any simplification. In this sense, programming implementation is arduous since it will take a long time to look up the table, theorize and make decisions, which further results in the poor real-time performance. In addition, it is difficult to reduce the number of rules within an acceptable range. Hence, the direct fuzzy logic method is not suitable for the onboard fault inference for the URV.

Nevertheless, the fuzzy logic method is a powerful tool to deal with uncertainty and nonlinear problems (Ramot et al., 2003; Wang and Er, 2016; Xiang et al., 2017). For instance, the fuzzy controller is adopted to complete the diagnostics of an electrical power system for a boiling water reactor nuclear power plant (Gmytrasiewicz et al., 1990). On the other hand, the neural network method has adaptive learning capacity and is widely used in fault diagnosis for thrusters (Yeh et al., 2011; Chu et al., 2016; Peng et al., 2017. A combined artificial neural network and fault tree analysis model is used to avoid the coal and gas outburst fault (Ruilin and Lowndes, 2010). Motivated by those considerations, the paper builds the fault tree model and then proposes the MFNN method integrating the advantages of both fuzzy logic and neural network to complete the system risk analysis of the URV. The parameters of risk analysis subsystem are set by adopting neural network to offline learn a small amount of samples with rational distribution in MATLAB, and then the online fuzzy inference based on the Mamdani model on the PC/104 can be completed, since the MFNN method is able to solve the problems of complicated enumeration and demanded real-time response. Therefore, the MFNN method is qualified for onboard risk analysis of the URV with uncertainty and complexity.

3.2.2. Architecture of MFNN

By referring to Figueroa-Garcia et al. (2015), a five-layer fuzzy neural network with multi-input and multi-output is proposed, as shown in Fig. 6, where the circle represents the node of the network structure, and the arrow represents the relationship between the nodes of different layers. The five layers are described as follows:

(1) Layer 1 - Normalization: The node of this layer is directly connected to each component of the fault input vector and represents the fault status. All input data $x_1, x_2, \ldots, x_n$ constituting a vector $x = [x_1, x_2, \ldots, x_n]^T$ should be normalized to the interval [0, 1], where n represents the dimension of input data. As illustrated in Table 3 and Fig. 6, this layer has seven nodes, namely $N_1 = n = 7$.

(2) Layer 2- Fuzzification: This layer is used to calculate the membership function $\mu_i^{j_i}(=\mu_{A_i}^{j_i}(x_i))$ of each component of the input data in a fuzzy set A_i, where $i = 1, 2, \ldots, N_1$, and $j_i = 1, 2, \ldots, m_i$. As listed in Table 3, m_i denotes the fuzzy partition of the ith input is equal to 3. As each node of this layer represents a linguistic variable, its total number is 3×7, namely $N_2 = \Sigma_{i=1}^{N_1} m_i = 21$. As shown in Fig. 7, the following Gaussian membership function is chosen in this layer

$$\mu_i^{j_i} = e^{-\frac{(x_i - c_{ij_i})^2}{\sigma_{ij_i}^2}} \tag{1}$$

where c_{ij_i} and σ_{ij_i} represent the center and the width of membership function, respectively.

Table 3
Fault set and Fuzzy ratings.

No.	Fault type	Fuzzy rating	Detection range
x_1	Communication interruption	PS-The communication function is normal	0.0 ~ 0.25
		PM- There is occasional interruption in the process of communication	0.25 ~ 0.75
		PB- The communication is completely interrupted	0.75 ~ 1.0
x_2	Sensor fault	PS - The PC/104 can receive updated data from AHRS, FOSN and MB300	0.0 ~ 0.25
		PM -The PC/104 cannot receive updated data from AHRS or FOSN	0.25 ~ 0.75
		PB -The PC/104 cannot receive updated data from MB300	0.75 ~ 1.0
x_3	Over-depth	PS-No over-depth	20 ~ 200
		PM-The depth of the URV is close to the bounding depth	200 ~ 300
		PB-The depth of the URV is larger than the bounding depth	300 ~ 310
x_4	Power supply fault	PS - No alarm	0.0 ~ 0.25
		PM - The undervoltage alarm or/and low temperature alarm	0.25 ~ 0.75
		PB - The overvoltage alarm or/and high temperature alarm	0.75 ~ 1.0
x_5	Water leakage	PS -No leakage	0.0 ~ 0.25
		PM - Bow leakage or stern leakage	0.25 ~ 0.75
		PB - Serious bow leakage and stern leakage	0.75 ~ 1.0
x_6	Timeout	PS-No timeout	0.0 ~ 6.0
		PM-The total running time of the onboard system is close to the set time	6.0 ~ 8.0
		PB-The total running time of the onboard system is larger than the set time	8.0 ~ 10.0
x_7	Actuator fault	PS - No failure of all the actuators	0.0 ~ 0.25
		PM - Failure of the main propeller, rudders or horizontal thrusters	0.25 ~ 0.75
		PB - Failure of two vertical thrusters	0.75 ~ 1.0

(3) Layer 3- Intersection: Each node of this layer is a rule of the fuzzy logic system, and each one uses a *t*-norm to compose the intersection operation. The output of the *j*th node is defined as its activation level, namely α_j, which can be represented as

$$\alpha_j = \mu_1^{j_1}\mu_2^{j_2}\cdots\mu_n^{j_n} \tag{2}$$

where

$$j_i \in \{1, 2, \ldots, m_i\},\ i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, m,\ m = \prod_{i=1}^{N_1} m_i$$

Therefore, there are 3^7 nodes in this layer, namely $N_3 = 3^7$. As shown in Fig. 7, only the linguistic variable near the center of membership function has large degree of membership, while the degree of membership of the linguistic variable away from the center is quite small for a given fault state input. Hence, most of α_j is close to zero except for a few node output.

(4) Layer 4 - Aggregation: The number of nodes in this layer is the same as that of layer 3, namely $N_4 = m = 3^7$. This aggregation process is to normalize each α_j by using the sum of all activation levels coming from layer 3 as follows:

$$\bar{\alpha}_j = \alpha_j / \sum_{j=1}^{m} \alpha_j \tag{3}$$

(5) Layer 5 - Defuzzification: This layer calculates the defuzzification output y of the MFNN model by using all aggregated values $\bar{\alpha}$ and the connection weights ω between layer 4 and layer 5 as follows:

$$y = \omega\bar{\alpha} \tag{4}$$

where

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_r \end{bmatrix},\ \omega = \begin{bmatrix} \omega_{11} & \omega_{12} & \cdots & \omega_{1\,m} \\ \omega_{21} & \omega_{22} & \cdots & \omega_{2\,m} \\ \vdots & \vdots & \vdots & \vdots \\ \omega_{r1} & \omega_{r2} & \cdots & \omega_{rm} \end{bmatrix},\ \bar{\alpha} = \begin{bmatrix} \bar{\alpha}_1 \\ \bar{\alpha}_2 \\ \vdots \\ \bar{\alpha}_r \end{bmatrix} \tag{5}$$

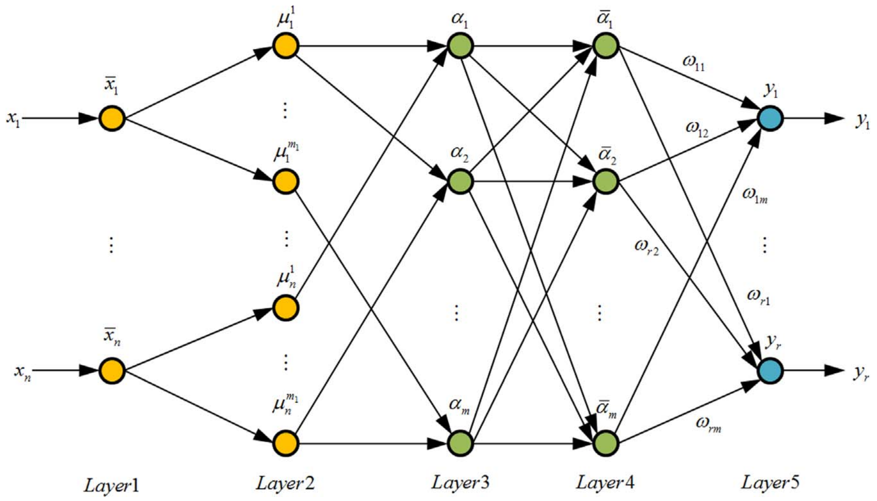


Fig. 6. The architecture of MFNN.

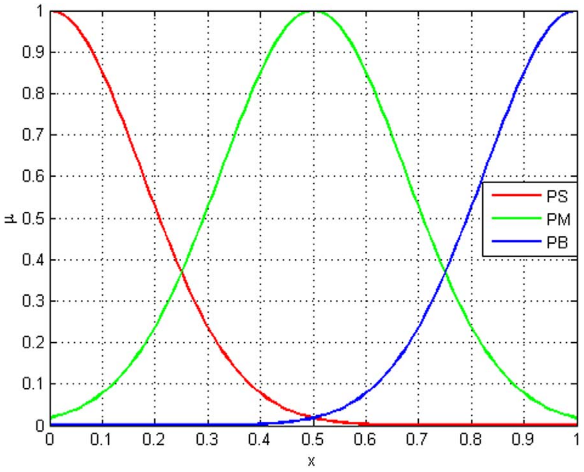


Fig. 7. The membership function of input variables.

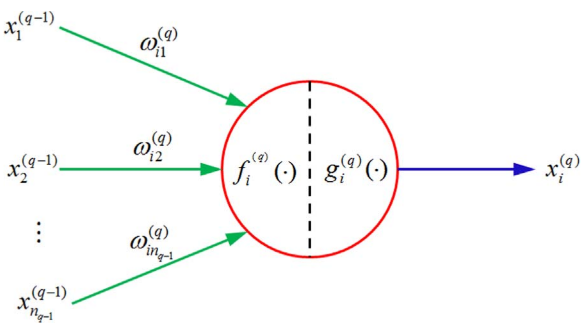


Fig. 8. The basic structure of single node.

Table 4
Critical decision and emergency operations.

No.	Fault state	Recovery operation
I	Normal state (ZO)	Continue to work
II	Minor fault (PZ)	Continue to work and wait for the emergent coperation
III	Small fault (PS)	Turn off the power supply & let the URV surface up by its own buoyancy
IV	Middle fault (PM)	Stop the propeller & turn on the vertical thrusters to drive the URV to surface up
V	Large fault (PB)	Turn off the power supply & release the ballast to drive the URV to surface up

In fact, the defuzzification output y implies the risk degree of system fault, which can be used to evaluate the system risk and render critical decision by taking the following learning algorithm.

3.2.3. Learning algorithm

The MFNN model in this paper only needs to adjust the connection weights ω between the layer 4 and the layer 5, since the fuzzy partition of each component of input variables and the center and width of the membership function in layer 2 have been determined. Although there is a large amount of learning strategies which can be applied to the learning algorithm of connection weights, the back propagation algorithm is chosen as the MFNN model essentially is a multi-layer feed forward network.

In order to derive the error back propagation algorithm, the relationship between the input s_i and the output x_i of single node should be described. Fig. 8 shows the basic structure of the ith node in layer q.

Table 5
Learning samples.

No.	Input: x	Output: y
1	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]	[0.0, 0.0, 0.0, 0.0, 0.0]
2	[0.2, 0.2, 160, 0.2, 0.2, 4.8, 0.2]	[0.7, 0.5, 0.0, 0.0, 0.0]
3	[0.0, 0.0, 0.0, 0.0, 0.0, 8.4, 0.0]	[0.0, 0.0, 1.0, 0.0, 0.0]
4	[0.2, 0.2, 160, 0.2, 0.2, 8.4, 0.2]	[0.0, 0.9, 0.3, 0.0, 0.0]
5	[0.0, 0.5, 0.0, 0.0, 0.0, 0.0, 0.0]	[0.0, 0.0, 1.0, 0.0, 0.0]
6	[0.0, 0.0, 0.0, 0.5, 0.0, 0.0, 0.0]	[0.0, 0.0, 1.0, 0.0, 0.0]
7	[0.0, 0.0, 0.0, 0.0, 0.0, 7.0, 0.5]	[0.0, 0.0, 0.0, 1.0, 0.0]
8	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]	[0.0, 0.0, 0.0, 0.0, 0.0]
9	[0.8, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]	[0.0, 0.0, 0.0, 0.0, 1.0]
10	[0.0, 0.8, 0.0, 0.0, 0.0, 0.0, 0.0]	[0.0, 0.0, 0.0, 0.0, 1.0]
11	[0.0, 0.0, 300, 0.0, 0.0, 0.0, 0.0]	[0.0, 0.0, 0.0, 0.0, 1.0]
12	[0.0, 0.0, 0.0, 0.8, 0.0, 0.0, 0.0]	[0.0, 0.0, 0.0, 0.0, 1.0]
13	[0.0, 0.0, 0.0, 0.0, 0.8, 0.0, 0.0]	[0.0, 0.0, 0.0, 0.0, 1.0]
14	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.8]	[0.0, 0.0, 0.0, 0.0, 1.0]

The input $s_i^{(q)}$ and the output $x_i^{(q)}$ of this node are defined as:

$$s_i^{(q)} = f_i^{(q)}(x_1^{(q-1)}, x_2^{(q-1)}, \dots, x_{n(q-1)}^{(q-1)}; \omega_{i1}^{(q)}, \omega_{(i1)}^{(q)}, \dots, \omega_{in_{q-1}}^{(q)}) \tag{6}$$

$$x_i^{(q)} = g_i^{(q)}(s_i^{(q)}) \tag{7}$$

For the layer 5, the input $s_i^{(5)}$ and output $x_i^{(5)}$ are given as follows:

$$s_i^{(5)} = f_i^{(5)}(\cdot) = \sum_{j=1}^{m} \omega_{(ij)} x_j^{(4)} = \sum_{j=1}^{m} \omega_{(ij)} \bar{\alpha}_j \tag{8}$$

$$x_i^{(5)} = g_i^{(5)}(s_j^{(5)}) = s_i^{(5)} = y_i \tag{9}$$

Taking into account r nodes in the layer 5, the total squared error can be calculated as:

$$E = \frac{1}{2}\sum_{i=1}^{r}(y_{di} - y_i)^2 \tag{10}$$

where y_{di} is the desired output and y_i is the actual output. Another popular measure method is the root mean squared error (RMSE), defined as:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{r}(y_{di} - y_i)^2}{r}} \tag{11}$$

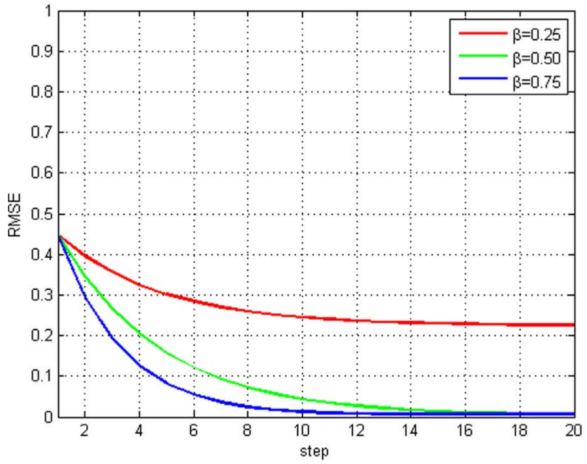
The first step of this error back propagation algorithm is to calculate the intermediate variable $\delta_i^{(5)}$ denoted as:

$$\delta_i^{(5)} \triangleq -\frac{\partial E}{\partial s_i^{(5)}} = -\frac{\partial E}{\partial y_i} = y_{di} - y_i \tag{12}$$

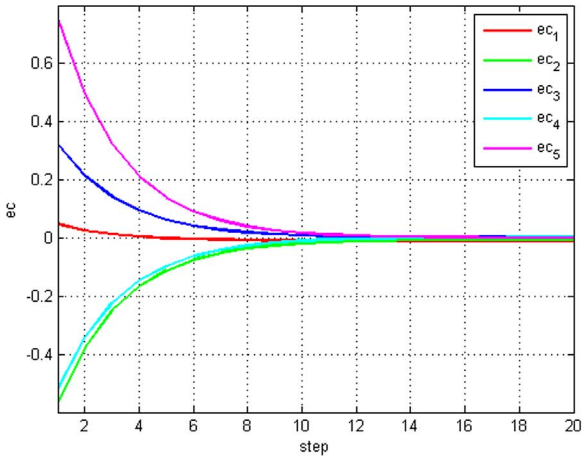
Second, the first-order gradient algorithm of the connection weights ω is derived as:

$$\frac{\partial E}{\partial \omega_{ij}} = \frac{\partial E}{\partial s_i^{(5)}}\frac{\partial s_i^{(5)}}{\omega_{ij}} = -\delta_i^{(5)} x_j^{(4)} = -(y_{di} - y_i)\bar{\alpha}_j \tag{13}$$

Finally, the gradient descend algorithm for updating the connection weights ω is designed as:



(a) RMSE of different learning rates



(b) Error $ec = y_{di} - y_i, \beta = 0.75$

Fig. 9. Learning error of fuzzy neural network. (a) RMSE of different learning rates, (b) Error $ec = y_{di} - y_i$, $\beta = 0.75$.

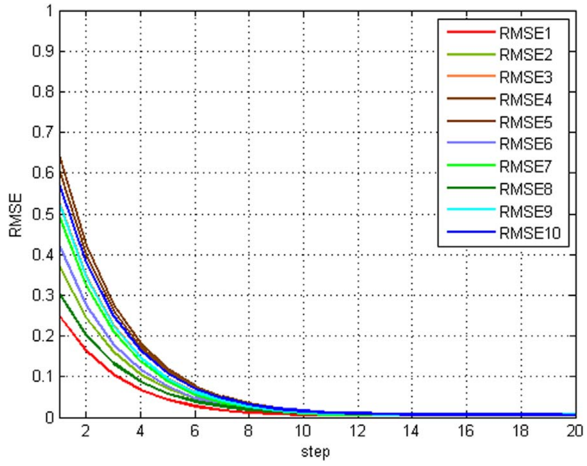


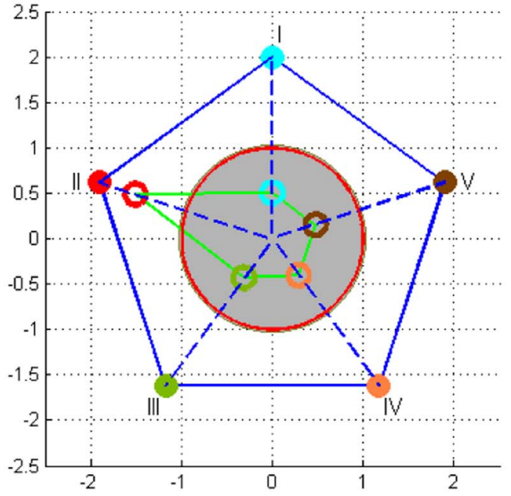
Fig. 10. RMSE of different initial values.

$$\omega_{ij}(k+1) = \omega_{ij}(k) - \beta\frac{\partial E}{\partial \omega_{ij}} \tag{14}$$

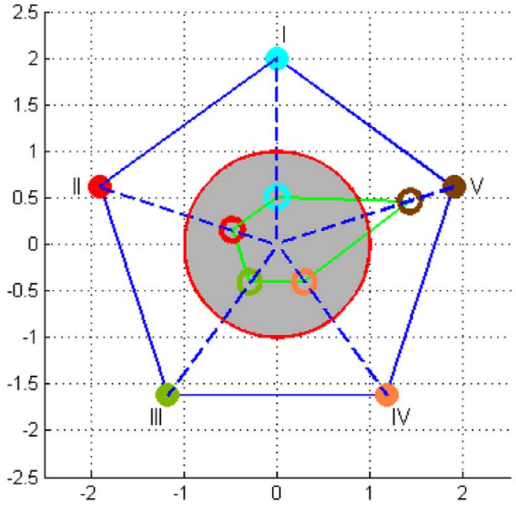
where $\beta \in [0, 1]$ is the learning rate. In general, smaller value of β might result in larger computing efforts, and larger value of β might result in faster convergence of the learning algorithm with higher deviations.

Table 6
Comparison of desired output and actual output.

No.	Output	Value
Sample 3	Desired output	[0.0, 1.0, 0.0, 0.0, 0.0]
	Actual output	[0.013, 1.092, 0.035, 0.004, 0.003]
	Shifted output	[0.513, 1.592, 0.535, 0.504, 0.503]
Sample14	Desired output	[0.0, 0.0, 0.0, 0.0, 1.0]
	Actual output	[0.007, 0.004, −0.001, −0.001, 0.999]
	Shifted output	[0.507, 0.504, 0.499, 0.499, 1.499]



(a) Result of sample 3



(b) Result of sample 14

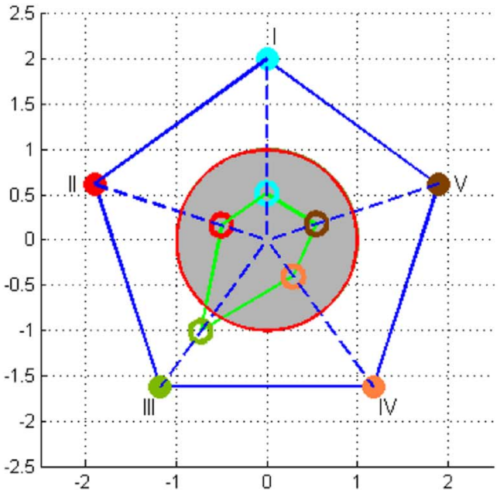
Fig. 11. The decision output of two different test samples. (a) Result of sample 3, (b) Result of sample 14.

3.3. Critical decision

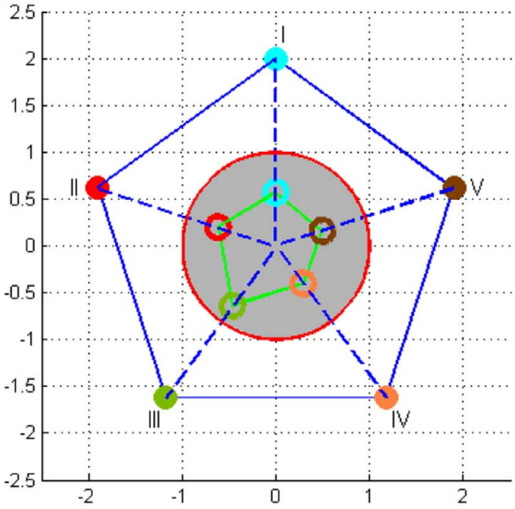
This section shows the emergent operations built by expert knowledge rules, and then describes the decision-making process based on maximum membership and threshold principles.

3.3.1. Emergency operations

According to the risk degree of system fault, the critical decision and corresponding emergent operations are drawn up in Table 4. The decisions and operations are dependent on expert knowledge rules. For instance, if the whole system is normal, the URV will continue to work



(a) Result of a random fault



(b) Result of another random fault

Fig. 12. The decision output of two random system-level faults. (a) Result of a random fault, (b) Result of another random fault.

according to the expert knowledge rules. If a serious system fault occurs, the onboard system will turn off the power supply and release the ballast to drive the URV to surface up rapidly. Decisions III, IV, and V are intelligent because the onboard fault treatment system can autonomously make rapid decision based on MFNN method to ensure the safety of the URV.

3.3.2. *How to make critical decision*

There are two principles for the fault treatment system to make critical decisions:

(1) Maximum membership principle: According to Table 4, the dimension of the output vector of risk analysis subsystem should be 5, namely *r*=5. As Gaussian membership functions are chosen for all the parameters of the output vector, and y_i represents the degree of membership of decision *i*. If $\max\{y_1, y_2, y_3, y_4, y_5\}$ is equal to y_i based on the maximum membership principle, the system-level fault is most likely to belong to the fault classification corresponding to decision *i*. Note that the maximum membership principle only considers the relative size of degree of membership, but ignores the absolute size. Hence, the summarized information of this principle is quite few despite of the simplicity, which results in the one-sidedness characteristics of making decision.

(2) Threshold principle: Taking into account the one-sidedness of making decision dependent on maximum membership principle, a threshold level $\lambda \in [0, 1]$ is introduced in the intelligent decision subsystem. If $y_i \geq \lambda$, the intelligent decision subsystem will make the decision *i*; If $y_i < \lambda$ and the URV is far away from the water surface, the fault decision subsystem will make the highest decision *V* to ensure the safety of the URV because a reliable decision is not made based on above two principles; Otherwise it will make a higher decision $i + 1$ to ensure no risk at all by considering the fact that the URV is near the water surface and relatively safe.

4. Numerical simulation

In order to validate the feasibility of the fault treatment system, the numerical simulation in MATLAB is firstly carried out. Through learning sample data, a suitable learning rate of the neural network is chosen. Subsequently, the fault treatment system including risk analysis subsystem and intelligent decision subsystem is tested by two samples and further validated by two random faults.

4.1. *Sample learning*

As listed in Table 5, 14 data set covering all the seven types of fault sources and five kinds of decisions are chosen as the learning sample. Note that for an URV with different onboard sensors or actuation configurations (i.e., fully-actuated versus under-actuated configurations Xiang et al., 2016c), the choices of the number and distribution for fault samples in this paper are also suitable for them except adjusting a small quantity of specificate data. Firstly, the MFNN model accumulates the learned knowledge into the connection weights ω with massive parameters by learning those samples. Subsequently, for any fault input, this risk analysis subsystem is able to adopt the MFNN method to complete the diagnosis of system-level fault.

When the initial value of the connection weights ω is set as 0 and the learning rate is chosen as 0.25, 0.50 and 0.75, respectively, the RMSE of neural network training is shown in Fig. 9(a), where the RMSE converges to 0.225, 0.007 and 0.006, respectively. The number of learning steps is 18, 16 and 12, respectively. In addition, the individual error of five output components converges to zero rapidly when the learning rate β is equal to 0.75, as shown in Fig. 9(b).

When the learning rate β is fixed as 0.75 and the initial value of the connection weights is generated by a random function, the RMSE of ten different initial values is depicted in Fig. 10. It is concluded that the RMSE under different initial states converges to the same error 0.006 when the number of learning steps is 12, which suggests that the structure of the MFNN model is feasible and the learning rate $\beta = 0.75$ is a suitable choice.

4.2. *Sample test*

In order to further validate the feasibility of the MFNN model, two different sample data are chosen in the sample test, i.e., sample 3 and sample 14 in Table 5, respectively. The input of sample 3 is $[0.0, 0.0, 0.0, 0.0, 0.0, 8.4, 0.0]$, which represents the URV has been working for 8.4 h and all the status is normal. According to Table 5, the desired output should be $[0.0, 1.0, 0.0, 0.0, 0.0]$ that illustrates a minor system fault occurred. The input of sample 14 is $[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.8]$, which represents the URV has a fault in a pair of vertical thrusters and the other status is normal according to Table 3. The desired output should be $[0.0, 0.0, 0.0, 0.0, 1.0]$ that illustrates a very serious system fault occurred. As listed in Table 6, the difference between the desired output and the actual one is small, which indicates that the proposed MFNN model completely meets the requirement of system risk analysis for the URV.

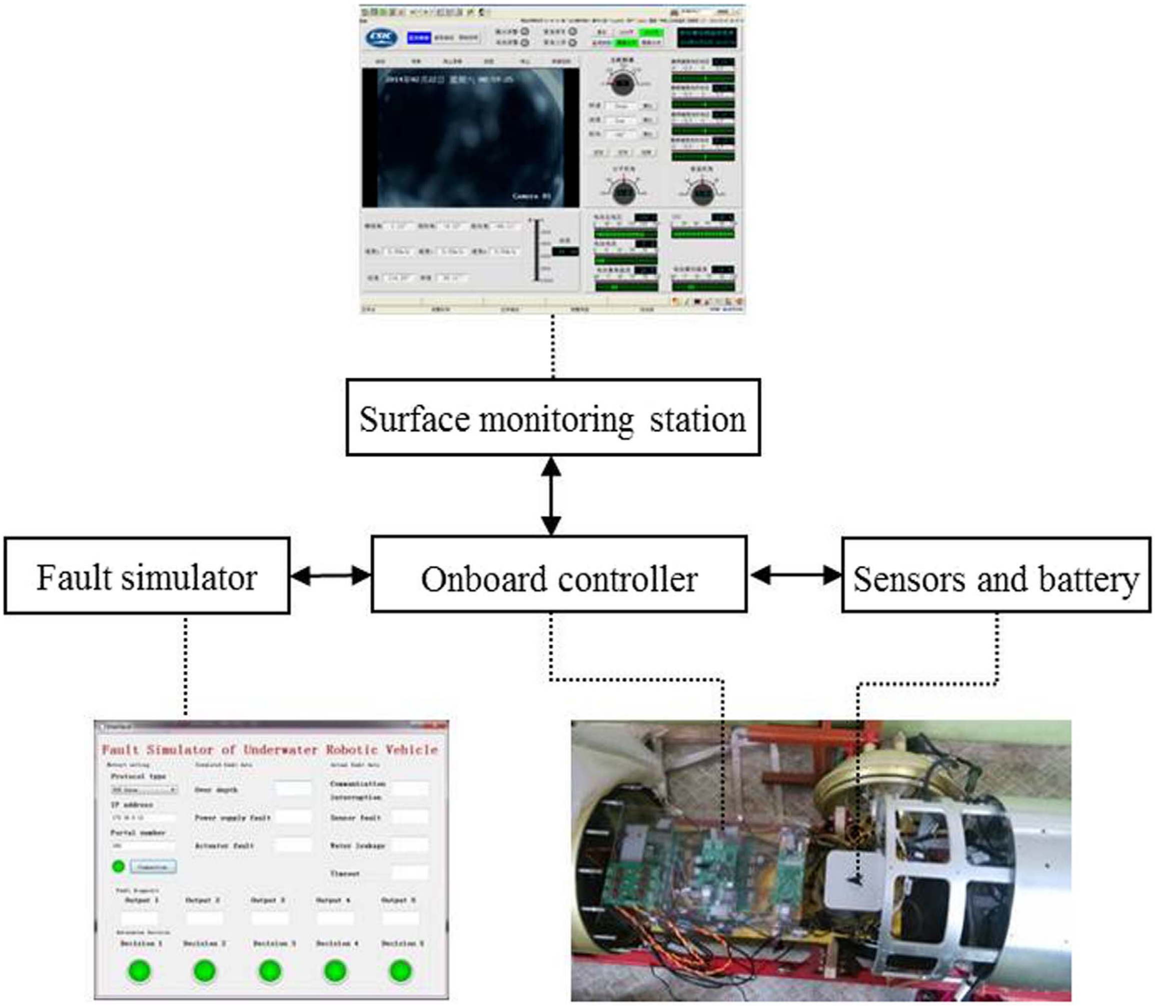


Fig. 13. Hardware in loop test platform.

Table 7
Fuzzification.

```
Step 1. Fuzzification in layer1 and layer2
Input: $x = [x_1, x_2, x_3, x_4, x_5, x_6, x_7]$, $c(7 \times 3)$, $\sigma(7)$
Normalize: $x = [\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4, \bar{x}_5, \bar{x}_6, \bar{x}_7]$
for $i$=1 to 7 do
    for $j_i$=1 to 3 do
        $\mu_i^{j_i} = e^{-\frac{(\bar{x}_i - c_{ij_i})^2}{\sigma_{ij_i}^2}}$
    end for
end for
Output: $\mu(7 \times 3)$
```

As shown in Fig. 11, a radar chart is used to reveal the intelligent decision result rendered by the maximum membership principle and threshold principle. Five vertices of the radar chart denote the decision I, II, III, IV and V, respectively, and the absolute length between the center and each vertex is 2, representing the interval $[-0.5, 1.5]$. Due to the existence of smaller negative numbers near zero among the output vector of the risk analysis subsystem, all the components of the output vector are shifted with a fixed compensation 0.5. The shifted results listed in Table 6 are just the length from the colorful hollow circle to the center of the radar chart, and the shifted threshold level λ' represented by a red circle is set to 1, as shown in Fig. 11. In this sense, we can visually verify whether or not the intelligent decision result meets the threshold principle. If it falls into the shaded area of the red circle, it will not meet the threshold principle and requests further decision.

Table 8
Fuzzy inference.

```
Step2. Fuzzy inference in layer3 and layer4
Input: $\mu(7 \times 3)$
for $j_7 = 1$ to 3 do
    ...
        for $j_1 = 1$ to 3 do
            $\alpha_j = \mu_1^{j_1} \mu_2^{j_2} \ldots \mu_7^{j_7}$
            $j = j + 1$
        end for
    ...
end for
Normalize: $\bar{\alpha}_j = \alpha_j / \sum_{j=1}^{m} \alpha_j, j = 1, 2, \ldots, 3^7$
Output: $\bar{\alpha}(3^7 \times 1)$
```

4.3. Random fault test

Although Fig. 11(a) and 11(b) show the test results of sample 3 and 14, Fig. 12 presents the further test results of two random fault cases that are not included in fourteen samples in Table 5. The input of the fault in Fig. 12(a) is taken as [0.0, 0.1, 8, 0.6, 0.0, 2.4, 0.1], which represents that the URV has been working for 2.4 h, the present depth is 8 m, the battery has undervoltage or low temperature, and the other status is normal. The shifted output vector of the risk analysis subsystem is [0.5148, 0.5285, 1.2471, 0.4994, 0.5771] and the corresponding decision is III owing to:

Table 9
Defuzzification and fault decision.

```
Step 3. Defuzzification in layer5 and fault decision
Input: $\omega(5^7)$, $\overline{\alpha}(3^7 \times 1)$, $\lambda$
$y = \omega\overline{\alpha}$
$y_{di} = \max\{y_1, y_2, y_3, y_4, y_5\}$
if $y_{di} \leq \lambda$
    make decision $i$
else
    if $h \leq 1$
        make decision $i + 1$
    else
        make decision $V$
    end if
end if
Output: decision value
```

$$\max(y_i) = y_3 = 1.2471 > 1 \tag{15}$$

Hence, the URV has a small fault, and the onboard controller will turn off the power supply and drive the URV to surface up by its own buoyancy.

In addition, the input vector of the fault in Fig. 12(b) is taken as [0.0, 0.1, 0.8, 0.6, 0.0, 6.2, 0.1], which represents that the URV has been working for 6.2 h, the present depth h is 0.8 m, the battery has undervoltage or low temperature, and the other status is normal. The shifted output vector y of fault diagnosis system is $[0.5697, 0.6486, 0.7891, 0.4947, 0.5197]$. Because $\max(y_i) = y_3 = 0.7891 < 1$ and h=0.8 m, the intelligent decision subsystem will make decision IV and the onboard controller will command the vertical thrusters to drive the URV surfacing up.

5. Experimental verification

In order to validate the performance of the fault treatment system for the URV, the hardware in loop test is carried out at the ARMS lab, which mainly focuses on the algorithm and function tests on the onboard PC/104 controller. This section describes the structure diagram of hardware in loop test platform at first, and then illustrates the implementation steps of the fault treatment system on the PC/104. Finally, the experimental results are presented.

5.1. Hardware in loop test platform

The hardware in loop simulation platform of fault treatment system for the URV consists of the surface monitoring station, onboard PC/104 controller, sensors and battery carried by the URV prototype, and fault simulator, as shown in Fig. 13, where the functions of four parts are described as follows:

1) Surface monitoring station: As the host of onboard PC/104 controller, it plays a role in developing, downloading, and debugging the program code, and can also show the data collected from the power supply, sensors and so on;

2) Fault simulator: It displays the real-time fault status of communication interruption, sensor failure, timeout and water leakage, and presents the output vector of risk analysis subsystem and the result of intelligent decision subsystem. In addition, it can simulate fault data of over-depth, power supply alarm, and actuator fault that might occur when the URV navigates in the water;

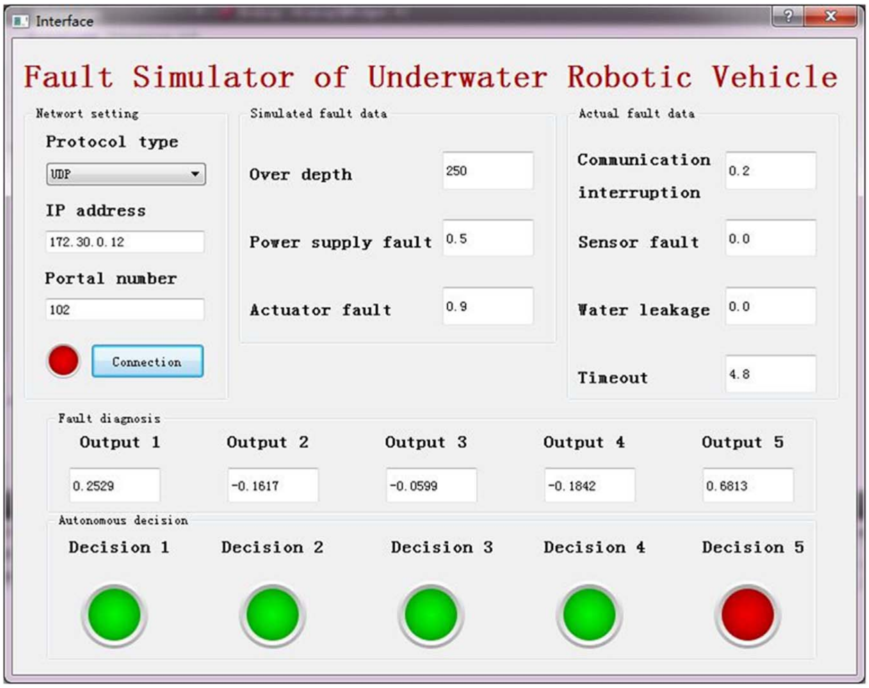
3) Onboard controller: It is a PC/104 stack, where system risk analysis based on the MFNN method and intelligent decision by applying the maximum membership and threshold principles are programmed in terms of multi-task and multi-thread operations;

4) Sensors and battery: The sensors mainly detect the real-time position and attitude of the URV, and the battery provides the power for the PC/104 and sensors.

5.2. Algorithm implementation

Similarly to numerical simulation, the algorithm implementation on the PC/104 platform running in the VxWorks RTOS goes through four steps, including fuzzification, fuzzy neural inference, defuzzification and intelligent decision by coding in C language instead of MATLAB language.

The first step is fuzzification of the input fault data including the layer 1 and the layer 2 in Fig. 6. First, the original fault data collected by the PC104 is normalized to the interval $[0, 1]$, and then it is fuzzified by Gaussian membership function. As illustrated in Table 7, $\overline{x}$ is the normalized vector of the original fault vector x, and the output μ is a



Fault Simulator of Underwater Robotic Vehicle

Networt setting
Protocol type: UDP
IP address: 172. 30. 0. 12
Portal number: 102

Simulated fault data
Over depth: 250
Power supply fault: 0. 5
Actuator fault: 0. 9

Actual fault data
Communication interruption: 0. 2
Sensor fault: 0. 0
Water leakage: 0. 0
Timeout: 4. 8

Fault diagnosis

Output 1	Output 2	Output 3	Output 4	Output 5
0. 2529	-0. 1617	-0. 0599	-0. 1842	0. 6813

Autonomous decision

Decision 1	Decision 2	Decision 3	Decision 4	Decision 5
🟢	🟢	🟢	🟢	🔴

Fig. 14. Fault simulator interface.

fuzzy set that represents the degree of membership, expressed by a matrix with seven rows and three columns.

The second step is fuzzy neural inference including the layer 3 and the layer 4 in Fig. 6. As described in Table 8, the key is that the PC/104 calculates the activation level α through the intersection operation based on the fuzzy set in the layer 2. The procedure in Table 8 denotes the assignment made to j_6, j_5, j_4, j_3, j_2 successively by using *for* function. Finally, α is normalized to a vector $\overline{\alpha}$ with 3^7 elements.

The third step is defuzzification including the layer 5 of Fig. 5 to render critical decision. Firstly, the output vector of risk analysis subsystem is calculated, which depends on the normalized activation level $\overline{\alpha}$ and the connection weights ω determined by offline sample learning via the MFNN method. Subsequently, a critical decision is made by intelligent decision subsystem based on the maximum membership and threshold principles. Finally, the corresponding emergency operation is taken to ensure the safety of the URV. In Table 9, λ and h represent the threshold level and the diving depth of the URV, respectively.

5.3. Experimental test

Fault simulator can simulate the status of over-depth x_3, power supply fault x_4 and actuator fault x_7, and the other four device-level faults are set as follows:

(1) Communication interruption x_1: Pull out the fiber from the ethernet switch.

(2) Sensor fault x_2: Turn off the power supply for the depth gauge, FOSN or Mini-AHRS.

(3) Water leakage x_5: Put the detection probe of the leakage sensor in the water.

(4) Timeout x_6: Calculate the ratio of the total running time of the onboard system to the predefined time threshold.

In this paper, one of the fault treatment results is given in Fig. 14. When the input of the fault status is taken as $[0.2, 0.0, 250, 0.5, 0.0, 4.8, 0.9]$, which represents that the communication function is normal, the PC/104 can receive the updated data from all the sensors, the present depth is 250 m, the power supply has undervoltage or low temperature, there is no water leakage, the URV has been working for 4.8 h, and a fault occurs in the vertical thrusters. By adopting the proposed MFNN method, the output vector of risk analysis subsystem is $[0.2529, -0.1617, -0.0599, -0.1842, 0.6813]$ corresponding to decision V. It means that the URV has a serious fault, and the onboard controller will turn off the power supply and release the ballast to drive the URV surfacing up rapidly. The hardware in loop tests run more than 140 times, and the output results of the risk analysis and intelligent decision-making conforms to reasonable safety operations, which indicates that the proposed fault treatment system of the URV has satisfactory performance.

6. Conclusions

A dedicated two-layered fault treatment system including risk analysis subsystem and intelligent decision subsystem is proposed to enhance onboard safety of the URV in this paper. According to the fault tree model, the device-level fault data is derived from the system status and the feedback information between the onboard controller and all the external devices. Subsequently, in the risk analysis subsystem, the device-level fault data goes through fuzzification, fuzzy neural inference and defuzzification steps in order to diagnose the risk degree of system-level fault. In the intelligent decision subsystem, the maximum membership and threshold principles are adopted to evaluate the system risk of the URV, and then the critical decision and emergent operation based on expert knowledge are rendered in order to ensure the safety of the URV.

At present, we have completed the design and implementation of the fault treatment system and tested many times through numerical simulations and hardware in loop tests. Experimental results demonstrate that the proposed fault treatment system for the URV has satisfactory feasibility and efficiency. In the future, we will test the fault treatment system in the tank or open waters.

Acknowledgements

This work was partially supported by National Natural Science Foundation of China (NNSF) (under Grant 51579111 and Grant 51209100), the Fundamental Research Funds for the Central Universities (under Grant 2017KFYXJJ005), and the State Key Lab Research Fund of Ocean Engineering (Grant 201504).

References

Akmal, M., Yusoff, M., Arshad, M., 2012. Active fault tolerant control of a remotely operated vehicle propulsion system. Procedia Eng. 41, 622–628.
Alessandri, A., Caccia, M., Veruggio, G., 1999. Fault detection of actuator faults in unmanned underwater vehicles. Control Eng. Pract. 7 (3), 357–368.
Arshad, M.R., 2009. Recent advancement in sensor technology for underwater applications. Indian J. Mar. Sci. 38 (3), 267–273.
Blidberg, D.R., Turner, R.M., Chappell, S.G., 1991. Autonomous underwater vehicles: current activities and research opportunities. Robot. Auton. Syst. 7 (2–3), 139–150.
Cheng, S., Li, Z., Mang, H.-P., Neupane, K., Wauthelet, M., Huba, E.-M., 2014. Application of fault tree approach for technical assessment of small-sized biogas systems in nepal. Appl. Energy 113 (0), 1372–1381.
Choyekh, M., Kato, N., Short, T., Ukita, M., Yamaguchi, Y., Senga, H., Yoshie, M., Tanaka, T., Kobayashi, E., Chiba, H., 2015. Vertical water column survey in the gulf of mexico using autonomous underwater vehicle sotab-i. Mar. Technol. Soc. J. 49 (3), 88–101.
Chu, Z., Zhang, M., 2014. Fault reconstruction of thruster for autonomous underwater vehicle based on terminal sliding mode observer. Ocean Eng. 88, 426–434.
Chu, Z., Zhu, D., Yang, S.X., 2016. Observer-based adaptive neural network trajectory tracking control for remotely operated vehicle. IEEE Trans. Neural Netw. Learn. Syst.. http://dx.doi.org/10.1109/TNNLS.2016.2544786.
Dearden, R., Ernits, J., 2013. Automated fault diagnosis for an autonomous underwater vehicle. IEEE J. Ocean. Eng. 38 (3), 484–499.
Figueroa-Garcia, J.C., Ochoa-Rey, C.M., Avellaneda-Gonzalez, J.A., 2015. Rule generation of fuzzy logic systems using a self-organized fuzzy neural network. Neurocomputing 151 (3), 955–962.
Gmytrasiewicz, P., Hassberger, J., Lee, J., 1990. Fault tree based diagnostics using fuzzy logic. IEEE T. Pattern Anal. 12 (11)), 1115–1119.
Lazakis, I., Dikis, K., Michala, A.L., Theotokatos, G., 2016. Advanced ship systems condition monitoring for enhanced inspection, maintenance and decision making in ship operations. Transp. Res. Proc. 14, 1679–1688.
Lazakis, I., Turan, O., Aksu, S., 2010. Increasing ship operational reliability through the implementation of a holistic maintenance management strategy. Ships Offshore Struct. 5 (4), 337–357.
Lee, W., Grosh, D., Tillman, F., Lie, C., 1985. Fault tree analysis, methods, and applications-a review. IEEE T. Reliab. 34 (3), 194–203.
Li, J.-H., Jun, B.-H., Lee, P.-M., Hong, S.-W., 2005. A hierarchical real-time control architecture for a semi-autonomous underwater vehicle. Ocean Eng. 32 (13), 1631–1641.
Liu, Q., Zhu, D., Yang, S., 2012. Unmanned underwater vehicles fault identification and fault-tolerant control method based on fca-cmac neural networks, applied on an actuated vehicle. J. Intell. Robot. Syst. 66 (4), 463–475.
Liu, W., Wang, Y., Yin, B., Liu, X., Zhang, M., 2016. Thruster fault identification based on fractal feature and multiresolution wavelet decomposition for autonomous underwater vehicle. P.I. Mech. Eng. C- J. Mech. 0 (0), 1–12.
Momma, H., Watanabe, M., Hashimoto, K., Tashiro, S., 2004. Loss of the full ocean depth rov kaiko - part 1: Rov kaiko - a review. In: Proceedings of the Fourteenth (2004) International Offshore and Polar Engineering Conference. Toulon, France, pp. 23–28.
Omerdic, E., Roberts, G., 2004. Thruster fault diagnosis and accommodation for open-frame underwater vehicles. Control Eng. Pract. 12 (12), 1575–1598.
Peng, Z., Wang, J., Wang, D., 2017. Distributed containment maneuvering of multiple marine vessels via neurodynamics-based output feedback. IEEE Trans. Ind. Electron. 64 (5), 3831–3839.
Podder, T.K., Sarkar, N., 2001. Fault-tolerant control of an autonomous underwater vehicle under thruster redundancy. Robot. Auton. Syst. 34 (1), 39–52.
Ramot, D., Friedman, M., Langholz, G., Kandel, A., 2003. Complex fuzzy logic. IEEE T. Fuzzy Syst. 11 (4), 450–461.
Roberts, G.N., Sutton, R. (Eds.), 2006. Advances in Unmanned Marine Vehicles. Control, Robotics & Sensors. Institution of Engineering and Technology.
Ruilin, Z., Lowndes, I.S., 2010. The application of a coupled artificial neural network and fault tree analysis model to predict coal and gas outbursts. Int. J. Coal Geol. 84 (2), 141–152.
Sarkar, N., Podder, T.K., Antonelli, G., 2002. Fault-accommodating thruster force allocation of an auv considering thruster redundancy and saturation. IEEE T. Robot. Autom. 18 (2), 223–233.
Showstack, R., 2014. Unmanned research vessel lost on deep sea dive. Eos Trans. Am.

Geophys. Un. 95 (20), 168.

Takai, M., Ura, T., 1999. Development of a system to diagnose autonomous underwater vehicles. Int. J. Syst. Sci. 30 (9), 981–988.

Wang, N., Er, M.J., 2016. Direct adaptive fuzzy tracking control of marine vehicles with fully unknown parametric dynamics and uncertainties. IEEE Trans. Contr. Syst. Technol. 24 (5), 1845–1852.

Wang, Y., Wilson, P.A., Liu, X., et al., 2015. Adaptive neural network-based backstepping fault tolerant control for underwater vehicles with thruster fault. Ocean Eng. 110, 15–24.

Xiang, X., Lapierre, L., Jouvencel, B., 2015a. Smooth transition of auv motion control: From fully-actuated to under-actuated configuration. Robot. Auton. Syst. 67, 14–22.

Xiang, X., Yu, C., Zheng, J., Xu, G., 2015b. Motion forecast of intelligent underwater sampling apparatus-part i: design and algorithm. Indian J. Mar. Sci. 44 (12), 1962–1970.

Xiang, X., Yu, C., Niu, Z., Zhang, Q., 2016a. Subsea cable tracking by autonomous underwater vehicle with magnetic sensing guidance. Sensors 16 (8), 1335.

Xiang, X., Yu, C., Zhang, Q., 2017. Robust fuzzy 3d path following for autonomous underwater vehicle subject to uncertainties. Comput. Oper. Res. 84, 165–177.

Xiang, X., Yu, C., Zhang, Q., Xu, G., 2016c. Path-following control of an auv: fully actuated versus under-actuated configuration. Mar. Technol. Soc. J. 50 (1), 34–47.

Yeh, C.-Y., Jeng, W., Lee, S.-J., 2011. Data-based system modeling using a type-2 fuzzy neural network with a hybrid learning algorithm. IEEE Trans. Neural Netw. 22 (12), 2296–2309.

Zhang, F., Marani, G., Smith, R.N., Choi, H.T., 2015. Future trends in marine robotics. IEEE Robot. Autom. Mag. 22 (1), 14–122.

Zhang, M.-J., Wang, Y.-J., Xu, J.-A., Liu, Z.-C., 2015. Thruster fault diagnosis in autonomous underwater vehicle based on grey qualitative simulation. Ocean Eng. 105, 247–255.

Zhang, M.-J., Wu, J., Chu, Z.-Z., 2014. Multi-fault diagnosis for autonomous underwater vehicle based on fuzzy weighted support vector domain description. China Ocean Eng. 28 (5), 599–616.

Zhang, W. (Ed.), 2012. Quantitative Process Control Theory. CRC.

Zhu, D., Sun, B., 2013. Information fusion fault diagnosis method for unmanned underwater vehicle thrusters. IET Elect. Syst. Trans. 3 (4), 102–111.