

Fault-tolerant Control of Unmanned Underwater Vehicles with Continuous Faults: Simulations and Experiments

Qian Liu and Daqi Zhu

Laboratory of Underwater Vehicles and Intelligent Systems, Shanghai Maritime University, Shanghai, 200135, China. e-mail:zdq367@yahoo.com.cn

Abstract: A novel thruster fault diagnosis and accommodation method for open-frame underwater vehicles is presented in the paper. The proposed system consists of two units: a fault diagnosis unit and a fault accommodation unit. In the fault diagnosis unit an ICMAC (Improved Credit Assignment Cerebellar Model Articulation Controllers) neural network information fusion model is used to realize the fault identification of the thruster. The fault accommodation unit is based on direct calculations of moment and the result of fault identification is used to find the solution of the control allocation problem. The approach resolves the continuous faulty identification of the UV. Results from the experiment are provided to illustrate the performance of the proposed method in uncertain continuous faulty situation.

Keywords: Fault diagnosis; Fault-tolerant control; Underwater Vehicles; Thrusters.

1. Introduction

The thruster of UV (Underwater Vehicles) is one of the most common and important source of faults, and it always makes direct effect in the control performance, therefore FTC(Fault-Tolerant Control)for thruster fault of UV is a hot research topic nowadays[1]. Some results about FTC of UV have been reported in recent ten years. A fault-tolerant system in an experimental AUV (Autonomous Underwater Vehicle) was outlined by Yoerger, Yang, Yuh and Choi[2-4]. The system was subdivided into individual fault tolerant subsystems for dealing with thruster and sensor failures separately, the thruster subsystem consisted of a rule base for detection and isolation purposes, and a control law reconstruction algorithm was realized by eliminating the corresponding control column. But only a whole (total) fault of the thruster was considered in these researches, and its scope of actual applications and control effects of FTC are poor. In order to improve control performance, the problem of optimal distribution of propulsion forces for over actuated UV is addressed in [5-7] by Podder after 2000. The authors investigate how to exploit the excess number of thrusters to accommodate thruster fault. This approach resolves the thruster redundancy in the Cartesian space and allows the AUV to track the task space trajectories with asymptotic reduction of the task space errors. But only the whole (total) fault of the thruster was still considered, that is different from the actual fault situations of the thrusters.

In recent years, the fault diagnosis method for UV based on the SOM (Self-Organizing Maps) was proposed in [8-9]. The fault accommodation subsystem used weighted

pseudo-inverse to find the solution of the control allocation problem [10-11]. Two approximations (truncation or scaling) are used to ensure feasibility of the solution. The cases of a partial fault for thruster are considered, the thruster with partial fault is typically allowed to continue operation with the restricted factor $s \in (0,1]$. The numerical value of s depends on the type of the fault. Normal state and three different fault cases are considered (jammed, heavy jammed and broken propeller) in literature [9]. However, in actual applications of UV, the degree of jammed is continuous and uncertain. Three or more fault cases can not comprise all kinds of situation. In order to effective diagnosis continuous and uncertain fault of thruster, a new on-line fault identification algorithm which is more closer to the actual condition is urgently needed to be designed.

This paper focuses on continuous and uncertain jammed fault situation, a novel approach to fault diagnosis and fault-tolerant control of thruster for a class of open-frame unmanned underwater vehicles is presented. In this approach, ICMAC neural network [14-15] is used to realize the information fusion fault identification. The fault accommodation unit uses fault information provided by the neural network to accommodate fault and perform an appropriate control reallocation.

This paper is organized as follows: in the second section, the experiment system of the OUTLAND1000, configuration of thrusters, force allocation and the basic control architecture are introduced. In the third and fourth sections, fault diagnosis unit and fault accommodation unit are presented separately. In order to illustrate the proposed method effectively, a simulation example is given. Finally, some concluding remarks are summarized in the fifth section.

2. The experiment system of the OUTLAND1000

The OUTLAND1000 ROV experiment system has 4 major components, the frame can be seen in Fig.1, the sensor system, the thrusters system, the communication system, and the control system.

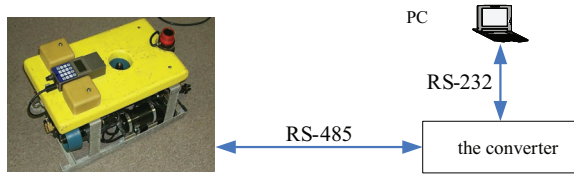


Fig. 1. the system of the OUTLAND1000

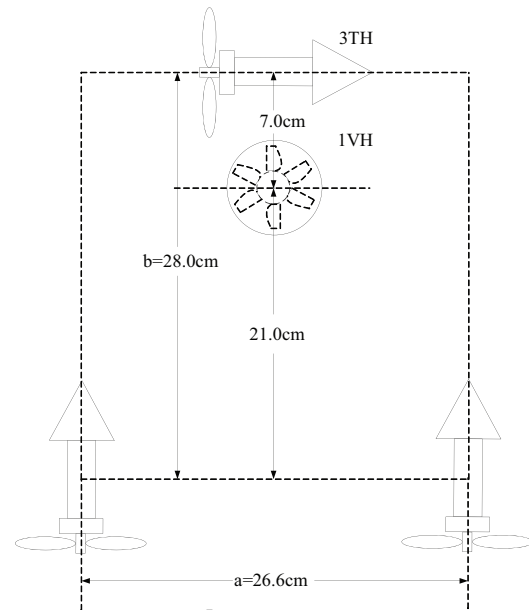
1. the sensor system: including depth gauge, compass and sonar.
2. the thrusters system: the OUTLAND1000 has three horizontal thrusters and one vertical thruster, these will be covered later in this paper.
3. the communication system: including RS485&RS232 interface. The controller and the underwater vehicle are connected by the cable (RS485), the PC and the controller is connected by the series port cable (RS232).
4. the control system on the water: including the converter, interfaces, LCD monitor, PC and hand controller.

2.1 Thruster configuration

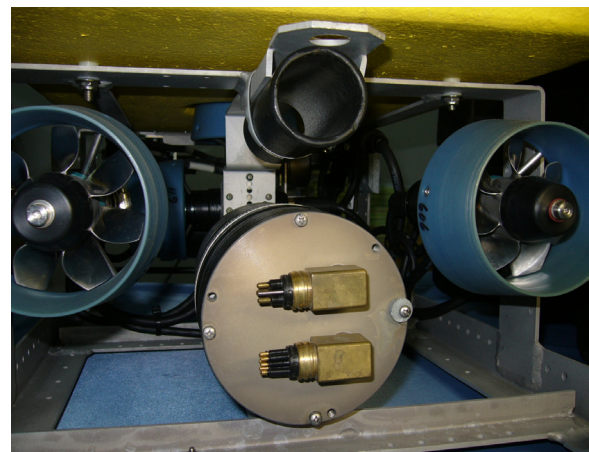
The UV (OUTLAND1000) is used to demonstrate the performance of the fault diagnosis and fault-tolerant system. It has four fixed direction thrusters, the following discussion will be concentrated on this kind of actuators, there are three horizontal thrusters, denoted as HT^i , $i \in [1,3]$ and one vertical denoted as VT^1 . Thruster's configuration of the ROV enables direct control of: surge, yaw and heave, as indicated in Fig.2 Vector τ_d can be decomposed into two parts : $\tau_d = [\tau_d^{HT} \quad \tau_d^{VT}]$, where τ_d^{HT} represents desired surge force τ_x, τ_y and desired yaw moment τ_N for motion in the horizontal plane, the main objective of the fault-tolerant control allocation is to ensure that the ROV can maintain the current status τ_d with a certain degree of fault. Take surge for example, if one of the thrusters is jammed with float grass or something else, the yaw moment τ_N will be changed, and it will go off the course because of the unbalanced moment. How to make the ROV hold to the desired course τ_d in case of failure is primary coverage of the paper.

2.2 Control architecture

A simplified block diagram of full control architecture model is shown in Fig.3, including the fault diagnosis unit and the fault accommodation unit. The desired vector of surge force and moments τ_d is generated by the PC. The force of the each thruster is controlled by the input parameter $u_i \in [-1, 1]$, $i = 1, 2, 3, 4$ which have been



(a) configuration of the thrusters



(b) two sternward thrusters

Fig. 2. configuration of the thrusters

normalized. It means that vector components are divided by their maximum values, such that each thruster is dimensionless number that is between -1 and +1. The ROV has a orientation sensor, which measures actual heading angle of the ROV. A PID controller is used to control the surge and yaw by comparing the actual and the desired heading angle. When fault happens, the fault diagnosis unit will diagnose the fault according to the yaw rate of the heading angle and the current control voltage, the ICMAC is used as a fault diagnosis system to reveal the relationship between the yaw rate, the control voltage and the degree of the fault, and the result of the identification with the information of the fault will be the input of the fault accommodation unit. In this unit, the reasonable reconfiguration of the force and moment with the redundant thruster will be calculated based on the output of the fault diagnosis unit. The tiny difference can be rectified by the PID controller.

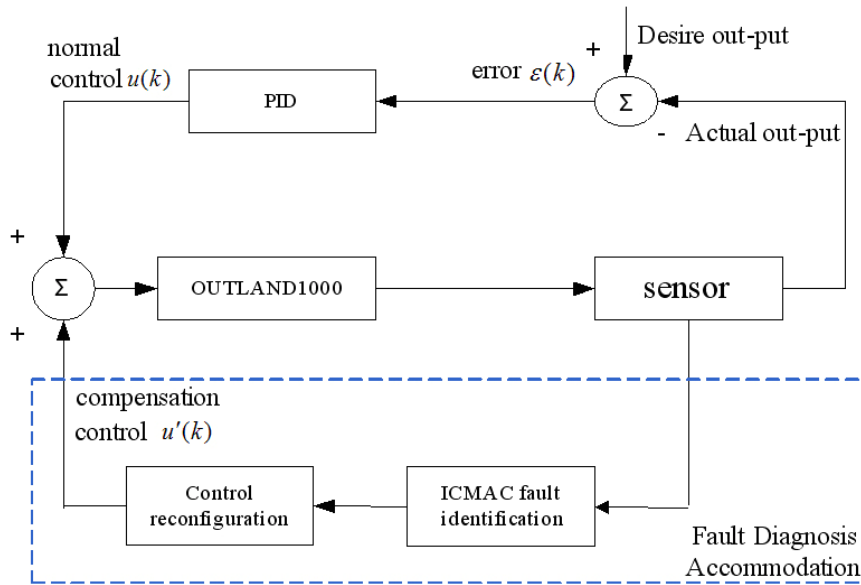


Fig. 3. ROV fault-tolerant control architecture

3. Fault diagnosis method and experiment

The thruster fault is one of the most common sources of the trouble in the system of the UV, and they are classified into two main classes:

1. Internal fault: mainly refers to internal electronic device fault or communication of the controllers, e.g. temperature of the windings is out of range, lost communication between the thruster and main processor, drop in bus voltage, etc.
2. External fault: mainly refers to the fault resulting from the complex dynamic environment. e.g. the propeller jammed by float grass and silt or broken propeller etc.

This paper will focus on external fault diagnosis and fault-tolerant control. A thruster fault may occur due to various reasons. Some of these external faults (partial faults) are not seriously and the thruster is able to continue operation in the presence of a fault with the restricted usage. In Edin Omerdic and Geoff Roberts [9], normal state and three different fault cases are considered (jammed, heavy jammed and broken propeller), however, the output of the SOM fault diagnosis is discrete, it can only give one of the fault cases, jammed or heavy jammed, never a case between them, but in real applications the faulty cases will be much more, and it is continuous, it could be any value between two cases. In this paper, ICMAC network is exploited and used as the online estimator for the unknown continuous fault, and the greatest virtue of the ICMAC is that the output is continuous, so it comes much nearer the practical situation than SOM neural network in fault identification model.

3.1. Fault code table

Fault code table is used to store the relationships between thruster states and fault types. Under the same control

voltage, if the damage of thruster is more extensive, the yaw rate of the heading angle will change faster, meanwhile under the same fault situation, the higher the control voltage is, the yaw rate will change faster also. The ICMAC can be used to mirror these kinds of relationships in the fault diagnosis unit.

The fault is simulated by tangling a piece of string onto right (or left) sternward propeller, the length of the string represents the degree of the fault. In normal case, the ROV will sail in the straight line by the given heading angle. From Fig.2 configuration of the thrusters for OUTLAND1000, if one of thrusters has some jammed fault, the ROV will change its direction, different length of the string means the different degree of fault, and the yaw rate of the heading angle will be different. So the output of the parameter s , which represents the degree of the fault, is correlated with the control voltage and the yaw rate both. The ICMAC can be used to reveal the relationship of them, and the training sample of the ICMAC can get by experiment which 6 kind of fault cases has been simulated as follows:

1. normal state: restricted parameter $s = 1.0$
2. jammed1: restricted parameter $s = 0.75$, tie a piece of string onto right sternward propeller (15cm),
3. jammed2: restricted parameter $s = 0.65$, tie a piece of string onto right sternward propeller(20cm),
4. jammed3: restricted parameter $s = 0.50$, tie a piece of string onto right sternward propeller(30cm),
5. jammed4: restricted parameter $s = 0.25$, tie a piece of string onto right sternward propeller(45cm),
6. broken: restricted parameter $s = 0.0$, one of the sternward propeller is removed from the ROV.

This fault code table is just a suggestion, intended to reveal the main ideas of the proposed fault diagnosis, and the existing relationships can be changed, in order to cover intermediate cases and meet specific requirements.

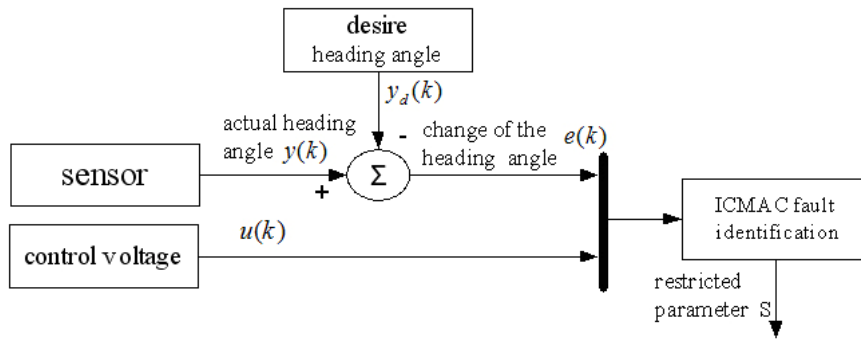


Fig. 4. structure of the information fusion fault diagnosis unit

3.2. Structure of ICMAC for fault diagnosis

The fault diagnosis unit is used for monitoring the thruster states and detecting any faulty situation. The structure of the fault diagnosis unit for stable state can be seen in Fig.3, $y(k)$ is the signal of the sensor, and it is the actual heading angle of the ROV, and the output of the sensor will be compared with the desire heading angle $y_d(k)$ which is given by the PC at the every sampling time, a simple but effective fault detection method is used which is shown as follows:

$$e(k) = \frac{1}{5} \cdot \sum_{k=k_0}^{k_0+5} (y(k) - y_d(k))^2 \quad (1)$$

If $e(k) \geq \lambda$ (fault threshold) means the fault occurs, or else the system is failure-free operation. Figure 4 is the structure of information fusion fault identification based on ICMAC neural network. The yaw rate and $u(k)$ are the inputs of the ICMAC which has been trained by the sample off-line. The output of the ICMAC is restricted parameter s which represents the degree of the fault. By using multi-dimension information fusion processing method based on the ICMAC neural network, it reduces the uncertainty of UV thruster fault diagnosis, and exactly recognizes the fault degree.

In addition, the identification accuracy of ICMAC is higher than conventional CMAC and the error descent is quicker also. In this paper, the ICMAC neural networks fusion model is used to online identify the jammed degree of the thrusters.

3.2.1. Conventional CMAC

The basic concept of CMAC is to store the learning data (knowledge) into overlapped storage hypercube (remembering space), and its output is the addition of data in addressed hypercube. Two kinds of operating are included in the conventional CMAC, one is calculating the output result and the other is learning and adjusting the weight. The mapping structure of CMAC is shown in fig.5. The CMAC network can be applied to approximate function $y = f(x)$, in which $x \in X \subset R^n$, $y \in Y \subset R^m$, and can be realized by mapping $X \rightarrow A \rightarrow Y$, A stands for N dimensional storage space, $a \in A \subset R^N$ is the binary associate vector, and let the input x addressed $N_L (N_L < N)$ storage hypercube; mapping $A \rightarrow Y$ realized the weight sum of the storage hypercube:

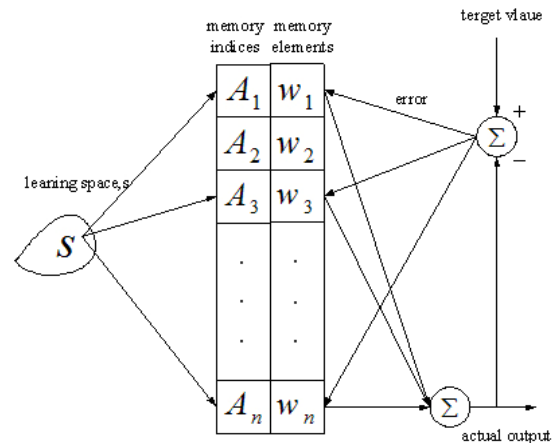


Fig. 5. Basic operational concept of CMAC

$$y_i = \sum_{j=1}^{N_L} w_j a_j(x) \quad i = 1, \dots, m \quad (2)$$

In (2), w_j is weight of the j th storage hypercube, if $a_j(x)$ is addressed, then its values is 1, else is 0, and there are only N_L storage hypercube has an affection to the output.

On the stage of weight learning and adjusting of the network, the conventional CMAC equally distributed the correcting amounts of errors into all addressed hypercube, and to storage hypercube j which is addressed by a certain input x , its weight adjusting rule is:

$$w_j^k = w_j^{k-1} + \beta (y_d - \sum_{j=1}^{N_L} w_j^{k-1} a_j(x)) / N_L \quad j = 1, 2, \dots, N_L \quad (3)$$

Where y_d is the desired value, β is the learning step, and β is a constant in the conventional CMAC, then the errors were equally distributed into N_L addressed hypercube. In the conventional learning process, the errors are equally distributed into the hypercube being addressed. However, not every hypercube has the same learning history at the k training step, hence those hypercube do not have the same credibility. Disregarding such differences, all the addressed hypercube get equal shares for error correcting in (3). As a result, previous learned information may be corrupted due to large error caused by an unlearned state, which makes the weights (stored data) of the hypercube will modified again and again, the learning efficiency of the network is poor.

fault cases	ICMAC training sample							
Normal S = 1	In-put	control voltage	0.25	0.5	0.75	-0.75	-0.5	-0.25
		the yaw rate	0.0	0.0	0.0	0.0	0.0	0.0
	out-put	parameter s	1	1			1	1
jammed1 S = 0.75	In-put	control voltage	0.25	0.5	0.75	-0.75	-0.5	-0.25
		the yaw rate	0.075	0.225	0.165	-0.16	-0.17	-0.04
	out-put	parameter s	0.75	0.75	0.75	0.75	0.75	0.75
jammed3 S = 0.5	In-put	control voltage	0.25	0.5	0.75	-0.75	-0.5	-0.25
		the yaw rate	0.43	0.38	0.415	-0.41	-0.34	-0.28
	out-put	parameter s	0.50	0.50	0.50	0.50	0.50	0.50
jammed4 S = 0.25	In-put	control voltage	0.25	0.5	0.75	-0.75	-0.5	-0.25
		the yaw rate	0.5	0.39	0.42	-0.42	-0.48	-0.68
	out-put	parameter s	0.25	0.25	0.25	0.25	0.25	0.25

Table 1. The sample data

3.2.2. Credit assignment CMAC

In order to improve the learning efficiency of conventional CMAC, the correcting errors should be distributed in accordance to the reliability of hypercube. However, no effective methods have been developed to decide which hypercube should pay much more responsibilities to the current errors. In other words, no good methods have been produced to decide the reliability of the hypercube. The only available information about reliability of the hypercube is the current weight updating times of the hypercube. The more hypercube updates, the more reliable the stored data are. So the learning times of the hypercube is seen as credit index. The more times the addressed hypercube have learned, the higher the credit assignment is, and the less the weight will be adjusted. In the ICMAC the proposed improved learning approach is to use the learned times of the addressed hypercube as the credibility values of hypercube learned, the corrected amounts of errors are proportion to the inversion of the learned times of the addressed hypercube:

$$w_j^k = w_j^{k-1} + \beta \left\{ \frac{(f(j)+1)^{-1}}{\sum_{l=1}^{N_l} (f(l)+1)^{-1}} \right\} (y_d - \sum_{j=1}^{N_l} a_{ff} w_j^{k-1}) \quad (4)$$

The learning accuracy of ICMAC is higher than conventional CMAC and the error descent is very quick. The detailed algorithm and results can be seen in [15-16].

3.2.3. ICMAC information fusion fault identification and experiment result

There are two operating process in ICMAC information fusion fault identification of thruster: off-line training of ICMAC and on-line fault identification.

① Off-line training of ICMAC

The first stage in the training phase is acquisition of training data, normal state and four kinds of different fault cases are considered, and jammed propeller is simulated by tying a piece of string onto right sternward propeller. When the thruster starts, the propeller and the

string rotate together, representing additional load for the motor. In order to simulate broken propeller, blades are removed from the thruster of OUTLAND1000, and the balance of the ROV will be broke, the yaw rate will change also, the data from the sensor $y(k)$ (the yaw rate) and control voltage $u(k)$ will be record in file, and sampling time is 0.2s. 6 kinds of possible thruster velocities are covered, and the experiments are undertaken in the pool. The diagrams of the training data are shown as Table.1, and the data can be used to train the ICMAC off-line, each group has two in-puts and one out-put which correspond to the in-puts and out-put of the ICMAC, because the parameter s is correlated with the control voltage and the yaw rate both. In this way, heavy demanding calculations are performed off-line, which enables fast and efficient fault identification during the on-line phase. It is significant that the case ‘jammed2’ and ‘broken’ will not be trained as the sample data, and it can be used as the test data to prove the result of the ICMAC.

② On-line fault identification

If a fault is detected with equation (1), the current yaw rate of the heading angle and the current control voltage will be sent into the ICMAC which has been trained by fault samples, and the out-put is the corresponding restricted parameter s which means the degree of the fault. A method based on the SOM was proposed in [9], if the fault is just one of the fault cases, the corresponding restricted parameter s will be out-put, but in most of the cases, the fault will not be the sample situation in the table.1, it might be a certain case between jammed2 and jammed3, only the closest case will be chose as the out-put in SOM method, in other words, the output is discrete, it must either be jammed2 or jammed3, and error will impossible to avoid. But the problem can be solved with the ICMAC information fusion, and the out-put will be continuous not discrete, it may be a corresponding certain value between the two cases. The fault accommodation unit uses information of the fault provided by the neural network to accommodate fault and perform an appropriate control reallocation. Finally,

fault cases	result of the ICMAC and SOM							
jammed3 S=0.5	In-put	control voltage	0.25	0.5	0.75	-0.75	-0.5	-0.25
		the yaw rate	0.43	0.38	0.415	-0.41	-0.34	-0.28
	out-put	ICMAC identification	0.500	0.4984	0.4996	0.4938	0.5005	0.5
		SOM identification	0.5	0.5	0.5	0.5	0.5	0.5
jammed2 S=0.65	In-put	control voltage	0.25	0.5	0.75	-0.75	-0.5	-0.25
		the yaw rate	0.30	0.28	0.223	-0.214	-0.25	-0.16
	out-put	ICMAC identification	0.625	0.5708	0.6517	0.6244	0.5937	0.695
		SOM identification	0.5	0.75	0.75	0.75	0.75	0.5
broken S=0	In-put	control voltage	0.25	0.5	0.75	-0.75	-0.5	-0.25
		the yaw rate	0.75	0.55	0.65	-0.616	-0.717	-0.84
	out-put	ICMAC identification	0.0	0.06	0.04	0.07	0.02	0.2
		SOM identification	0.25	0.25	0.25	0.25	0.25	0.25

Table 2. The results of the ICMAC and SOM

out-put of the fault diagnosis units are integrated to form the total restricted parameter $S = [S_1 \ S_2 \ S_3]^T$. In a fault-free case all three parameter S_i are equal to 1, in totally broken case S_i are equal to 0. For example, when the control voltage $u_1 = 0.5$ and the yaw rate $y = 0.1$ (dep), it is between normal and jammed1, the out-put of the SOM will be $S_1 = 1$, because the closest case is normal, that means the thruster is fault-free, but the practical situation is that heading angle of the ROV has changed, and it is fault obviously. By contrast, the out-put of the ICMAC will be $S_1 = 0.9$, the value is between normal and jammed1, and this is more closer to the actual condition. Table.2 is the result of the identification, which show the advantage of the ICMAC, and the SOM will be taken to compare with. If the actual fault case is same as one of the sample data (jammed3), the different of the two results is very small, but in jammed2 and broke cases which haven't been trained as the sample the different will be huge. The SOM can only identify the fault which has been trained, but for the new one, the SOM will only take the nearest case as the out-put, so the result of the SOM will depart from the real situation. But the situation is different when the ICMAC has been used, fault cases jammed2 and broken never turned up in sample data, and the out-put of the ICMAC is more closed to actual situation, which can be seen in Table.2.

4. Fault accommodation method and experiment

In this unit, the reasonable reconfiguration of the force and moment with the redundant thrust will be calculate based on the fault diagnosis unit, and total restricted parameter S has the information about healthy state of each thruster which can be used to solve the control allocation problem.

Because the OUTLAND1000 only has three horizontal thrusters, it's proved insufficient for complexity algorithm. A method of direct calculations of moment has been used for the fault accommodation, and the tiny difference can be rectified by the PID controller.

For orientation control in fault free case, a desired heading angle will be given by PC, and the ROV will

track the desired trajectory with the PID control. Fault-tolerant control method is shown in Fig.3.

A simulated fault experiment has been made to illustrate the accommodation method for orientation control. A rope will be used to simulated the fault, obviously the unbalance force and moment will drive the ROV depart from the desired trajectory, and do an uniform circular motion, and the reasonable reconfiguration of the force and moment with the redundant thrust will be calculated by equation (5):

$$\tau_N = \frac{a}{2}u_1S_1 - \frac{a}{2}u_2 + (b-21)ku_3 \tag{5}$$

τ_N is the yaw moment for motion in the horizontal plane $b = 28.0cm$, $a = 26.6cm$ are the length and width of the ROV(see in fig.2). $\frac{a}{2}u_1S_1 - \frac{a}{2}u_2$ stands for the difference yaw moment of the two sternward thrusters, and $(b-21)ku_3$ stands for the third thruster's moment, k is a parameter for orientation control which is need adjusting by the experiment.

For given desired yaw moment τ_N^d , find the appropriate control voltage u_3 , such that $\tau_N = \tau_N^d$.

$$\tau_N = \frac{a}{2}u_1S_1 - \frac{a}{2}u_2 + (b-21)ku_3 = \tau_N^d \tag{6}$$

$$u_3 = \frac{\tau_N^d - [\frac{a}{2}u_1S_1 - \frac{a}{2}u_2]}{(b-21)k} \tag{7}$$

When thruster fault happens, the loss force of the thruster can be estimated according to the restricted parameter S from the ICMAC, and equation (7) is used to find the reasonable reconfiguration with the redundant thrust, the ROV can be driven back to the desired state. In this paper only fault in single thruster case is considered because of the quantitative restriction of the thrusters.

For orientation control, parameter k is a constant which can be calculated by simulating one of the fault cases, and

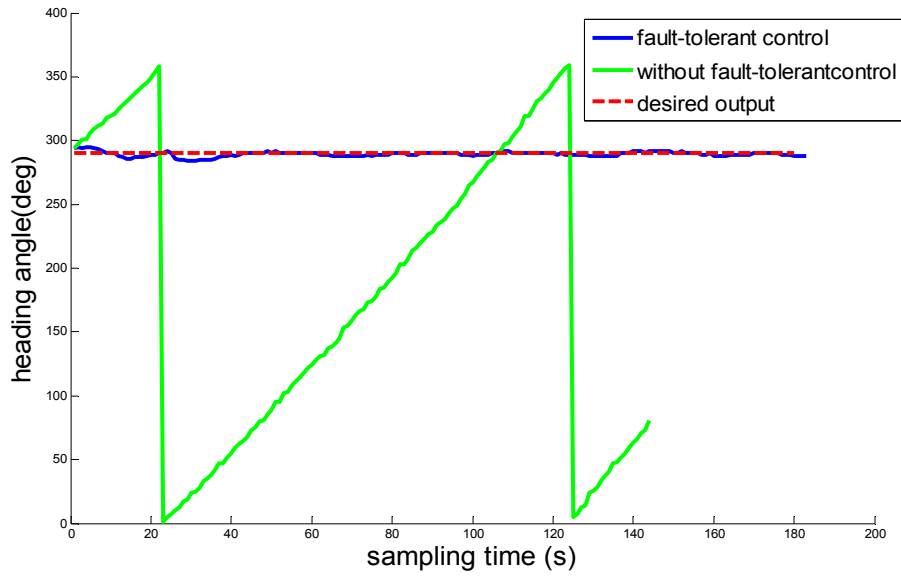


Fig. 6. The desired output vs. reconfiguration output

Sample time	without Fault tolerant Control	Fault-tolerant Control	Error with FTC	Error without FTC	Sample time	without Fault tolerant Control	Fault-tolerant Control	Error with FTC	Error without FTC
1	294	294	-4	-4	43	73	290	0	217
2	297	295	-5	-7	44	75	290	0	215
3	301	294	-4	-11	45	80	290	0	210
4	301	295	-5	-11	46	81	290	0	209
5	306	295	-5	-16	47	85	290	0	205
6	309	294	-4	-19	48	89	290	0	201
7	312	293	-3	-22	49	95	292	-2	195
8	313	292	-2	-23	50	95	290	0	195
9	318	290	0	-28	51	102	292	-2	188
10	319	290	0	-29	52	103	290	0	187
11	319	290	0	-29	53	108	290	0	182
12	321	288	2	-31	54	111	290	0	179
13	325	287	3	-35	55	114	290	0	176
14	328	286	4	-38	56	118	290	0	172
15	331	286	4	-41	57	122	290	0	168
16	334	287	3	-44	58	124	290	0	166
17	337	287	3	-47	59	128	290	0	162
18	340	287	3	-50	60	131	290	0	159
19	343	288	2	-53	61	132	289	1	158
20	345	289	1	-55	62	137	289	1	153
21	358	289	1	-68	63	139	289	1	151
22	1	290	0	289	64	137	288	2	153
23	5	290	0	285	65	145	288	2	145
24	7	292	-2	283	66	153	288	2	137
25	10	290	0	280	67	155	288	2	135
26	13	286	4	277	68	159	288	2	131
27	17	285	5	273	69	163	288	2	127
28	19	285	5	271	70	166	288	2	124
29	24	284	6	266	71	168	288	2	122
30	25	284	6	265	72	173	288	2	117

31	28	284	6	262	73	174	289	1	116
32	33	285	5	257	74	178	288	2	112
33	35	285	5	255	75	184	289	1	106
34	38	285	5	252	76	185	289	1	105
35	42	285	5	248	77	189	289	1	101
36	47	286	4	243	78	192	289	1	98
37	47	287	3	243	79	196	290	0	94
38	52	288	2	238	80	203	290	0	87
39	55	288	2	235	81	203	290	0	87
40	59	289	1	231	82	208	290	0	82
41	62	288	2	228	83	214	289	1	76
42	68	289	1	222	84	216	289	1	74
85	223	289	1	67	106	307	288	2	-17
86	227	290	0	63	107	311	288	2	-21
87	228	290	0	62	108	319	288	2	-29
88	234	290	0	56	109	319	289	1	-29
89	236	290	0	54	110	325	288	2	-35
90	238	290	0	52	111	331	289	1	-41
91	243	290	0	47	112	333	289	1	-43
92	247	290	0	43	113	337	289	1	-47
93	249	290	0	41	114	340	289	1	-50
94	258	290	0	32	115	345	290	0	-55
95	265	290	0	25	116	349	290	0	-59
96	267	290	0	23	117	353	290	0	-63
97	271	289	1	19	118	357	290	0	-67
98	275	289	1	15	119	359	289	1	-69
99	279	289	1	11	120	5	289	1	285
100	283	288	2	7	121	7	289	1	283
101	290	288	2	0	122	13	290	0	277
102	292	288	2	-2	123	14	288	2	276
103	297	288	2	-7	124	24	288	2	266
104	298	288	2	-8	125	26	288	2	264
105	303	288	2	-13	126	29	289	1	261

Table 3. The result data of the fault-tolerant control and error

the desired yaw moment $\tau_N^d = 0$, the control voltage u_3 are adjusting to make that $\tau_N = \tau_N^d = 0$, when the ROV is back into its desired state, record the last value, as show in equation(8), and all the variable become to known parameters:

$$k = \frac{\tau_N^d - [\frac{a}{2}u_1S_1 - \frac{a}{2}u_2]}{(b-21)u_3} \quad (8)$$

In actual experiment, control voltage $u_1 = u_2 = 0.5$, yaw rate is 0.225(dep), and the out-put of the ICMAC (restricted parameter) is $S_1 = 0.75$, and $u_3 = 0.28$ which can be get by cyclical adjustment, equation (8) will be used to get the parameter $k = 0.848$. So, the final expression of the u_3 for orientation fault-tolerance control is:

$$u_3 = 2.24[u_1S_1 - u_2] \quad (9)$$

An experiment is given to demonstrate the result of the fault accommodation method. Fig.6 shows the result of the orientation fault-tolerant control, The desired heading angle of the ROV is 290° normal control voltage $u_1 = u_2 = 0.5$, and restricted parameter $S_1 = 0.7$ which is get from the ICMAC, when fault happens, heading angle of the ROV is change from 0° to 360° , do uniform circular motion, seriously deviated from its desired heading angle (green line). In order to maintain the balance of the moment, the third thruster control voltage $u_3 = 0.336$ can be calculated by equation(7), and the data of the experiment will be seen in Table.3, the out-put without fault tolerant control is changed a lot, but the result fault-tolerant control ranges quite limited, the error of the output is decrease gradually, finally, the reconfiguration control voltage make the heading angle of $290 \pm 2^\circ$ (blue line),the fault-tolerant control system successfully drives the unmanned underwater vehicles system back to the desired state (290° orientation control state) with the fault.

5. Conclusion

An efficient fault diagnosis unit and fault accommodation unit are presented in the paper. In the fault diagnosis unit ICMAC neural network is used to realize the fault identification. In fault accommodation unit, a method of direct calculations of moment is used to find the solution of the control allocation problem. The fault-tolerant control system successfully drives the unmanned underwater vehicles system back to the desired state with the unknown continuous fault.

6. Acknowledgements

This project is supported by the National Natural Science Foundation of China (50775136), the 863 High-technology Research and Development Program of China (2006AA09Z210) and the Natural Science Foundation of Shanghai (07ZR14045).

7. References

- [1] Gianluca A., Underwater Robots Motion and Force Control, Springer-Verlag Berlin Heidelberg, 2006, Second Edition, pp. 78-93.
- [2] Yoerger D.G. and Slotine J.J.E., Adaptive sliding control of an experimental underwater vehicle, *Proceedings of IEEE International Conference on Robotics and Automation, California, USA, 1991*, pp.2746-2751.
- [3] Yang K. C., Yuh J. and Choi S.K., Experimental study of fault-tolerant system design for underwater robots, *Proceedings of the IEEE international conference on Robotics and Automation, Leuven Belgium, 1998*.
- [4] Yang K.C. and Choi S.K., Fault-tolerant system design of an autonomous underwater vehicle-ODIN: an experimental study, *International Journal of Systems Science, 1999, 30 (9)*, pp.1011-1019.
- [5] Podder T.K. and Sarkar N., Fault-tolerant decomposition of thruster forces of an autonomous underwater vehicle, *Proceedings of the IEEE international conference on Robotics and Automation Detroit, MIT, USA, 1999*.
- [6] Podder T.K., Antonelli G. and Sarkar N., Fault tolerant control of an autonomous underwater vehicle under thruster redundancy: Simulation and experiments, *Proceedings of IEEE International Conference on Robotics and Automation, San Francisco, USA, 2000*, pp.276-285.
- [7] Podder T.K. and Sarkar N., Fault-tolerant control of an autonomous underwater vehicle under thruster redundancy, *Robotics and Autonomous Systems, 2001, 34(1)*, pp.39-52.
- [8] Cuadrado A.A. and Diaz I., Fuzzy inference map for condition monitoring with self-organizing maps, *International Conference fuzzy logic and technology, Leicester, UK, 2001*, pp.55-58
- [9] Edin O. and Geoff R., Thruster fault diagnosis and accommodation for open-frame underwater vehicles, *Control Engineering Practice, 2004, 12 (12)*, pp.1575-1598
- [10] Fossen T.I., *Guidance and control of ocean vehicles*. Chichester: Wiley, 1995
- [11] Tondel P., Johansen T.A. and Bemporad A., Evaluation of piecewise affine control via binary search tree, *Automatic, 2003, 39 (3)*, pp.743-749.
- [12] Johansen T.A., Fossen T.I. and Tndel P., Efficient optimal constrained control allocation via multi-parametric programming, *AIAA Journal of Guidance, Control and Dynamics, 2002*.
- [13] Fossen T.I., *Marine control systems*. Trondheim, Norway: Marine Cybernetics AS, 2002.
- [14] Omerdic E., Roberts G.N. and Ridao P., Fault detection and accommodation for ROVs, *Sixth IFAC conference on manoeuvring and control of marine craft .MCMC, Girona, Spain, 2003*.
- [15] Zhu D.Q. and Kong M., A fuzzy CMAC neural network model based on credit assignment, *International Journal of Information Technology, 2006, 12(6)*, pp.1-8.
- [16] Zhu D.Q. and Kong M., Adaptive Fault-tolerant Control of Nonlinear System: An Improved CMAC based fault learning approach, *International Journal of Control, 2008, 80(10)*, pp.1576-1594.