

An Active Fault-Tolerant Control Method Ofunmanned Underwater Vehicles with Continuous and Uncertain Faults

Daqi Zhu, Qian Liu and Yongsheng Yang

Laboratory of Underwater Vehicles, College of Information Engineering, Shanghai Maritime University, Shanghai, China, 200135. e-mail:zdq367@yahoo.com.cn

Abstract: This paper introduces a novel thruster fault diagnosis and accommodation system for open-frame underwater vehicles with abrupt faults. The proposed system consists of two subsystems: a fault diagnosis subsystem and a fault accommodation sub-system. In the fault diagnosis subsystem a ICMAC(Improved Credit Assignment Cerebellar Model Articulation Controllers) neural network is used to realize the on-line fault identification and the weighting matrix computation. The fault accommodation subsystem uses a control algorithm based on weighted pseudo-inverse to find the solution of the control allocation problem. To illustrate the proposed method effective, simulation example, under multi-uncertain abrupt faults, is given in the paper. **Keywords:** Fault diagnosis; Fault-tolerant control; Underwater Vehicles; Thruster; Biomimetic;

1. Introduction

Thrusters of UV(underwater vehicles) are one of the most common and most important sources of faults, and its faults always make direct effect in the control performance, therefore FTC(fault-tolerant control)for thruster of UV is a research hot topic nowadays (Gianluca, A., 2006). Some results of FTC about UV had been reported in recent ten years. A fault-tolerant system for use in an experimental AUV (Autonomous Underwater Vehicle) was outlined by Yoerger, Yang, Yuh, and Choi (Yoerger, D. & Slotine, E., 1991; Yang, K. & Yuh, J., 1998; Yang, C. & Choi.K., 1999). 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 column. But only a whole(total) fault of the thruster was considered in these researches, and 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 (Podder, K. & Sarkar, N., 1999; Podder, K. & Antonelli, G., 2000; Podder, K. & Sarkar, N., 2001) by Podder T.K. after 2000. The authors investigate how to exploit the excess number of thrusters to accommodate thruster faults. 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 was different from the actual fault situations of the thrusters. In recent years, the method based on the SOM(selforganizing maps) was proposed in (Cuadrado,A. & Geoff,R.,2004). Edin,O. & The fault Diaz,I.; accommodation subsystem used weighted pseudoinverse to find the solution of the control allocation problem (Fossen, T., 1995; Tondel, P., 2003). Two approximations (truncation or scaling) can be used to ensure feasibility of the solution. In the case of a partial fault, the thruster 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(Edin,O. & Geoff, R., 2004). However, in actual applications of UV, the degree of jammed is continuous and uncertain. So a new on-line identification algorithm which is much more closer to the actual condition is urgently needed to be designed.

This paper focuses on multi-uncertain abrupt timevariant faults 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(Zhu, Q. & Kong, M.,2006; Zhu, Q. & Kong, M., 2007) is used to realize the on-line fault identification. The fault accommodation unit uses information of the fault 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 basic control architecture, thruster configuration and force allocation are introduced. In the third section, fault diagnosis subsystem and fault accommodation subsystem is presented separately. To illustrate the proposed method, a simulation example, is given in the fourth section. Finally, some concluding remarks are made in the fifth section.

2. Thruster configuration

2.1. Control architecture

The UV (FALCON, seen Fig.1.) (Edin,O.&Geoff,R.,2004) are used to demonstrate the performance of the fault diagnosis and fault-tolerant system. This ROV has no other actuators except fixed direction thrusters, and the following discussion will be concentrated on this type of actuators, while more information about other types can be found in(Johansen,A.&Fossen,I.,2004; Fossen, I.,2002). It has four horizontal thrusters, denoted as ${}^{i}HT, i \in [1,4]$ and one vertical ${}^{1}VT$ (not shown in Fig.1-bottom). Thruster configuration of the FALCON enables direct control of 4 DOF(Degree Of Freedom): surge, sway, yaw and heave.

Vector τ_d can be decomposed into two parts as $\tau_d = [\tau_d^{HT} \quad \tau_d^{VT}]^T$, where τ_d^{HT} represents desired surge force τ_X , sway force τ_Y and yaw moment τ_N for motion in the horizontal plane, and τ_d^{VT} is equal to heave force τ_Z for motion in the vertical plane. The main objective of the control allocation u_i is to ensure that the condition $\tau = \tau_d$ is satisfied for all attainable τ_d .

2.2 Thrusters force allocation of the FALCON

In the paper only the horizontal plane motion control is discussed. Each thruster exerts thrust (force) **T** and torque (moment) **Q** The position vector ${}^{i}r = [{}^{i}r_{x} {}^{i}r_{y} {}^{i}r_{z}]^{T}$



Fig. 1. configurations of the horizontal thruster

determines the position of the point of attack of the force **T**. The orientation of the thruster is defined by the unit vector ${}^{i}e = [{}^{i}e_{x} {}^{i}e_{y} {}^{i}e_{z}]^{T}$. The force **T** also generates the moment **Q=r**×**T**. Vector of forces and moments can be written as:

$$T_{i} = Ku_{i}, Q_{i} = r_{i} \times Ku_{i}, i = 1,2,3,4$$
(1)

Where *K* is the motor control parameter, superposition of the $A = (b/2)\sin \alpha + (a/2)\cos \alpha$ leads to total vector of propulsion forces and moments τ in the horizontal plane:

$$\tau = \begin{bmatrix} \tau_{X} \\ \tau_{Y} \\ \tau_{N} \end{bmatrix} = \begin{bmatrix} K \cos \alpha & K \cos \alpha & K \cos \alpha & K \cos \alpha \\ K \sin \alpha & -K \sin \alpha & K \sin \alpha & -K \sin \alpha \\ KA & -KA & -KA & KA \end{bmatrix} \begin{bmatrix} u_{1} \\ u_{2} \\ u_{3} \\ u_{4} \end{bmatrix} = B \cdot u$$
(2)

Where $\tau = [\tau_x \quad \tau_y \quad \tau_N]^T$ are represented as the motion of surge, sway and yaw respectively. Each component of the control vector is limited by constraint:

$$-u_m \le u_i \le u_m$$
 $i = 1, 2, 3, 4$ (3)

Where u_m is the thruster maximum control velocity, from (1) and (2):

$$\tau_{Xm} = 4T_m \cos \alpha = 4Ku_m \cos \alpha$$
$$\implies K \cos \alpha = \frac{\tau_{Xm}}{4u_m}$$
(4)

$$\tau_{\rm Ym}=4T_m\sin\alpha=4Ku_m\sin\alpha$$

$$\implies K \sin \alpha = \frac{\tau_{Ym}}{4u_{m}} \tag{5}$$

$$\tau_{Nm} = 4T_m A = 4Ku_m A$$

$$\implies KA = \frac{\tau_{Nm}}{4u_m} \qquad 6)$$

Constraint (3) represents thruster velocity saturation, because the thruster cannot rotate faster than its maximum velocity. So from(4) (5) (6), expression (2) can be rewritten in the normalized form (7), detailed can be seen in literature(Podder,K.&Antonelli,G.).

In (7), $\tau_m = [\tau_{Xm} \ \tau_{Ym} \ \tau_{Nm}]^T$ are represented as maximum values of the surge and sway forces and the yaw moment. Finally, the general constrained control allocation problem for open-frame UV can be formulated as: For given normalized $\underline{\tau}$, find \underline{u} feasibility such that.

$$\underline{\tau} = \underline{B} \cdot \underline{u}$$

$$\begin{bmatrix} \frac{\tau_{X}}{\tau_{Xm}} \\ \frac{\tau_{Y}}{\tau_{Ym}} \\ \frac{\tau_{N}}{\tau_{Nm}} \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} \\ \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & \frac{1}{4} \end{bmatrix} \cdot \begin{bmatrix} \frac{u_{1}}{u_{m}} \\ \frac{u_{2}}{u_{m}} \\ \frac{u_{3}}{u_{m}} \\ \frac{u_{3}}{u_{m}} \\ \frac{u_{4}}{u_{m}} \end{bmatrix} \Leftrightarrow \underline{\tau} = \underline{B} \cdot \underline{u}$$
(7)

For formula(7), when the thruster fault-free, the motor control voltage u will produce the corresponding rotational speed n, and get the corresponding propulsion forces and moments τ ; But at thruster fault situation, with the same motor control voltage u, the thruster can not produce the same rotational speed n, and do not get the corresponding propulsion forces and moments au , there is some propulsion forces and moments will lose, if we still use formula (7) to calculate propulsion forces and moments au , it is not accurate. Therefore at thruster fault condition, it is better to use the actual rotational speed of thruster n to compute propulsion forces and moments, generally speaking, the rotational speed of thruster is approximately proportional to the motor control voltage, so for"FALCON" formula (7) can be rewritten as:

$$\begin{bmatrix} \frac{\tau_{X}}{\tau_{Xm}} \\ \frac{\tau_{Y}}{\tau_{Ym}} \\ \frac{\tau_{N}}{\tau_{Nm}} \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} \\ \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & \frac{1}{4} \end{bmatrix} \cdot \begin{bmatrix} \frac{n_{1}}{n_{m}} \\ \frac{n_{2}}{n_{m}} \\ \frac{n_{2}}{n_{m}} \\ \frac{n_{3}}{n_{m}} \\ \frac{n_{3}}{n_{m}} \\ \frac{n_{4}}{n_{m}} \end{bmatrix} \Leftrightarrow \underline{\tau} = \underline{B} \cdot \underline{n}^{(8)}$$

Where formula (8) $-1 \le \underline{\tau}, \underline{n} \le 1$. Components of the vectors $\underline{\tau}$ and \underline{n} are dimensionless number, restricted to the standard interval [-1, 1], This enables better understanding and easier visualisation of the problem. The details please to see the reference (Edin,O.&Geoff,R.,2004)

In the paper formula (8) can be used to calculate the propulsion forces and moments $\underline{\tau}$. By comparing actual propulsion forces and moments $\underline{\tau}$ to desired propulsion forces and moments $\underline{\tau}_d$, and we will find the fault-tolerant control performance.

2.3 *The relationship between thrusters faults and weighting matrix*

In the control system of UV with multi-thruster, a weighting matrix is designed and can be used to decide which thruster should be used primarily. The weighting matrix $W^{\mu\tau}$ (in the horizontal plane) is chosen to be a diagonal matrix:

$$W^{HT} = \begin{bmatrix} W_1^{HT} & 0 & 0 & 0 \\ 0 & W_2^{HT} & 0 & 0 \\ 0 & 0 & W_3^{HT} & 0 \\ 0 & 0 & 0 & W_4^{HT} \end{bmatrix}$$
(9)

In the fault-free case, all horizontal thrusters have the same priority and W^{HT} is chosen to be equal to identity matrix. In the case of a partial fault, the thruster is typically allowed to continue operation with the new restricted usage:

$$-s_i \le u_i \le s_i$$
 $i = 1, 2, 3, 4$ (10)

The restricted factor value S_i is $0 \le s_i < 1$, and depends on the fault value. $s_i = 1$, which means that thruster has non-fault; When $s_i \rightarrow 0$ means thruster has total fault, and completely ineffective; When $0 \le s_i < 1$, means that thruster has partial fault. s_i can be calculated by (11):

$$s_i = \frac{n - f_i}{n} = 1 - \frac{f_i}{n}$$
 $i = 1, 2, 3, 4$ (11)

Where *n* is normal shaft speed of the thruster, f_i is the fault level which is the output of the neural network. In this paper an ICMAC neural network is used to realize the on-line fault identification. In addition to the change of the constraint bounds, the weight W_i^{HT} of the faulty thruster is increased using:

$$W_i^{HT} = 1 + \Delta W_i^{HT}$$
 $i = 1, 2, 3, 4$ (12)

Where:

$$\Delta W_i^{HT} = \frac{1}{s_i} - 1 \quad i = 1, 2, 3, 4 \tag{13}$$

The weight update (12) is introduced to penalise the faulty thruster, prioritise healthy thrusters and to compensate restricted usage of the faulty thruster in an optimal way.

3. Fault diagnosis and accommodation system

3.1. Fault diagnosis subsystem

Thruster fault patterns are classified into two classes (Omerdic, E &Roberts, N), ①Internal faults (e.g. temperature of the winding is out of range, lose communication, drop in bus voltage, etc.), ②External faults (e.g. jammed or broken propeller). In two classes thruster faults, external faults is main and common. Thrusters usually work under the formidable circumstance. A thruster fault can occur due to various reasons. Some of these faults (partial faults) are not critical and the thruster is able to continue operation in the presence of a fault with the restricted usage. In literature(Edin,O.&Geoff,R.,2004) the restricted constraint bound s_i is discrete. Only normal state and three different fault cases were considered (jammed, heavy jammed and



Fig. 2. the fault diagnosis subsystem

broken propeller), however, in actual applications the number of faulty cases can be very much, and it is continuous change and uncertain. it is not suitable to regard the jammed fault f_i as several fault states. In the paper the ICMAC neural networks (Zhu, Q. & Kong, M., 2006; Zhu, Q. & Kong, M., 2007) is used to online identify the jammed degree of the thrusters.

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 the adjustment of weight (the corrected amounts of errors). In the conventional CMAC learning scheme, the corrected amounts of errors are equally distributed into all addressed hypercubes, regardless of the credibility of those hypercubes values. But the correcting errors should be distributed in accordance to the hypercubes reliability, the only available information about hypercubes reliability is the current weight updating times of the hypercube. The more the 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 hypercubes as the credibility values of hypercubes learned, the corrected amounts of errors are proportion to the inversion of the learned times of the addressed hypercubes. 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 (Zhu, Q. & Kong, M., 2006).

The Fig.2 is structure of the fault diagnosis subsystem. Control signals given by Fault Accommodation Unit, can be transferred to normal thruster model and actual faults thruster respectively. The output of the model is desired output in every step. Actual thruster is simulated by a DC motor control system model with unexpected fault, where fault is assumed to be unknown. During the normal state, the nominal control signal drives the behaviour. Once an abnormal system behavior is detected by the fault detection subsystem, an online estimator initially starts to learn the failure by using the difference between the desired output and actual output as the desired target of the ICMAC network, and after some steps of training the network will output the forecast value of fault, no matter what kind of fault (i.e. some state range between jammed and heavy jammed), and the



Fig. 3. ROV control structure

control law is computed exactly and reconfigured by the fault accommodation subsystem based on the current knowledge of the failure dynamics provided by the online estimator.

3.2 Fault accommodation subsystem

A standard ROV control structure is shown in Fig.3. The description of the architecture will be given in two subsystem: fault diagnosis subsystem and fault accommodation subsystem. The desired vector of propulsion forces and moments $\underline{\tau}_{d}$, smoothed by filter is generated by the HCU(Hand Control Unit). The control law is reconfigured by the fault accommodation unit based on the current knowledge of the failure dynamics provided by the online estimator ICMAC. Here we consider the situation when two thrusters are faulted. Our objective is to reallocate the thruster forces among the functioning thrusters in such a way that the UV still follows the desired trajectories. In order to incorporate the thruster faults into the mathematical formulation, we introduce the weighted pseudo-inverse of the thruster control matrix(Fossen,T.,1995;Tondel,P.,2003).

For given $\underline{\tau}_d$, subject to $\underline{Bu} = \underline{\tau}_d$, \underline{u} is given by,

$$\underline{u} = \underline{B}_{w}^{+} \cdot \underline{\tau}_{d} = (W^{-1} \underline{B}^{T} (\underline{B} W^{-1} \underline{B}^{T})^{-1}) \underline{\tau}_{d}$$
(14)

Where \underline{B}_{w}^{+} is weighted pseudo-inverse of \underline{B} . For motion in the horizontal plane the weighted pseudo-inverse matrix \underline{B}_{w}^{+} is given by (Edin,O. & Geoff,R.,2004):

$$\underline{B}_{w}^{+} = \frac{1}{\sum_{i=1}^{4} w_{i}^{HT}} \times \begin{bmatrix} 2(w_{3}^{HT} + w_{4}^{HT}) & 2(w_{2}^{HT} + w_{3}^{HT}) & 2(w_{2}^{HT} + w_{4}^{HT}) \\ 2(w_{3}^{HT} + w_{4}^{HT}) & -2(w_{1}^{HT} + w_{4}^{HT}) & -2(w_{1}^{HT} + w_{3}^{HT}) \\ 2(w_{1}^{HT} + w_{2}^{HT}) & 2(w_{1}^{HT} + w_{4}^{HT}) & -2(w_{2}^{HT} + w_{4}^{HT}) \\ 2(w_{1}^{HT} + w_{2}^{HT}) & -2(w_{2}^{HT} + w_{4}^{HT}) & 2(w_{1}^{HT} + w_{4}^{HT}) \end{bmatrix} \begin{bmatrix} (15) \\ (15)$$

4. Simulation of the online fault diagnosis and faulttolerant control

4.1. Simulation of fault diagnosis and fault-tolerant control

A large number of open-frame underwater vehicles use DC motor to drive the propeller. It is well known that a DC motor can be modeled as a linear time-invariant system(Lingli,N.,2001), the nominal transfer function can be represented by equation (16):

$$G(s) = \frac{\mathbf{n}(s)}{u(s)} = \frac{K_m}{R_a(J \cdot s + d) + K_b K_m}$$
(16)

Where n(s), u(s) denote the rotational speed and motor voltage in s domain, respectively. K_b , K_m , R_a , J and d are motor constants. Equation(16)can be reorganized as:

$$A \cdot \dot{n} + b \cdot n = c \cdot u \tag{17}$$

Using the forward Euler approximation, the discrete time nominal model can be derived and shown in equation(18):

$$n(k+1) = \left[1 - b \cdot \frac{\Delta t}{A}\right] \cdot n(k) + \left[c \cdot \frac{\Delta t}{A}\right] u(k) = f_{linear} n(k) + g_{linear} u(k)$$
⁽¹⁸⁾

Where n(k+1) and u(k) represent the system output and control input at the time step k+1 and k, respectively. In the paper $f_{linear} = 0.9$, $g_{linear} = 0.2$. Under two unexpected abrupt faults modes $f_i(k) = U(k-T)q_i(k)$, the every thruster system is represented by the following equations:

$$n_{i}(k+1) = f_{linear}n_{i}(k) + g_{linear}u_{i}(k) - U(k-T)q_{i}(k)$$
(19)

Where two unexpected abrupt faults are generated at the first thruster and third thruster respectively, $q_1(k) = 0.1 \cdot \cos(\frac{\pi}{100} \cdot k) + 0.2$, $T_1 = 200$, $q_3(k) = 0.1 \cdot \sin(\frac{\pi}{100} \cdot k) + 0.15$, $T_3 = 40$ are assumed to be unknown, and unexpected abrupt failure f_i is identified on-line by ICMAC.

$$U(k-T) = \begin{cases} 0 & k < T \\ 1 & k \ge T \end{cases}$$
(20)

Let desired $\underline{r}_{d} = \begin{bmatrix} 0 & 0.5 & 0.2 \end{bmatrix}^{r}$ for thruster configuration, a fault happens at the first thruster with the step 200, the third thruster with the step 40, and fault-free in others. All of these are assumed to be unknown in order to test the effect of control system.

The control objective in this experiment is to maintain the $\underline{\mathcal{T}}_d$ at the desired output with two unanticipated and continuous faults. Firstly, fault diagnosis unit detects the states of the thrusters and identifies fault vector f_1 and f_3 , then using equation (11) we can get the vector S_i , which relates saturation bounds of each thruster. According to equation (12-13) we can update the weighting matrix W^{TH} , and finally with the equation (14-15) we can find the new, feasible control vector \underline{u} for the given input $\underline{\mathcal{T}}_d$. Using equation (19) we can calculate thruster rotational speed $n_i(k+1)$ with fault-tolerant control law, and further evaluate fault-tolerant control result by using equation (8).

4.2. Analysis of fault-tolerant control results

Fig.4. shows the actual rotational speed of the four thrusters and the fault-free speed within the steps 600. When the unknown abrupt faults happen at the first thruster at the step 200, the third thruster at the step 40 respectively. As seen, only the speed of the first and third thruster performance degrades at the time steps 200 and 40 respectively, and the rest of the thrusters are fault-free. Fig.5. shows output of the forces and moments and the magnitude error without the fault-tolerant control unit. We can see the system performance degrades from the $t_x = 0$ (surge), $t_y = 0.5$ (sway), $t_x = 0.2$ (yaw).

The magnitude error is generated after the time steps 40, and it becomes more large after the time steps 200. It is because the unknown abrupt faults happen at the first thruster after the step 200, the third thruster after the step 40 respectively.



Fig. 4.The actual speed vs. fault-free speed of the motor without the fault-tolerant control



Fig. 5. The actual output vs. desired output without the FTC and magnitude error



Fig. 6.The control voltage of the fault-tolerant control



Fig. 7.The actual output vs. desired output with the FTC and magnitude error

Fig.6. shows the control voltage of all thrusters, we can see that all thrusters have been accommodated according to the faults, the first and third thrusters with fault are penalized by increasing its weight, in order to maintain the balance of the forces and moments, and the corresponding thrusters are all changed according to the weighting matrix.

Fig.7. shows output of the forces and moments and the magnitude error with the fault-tolerant control unit. Clearly that the system performance is greatly improved. The fault accommodation subsystem successfully drives the output of the unknown faulty system back to the desired state: $\underline{r}_d = [0 \quad 0.5 \quad 0.2]^T$. From the error we can see that the fault is actually detected by the fault diagnosis subsystem after the step 40 and 200 respectively, and the ICMAC neural networks is initialized and learns the unknown faults online. The system becomes stability after the step 215 with the very small error.

5.Conclusion

In this paper, the online fault accommodation for openframe underwater vehicles system is investigated. An efficient online fault diagnosis subsystem and fault accommodation subsystem are presented. The online fault diagnosis subsystem ICMAC greatly improves the performance of the on-line faults identification. With the algorithm based upon weighted pseudo-inverse, the error has been greatly reduced. The fault-tolerant control system successfully drives the system back to the desired state with the unknown abrupt faults.

6. Acknowledgment

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

- Alessandri, A. Caccia, M. and Veruggio, G. (1997,7) Fault detection of actuator faults in unmanned underwater vehicles, Control Engineering Practice, P.357-368.
- Cuadrado,A,A, and Diaz, I. Fuzzy inference map for condition monitoring with self-organizing maps, International Conference fuzzy logic and technology, PP.55-58, Leicester, UK
- Edin, O. and Geoff, R. (2004). Thruster fault diagnosis and accommodation for open-frame underwater vehicles. Proceedings Control Engineering Practice 12 1575–1598
- Fossen, T, I. (1995). Guidance and control of ocean vehicles. Chichester: Wiley.
- Fossen, T, I. (2002). Marine control systems. Trondheim, Norway: Marine Cybernetics
- Gianluca, A. (2006). Underwater Robots Motion and Force Control, Springer-Verlag Berlin Heidelberg, Second Edition. PP.78-93.
- Johansen, T, A. Fossen, T, I. and Tndel, P. (2004). Efficient optimal constrained control allocation via multiparametric programming. AIAA Journal of Guidance, Control and Dynamics
- Lingli, N. (2001). Fault-tolerant control of unmanned underwater vehicles, Ph.D. Dissertation, Blacksburg, Virginia.
- Omerdic, E. Roberts, G, N. and Ridao, P. Fault detection and accommodation for ROVs. Sixth IFAC conference on manoeuvring and control of marine craft, Girona, Spain.

- Podder, T, K. and Sarkar, N. (1999). Fault tolerant decomposition of thruster forces of an autonomousunderwater vehicle. Proceeding of the IEEE international conference on Robotics and Automation Detroit, MI, USA.
- Podder,T,K. Antonelli,G. and Sarkar,N. (2000). 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, pp.276-285.
- Podder,T,K. and Sarkar,N. (2001). Fault-tolerant control of an autonomous underwater vehicle under thruster redundancy, Robotics and Autonomous Systems, 34(1):39-52.
- Tondel, P. Johansen, T, A. and Bemporad, A. (2003b). Evaluation of piecewise affine control via binary search tree. Automatika, 39,743–749.
- 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.

- Yang, K, C. Yuh, J. and Choi, S, K. (1998). Experimental study of fault-tolerant system design for underwater robots. Proceeding of the 1998 IEEE international conference on Robotics and Automation,Leuven, Belgium.
- Yang, K, C. and Choi, S, K., (1999). Fault-tolerant system design of an autonomous underwater vehicle—ODIN: an experimental study, International Journal of Systems Science, 30(9):1011-1019.
- Zhu, D, Q. and Kong, M. (2006). A fuzzy CMAC neural network model based on credit assignment, International Journal of Information Technology, 12(6):1-8.
- Zhu, D, Q. and Kong, M. (2007). Adaptive Fault-tolerant Control of Nonlinear System: An Improved CMAC Based Fault Learning Approach. International Journal of Control,80(10):1576-1594.