

A Novel Tracking Controller for Autonomous Underwater Vehicles with Thruster Fault Accommodation

Bing Sun¹, Daqi Zhu¹ and Simon X. Yang²

¹(Laboratory of Underwater Vehicles and Intelligent Systems, Shanghai Maritime University, Shanghai, 201306, China)

²(The Advanced Robotics and Intelligent Systems Laboratory, School of Engineering, University of Guelph, Guelph, ON, N1G2W1, Canada)
(E-mail: zdq367@alipay.com)

In this paper, for the over-actuated Autonomous Underwater Vehicle (AUV) system, a novel tracking controller with thruster fault accommodation is proposed. Firstly, a cascaded control method is proposed for AUV robust tracking control. Then, we deal with the tracking control problem when one or more thrusters are completely or partly malfunctioning. Different control strategies are used to reallocate the thruster forces. For the cases that thrusters are partly malfunctioning, a weighted pseudo-inverse is firstly used to generate the normalised thruster forces. When the normalised thruster forces are out of maximum limits, the Quantum-behaviour Particle Swarm Optimisation (QPSO) is used for the restricted usage of the faulty thruster and to find the solution of the control reallocation problem within the limits. Compared with the weighted pseudo-inverse method, the QPSO algorithm does not need truncation or scaling to ensure the feasibility of the solution due to its particle search in the feasible solution space. The proposed controller is implemented in order to evaluate its performance in different faulty situations and its efficiency is demonstrated through simulation results.

KEYWORDS

1. Thruster fault accommodation.
2. Tracking control.
3. Weighted pseudo-inverse method.
4. Quantum-behaved Particle Swarm Optimisation.
5. Autonomous Underwater Vehicles.

Submitted: 25 June 2014. Accepted: 6 October 2015. First published online: 14 December 2015.

1. INTRODUCTION. In recent years, the use of Autonomous Underwater Vehicles (AUV) has rapidly increased since such a vehicle can be operated in the deeper and riskier areas divers cannot reach. In order to complete their mission in the complex underwater environment, AUVs have to face a lot of challenges where two things are extremely important: the robust character of the controller itself and safety consideration under thruster fault conditions.

To deal with the robust control problem, several different approaches have been applied such as adaptive control techniques (Ge and Li, 2014; Woods et al., 2012;

Isermann, 2011), sliding-mode control (Sankaranarayanan and Mahindrakar, 2009; Soylu et al., 2008), backstepping control algorithms (Lapierre and Jouvencel, 2008; Sui et al., 2014), fuzzy-logic control (Das Sharma et al., 2012; Chen et al., 2013; Liu et al., 2014) and neural network methods (Zheng and Jun, 2012; Pan et al., 2013).

The sliding-mode method has the outstanding characteristic of including insensitivity to parameter variations, and good rejection of disturbances. These robust characteristics make it quite suitable for robust AUV tracking control. However, one major drawback of the sliding-mode approach is the high frequency of control action (chattering). The backstepping control algorithm is the most commonly used approach for mobile robot tracking control and has been adopted in AUV control systems. The design idea is quite simple and well understood. However, the velocity control law is directly related to the state errors, so large velocities will be generated in a big initial error condition and sharp speed jumps occur during sudden tracking errors.

To resolve the impractical speed jump problem resulting from the backstepping technique, fuzzy control methods and neural network control algorithms are proposed. Fuzzy rule-based tracking control approaches can solve the problem of large initial vehicle velocities. However, the difficult point is the fuzzy rule formulation, which is usually obtained by trial and error-based human knowledge. Neural networks have aptitude for dealing with non-linear problems, and they are envisaged to be beneficial when used on AUVs. Anyhow, the existing neural network-based AUV tracking control algorithms require either online learning or offline training procedures that could be computationally complicated.

The bio-inspired model has been proposed for path planning and tracking control of mobile robots (Luo and Yang, 2008; Yang et al., 2012). In the work of Zhu et al. (2012), a control method with a bio-inspired model has been applied to the AUV path-following problem. Inspired by the smooth and bounded character of the bio-inspired model, a cascaded control system based on a virtual velocity controller and sliding-mode dynamic controller with bio-inspired model is presented for AUV trajectory tracking control.

Based on the above design, the controller can deal with the tracking control problem well, but there is little discussion about the AUV tracking control with thruster faults. In fact, as an AUV is expected to work in unstructured and complicated environments, it must be liable to faults or failures during underwater missions. Since a thruster is one of the most common and important fault sources (Zhu et al., 2008; Zhang and Jiang, 2008), once a fault occurs, the AUV cannot accomplish its mission and may even be lost. Due to safety considerations, AUVs are usually designed as an over-actuated system with more thrusters than the desired degree of freedom. If the fault condition is not serious enough, the objective of thruster fault-tolerant control is to perform an appropriate reconfiguration among the functioning thrusters. In such a way, an AUV can still follow the desired task-space trajectories. It is important to mention that in this work we are only concerned about the fault-accommodating allocation and assume that fault information is already known by some fault diagnosis method.

The algorithm of General Inverse (GI) is a common method for the solution of the control reallocation problem (Omerdic and Roberts, 2004; Benosman and Kai-Yew, 2009). However, in real-life implementations, the thruster control constraints must be taken into account. The GI approach cannot handle the constrained control problem since the solution of this approach is only based on the attainable command set. To handle such cases where attainable control inputs cannot be

allocated, T-approximation (Truncation) and S-approximation (Scaling) methods were proposed (Zhu et al., 2011). The solution obtained by the above approximation methods can be contained inside the entire attainable command set, but the magnitude and direction errors caused by approximation still exist and the AUV cannot follow the desired trajectory completely.

For the over-actuated control system, the essence of the system is a constrained optimisation problem and a survey on this problem has been done by Fossen and Jojansen (2006). Zhu et al. (2011) and Liu et al. (2009) use the Quantum-behaved Particle Swarm Optimisation (QPSO) method to deal with the fault tolerant control of the 7000m manned submarine and URIS AUV model respectively while the fitness value is based on direction error and magnitude error. Inspired by the above design and in order to do a precise reallocation of the fault thrusters, the QPSO method is applied to the control reallocation problem and a thruster Fault Accommodation (FA) method is proposed for the *Odin* AUV. The maximum norm is selected as the optimisation criterion. Then the thruster fault accommodation method is incorporated with the tracking control problem under thruster fault conditions. The reason for choosing the QPSO approach is the flexibility for online reconfiguration, as for example a change in weight matrix may require the explicit solutions to be recalculated. By this method, the control forces/moments can be reconstructed precisely.

In this paper, a novel control method with thruster FA is proposed for the robust tracking control problem. Different control strategies are used to reallocate the thruster forces when one or more thrusters are completely or partially malfunctioning. For the cases where thrusters are partially malfunctioning, a weighted pseudo-inverse is firstly used to generate the normalised thruster forces. When the normalised thruster forces are out of maximum limit, QPSO is used for the restricted usage of the faulty thruster and to find the solution of the control reallocation problem within the available limits.

This paper is organised as follows. The modelling of the AUV and the controller design is briefly introduced in Section 2. Section 3 describes the *Odin* control allocation system. Section 4 describes the thruster fault tolerant control and experimental simulations and results are discussed in Sections 4 and 5. Finally, concluding remarks are given in Section 6.

2. MODELLING AND CONTROL

2.1. Modelling in the horizontal plane. In the following, the general control allocation problem will be related with the motion in the horizontal plane. The coordinate systems considered in the horizontal plane are illustrated in Figure 1. The dynamic equation of AUV motion in the horizontal plane can be presented as (Fossen, 2011):

$$\begin{aligned} M\dot{q} + C(\dot{q})q + D(\dot{q})q + g(\eta) &= \tau \\ \dot{\eta} &= J(\eta)q \end{aligned} \quad (1)$$

where $\eta = [x \ y \ \psi]^T$ is the position and orientation state vector with respect to the inertial frame; $q = [u \ v \ r]^T$ is the AUV surge, sway and yaw velocity with respect to its body-fixed frame (see Figure 1) (Do et al., 2004); $\tau = [\tau_X \ \tau_Y \ \tau_N]^T$ is the control forces and moments of surge, sway and yaw motion. $M = \text{diag}(m_{xy}, m_{xy}, m_{\psi})$ is the inertia matrix including the added mass effects, $D = \text{diag}(d_u, d_v, d_r)$ is the hydrodynamic

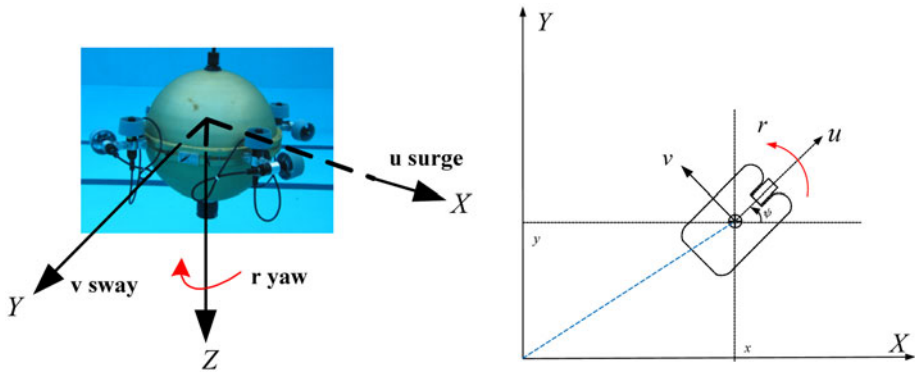


Figure 1. Mathematical model of AUV in the horizontal plane.

damping including added mass, $C = m_{xy} \begin{bmatrix} 0 & 0 & -v \\ 0 & 0 & u \\ v & -u & 0 \end{bmatrix}$ is the matrix of centrifugal and Coriolis terms, $J(\eta) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$ is the spatial transformation matrix between the inertial frame and the AUV's body-fixed frame.

2.2. *Filter-designed cascaded controller.* The basic control architecture of the system is illustrated in Figure 2. The design of the hybrid control strategy consists of two parts: (1) an outer loop virtual velocity controller using position and orientation state errors; (2) an inner loop sliding-mode controller using velocity state vector.

2.2.1. *Filter-designed velocity controller.* For AUV motion control in the horizontal plane, the desired state of a reference vehicle is described as $\eta_d = [x_d \ y_d \ \psi_d]^T$ and $q_d = [u_d \ v_d \ r_d]^T$. The actual state of the AUV is represented by $\eta = [x \ y \ \psi]^T$, $q = [u \ v \ r]^T$. The objective of the tracking controller is to make the AUV follow the known trajectory by controlling the forward and angular velocities. Thus the error $e = [e_x \ e_y \ e_\psi]^T$ between desired state and actual state converges to zero. $e = \eta_d - \eta = [e_x \ e_y \ e_\psi]^T$ is the tracking error in the inertial frame. The detailed description can be seen in Figure 3.

The auxiliary velocity controller based on the backstepping approach can be defined as

$$q_c = \begin{bmatrix} u_c \\ v_c \\ r_c \end{bmatrix} = \begin{bmatrix} k(e_x \cos \psi + e_y \sin \psi) + (u_d \cos e_\psi - v_d \sin e_\psi) \\ k(-e_x \sin \psi + e_y \cos \psi) + (u_d \sin e_\psi + v_d \cos e_\psi) \\ r_d + k_\psi e_\psi \end{bmatrix} \quad (2)$$

where k, k_ψ are constant coefficients.

Through deep analysis of Equation (2), the auxiliary speed is directly related to the tracking errors. As mentioned in Section 1, this typical backstepping method will undoubtedly produce sharp speed jumps. In order to resolve the speed jump and control constraint problem, a bio-inspired model is added in the controller to design the

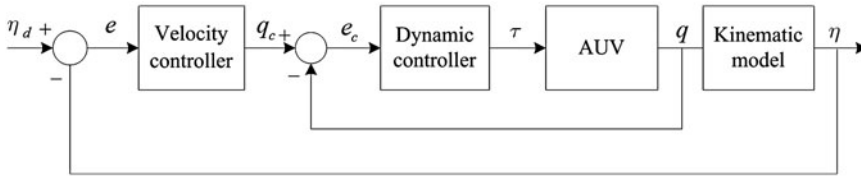


Figure 2. The scheme of the proposed AUV tracking control system (without thruster fault).

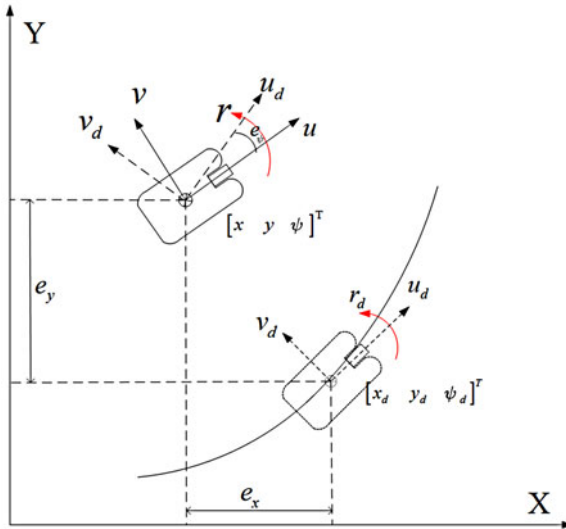


Figure 3. Tracking control formulation.

auxiliary velocity. To simplify the equation, the bio-inspired model can infer the following form (Yang et al., 2012)

$$\dot{V}_i = -AV_i + (B - V_i)f(e_i) - (D + V_i)g(e_i) \tag{3}$$

where $f(e_i) = \max(e_i, 0)$, $g(e_i) = \max(-e_i, 0)$, A, B, D are positive constants. The bio-inspired model is a continuous differential equation and can also be regarded as a low pass filter. The system's output V is guaranteed to stay in a region $[-D, B]$ for any excitatory and inhibitory inputs. In this paper, the tracking errors e are chosen as the inputs of the bio-inspired model, while the outputs V_i ($i = x, y, \psi$) will replace the tracking errors e .

The proposed filter designed velocity controller is given by:

$$\mathbf{q}_c = \begin{bmatrix} u_c \\ v_c \\ r_c \end{bmatrix} = \begin{bmatrix} k(V_x \cos \psi + V_y \sin \psi) + (u_d \cos e_\psi - v_d \sin e_\psi) \\ k(-V_x \sin \psi + V_y \cos \psi) + (u_d \sin e_\psi + v_d \cos e_\psi) \\ r_d + k_\psi V_\psi \end{bmatrix} \tag{4}$$

where k, k_ψ are the same parameters as in Equation (2). Due to the shunting characteristics of the bio-inspired model, the output of the bio-inspired model is bounded in a finite interval and smooth without any sharp jumps when inputs have sudden changes.

2.2.2. *Dynamic controller.* Sliding-mode control is applied to extend kinematic control to the dynamic control. It is a former work by the authors (Sun et al., 2014) and a brief description is given as follows.

As a rule, sliding-mode control can be divided into two parts. First, define a sliding manifold s . Second, find a control law to move toward the sliding manifold. The sliding manifold (or filter velocity error) can be chosen as (Slotine and Li, 1991):

$$s = \dot{e}_c + 2\Lambda e_c + \Lambda^2 \int e_c \quad (5)$$

where $e_c = \dot{q}_c - \dot{q}$ is the velocity error between the auxiliary velocity and the actual AUV velocity. As a derivation of Equation (5) the equivalent control law can be written as

$$\tau_{eq} = \hat{M}(\dot{q}_c + \frac{\ddot{e}_c}{2\Lambda} + \frac{\Lambda}{2} e_c) + \hat{C}q + \hat{D}q + \hat{g} \quad (6)$$

where \hat{M} , \hat{C} , \hat{D} , \hat{g} are estimated terms. To eliminate the chattering problem caused by the conventional discontinuous term, an adaptive term is added in the control law to replace the switching term

$$\tau_{ad} = \tilde{\tau}_{est} + (K + \frac{\hat{C}}{2\Lambda})s \quad (7)$$

$\tilde{\tau}_{est}$ is an adaptive term that estimates the lumped uncertainty vector $\tilde{\tau}$. The estimation of the lumped uncertainty vector is proposed to follow:

$$\dot{\tilde{\tau}}_{est} = \Gamma s \quad (8)$$

The total control law can be defined as

$$\tau = \tau_{eq} + \tau_{ad} = \tau_{eq} + \tilde{\tau}_{est} + (K + \frac{\hat{C}}{2\Lambda})s \quad (9)$$

3. CONTROL ALLOCATION OF ODIN. In this paper, our work mainly focuses on the filter-designed cascaded tracking control with fault thruster reallocation in the horizontal plane. In order to demonstrate the cascaded tracking control effectiveness, an example of *Odin* AUV (Do et al., 2004) tracking control is discussed. A brief sketch of the *Odin* vehicle's horizontal thruster distribution is shown in Figure 4. ODIN has four horizontal thrusters fixed in symmetrical layout, denoted as iHT , $i \in [1, 4]$, and four vertical thrusters. The thruster configuration of *Odin* enables direct control of surge, sway and yaw control in the horizontal plane. This construction puts redundancy into the system in case of thruster failure. Thruster fault tolerant control in the horizontal plane will be discussed in this paper.

The total vector of propulsion forces and moments in the horizontal plane can be written as:

$$\begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_N \end{bmatrix} = \begin{bmatrix} -\sin \alpha & \sin \alpha & \sin \alpha & -\sin \alpha \\ \sin \alpha & \sin \alpha & -\sin \alpha & -\sin \alpha \\ R_z & -R_z & -R_z & R_z \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \quad (10)$$

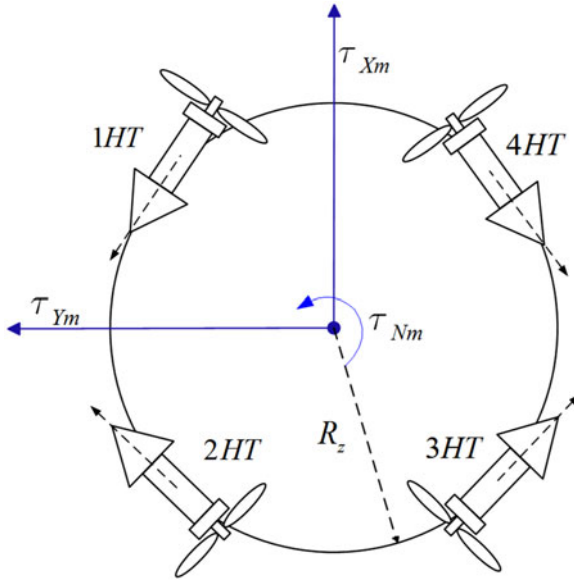


Figure 4. Thruster Distribution.

where $[\tau_X \ \tau_Y \ \tau_N]^T$ is the total forces/moments acting on the AUV centre of mass, $[T_1 \ T_2 \ T_3 \ T_4]^T$ is the forces of the four thrusters individually, $R_z = 0.4816$, $\alpha = 45^\circ$. By simplification, Equation (10) can be written as:

$$\begin{bmatrix} \tau_X \\ \tau_{Xm} \\ \tau_Y \\ \tau_{Ym} \\ \tau_N \\ \tau_{Nm} \end{bmatrix} = \begin{bmatrix} -\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & -\frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} \end{bmatrix} \begin{bmatrix} T_1/T_m \\ T_2/T_m \\ T_3/T_m \\ T_4/T_m \end{bmatrix} \tag{11}$$

where τ_{Xm} , τ_{Ym} , τ_{Nm} represents the maximum values of force and moment vectors for *Odin* AUV respectively, T_m represents the max force of an individual thruster.

$$\begin{bmatrix} \bar{\tau}_X \\ \bar{\tau}_Y \\ \bar{\tau}_N \end{bmatrix} = \bar{B} \begin{bmatrix} \bar{T}_1 \\ \bar{T}_2 \\ \bar{T}_3 \\ \bar{T}_4 \end{bmatrix} \Leftrightarrow \bar{\tau} = \bar{B}\bar{T} \tag{12}$$

where $\bar{B} = \begin{bmatrix} -\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & -\frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} \end{bmatrix}$, $-1 \leq \bar{\tau}_j \leq 1, j = X, Y, N, \quad -1 \leq \bar{T}_i \leq 1,$
 $i = 1, 2, 3, 4.$

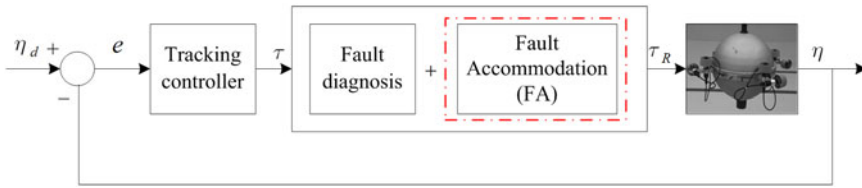


Figure 5. A general tracking control system with thruster fault.

4. THRUSTER FAULT TOLERANT CONTROL. In this section, the tracking control problem is extended to the thruster fault condition. When a thruster fault happens, thruster accommodation among the functioning thrusters is adopted in the system. A general sketch of tracking control with thruster fault is given in Figure 5. Thruster fault accommodation of an AUV such as *Odin* has been discussed in Sarkar et al. (2002) and Andonian et al. (2010). Unlike the former discussion only about a totally broken thruster, in this section, the thruster fault tolerant control strategy will be divided into two main categories: thruster completely malfunctioning and thruster partially malfunctioning. In this work, the fault condition is assumed to be known by some fault diagnosis method. The main work is to propose a fault accommodation strategy to generate the demanded control signal among the functioning thrusters. The block diagram of thruster fault accommodation control for the specific *Odin* AUV can be found in Figure 6 and it should be noted that this control strategy can be applied to different kind of AUVs in a similar way.

4.1. *Thrusters completely malfunctioning.* The thruster configuration matrix \bar{B} with no fault is given in Equation (12). When only one thruster (the i th thruster) is faulty, delete the corresponding column of thruster configuration matrix \bar{B} , and then \bar{B} turns into a 3×3 matrix \bar{B}_i . This leads to a unique 3×1 solution vector \bar{T} (the i th broken thruster output \bar{T}_i is deleted from the vector). The thruster solution can be given by simple inverse

$$\bar{T} = \bar{B}_i^{-1} \bar{\tau} \quad (13)$$

If the solution of Equation (13) is out of the limitation boundary $-1 \leq \bar{T} \leq 1$, the T-approximation or S-approximation method is used to ensure the limitation and optimisation algorithms cannot be applied in this case.

4.2. *Thruster partly malfunctioning.*

4.2.1. *Hybrid thruster reconfiguration strategy.* As can be seen from the block diagram of thruster fault tolerant control in Figure 6, a hybrid thruster reconfiguration strategy is proposed by using the weighted pseudo-inverse and QPSO. The hybrid thruster control reconfiguration approach can be described as follows:

First, the desired control signal $\bar{\tau}_d$ is calculated by the weighted pseudo-inverse to generate the thruster force \bar{T} . If all of the normalised thruster output is in the range of saturation point $-1 < \bar{T}_i < 1$, the QPSO method is not needed in this case. If any of the normalised thruster input is out of the saturation range $-1 < \bar{T}_i < 1$, then QPSO method is applied so that the demanded controller output can be reallocated among the functioning thrusters.

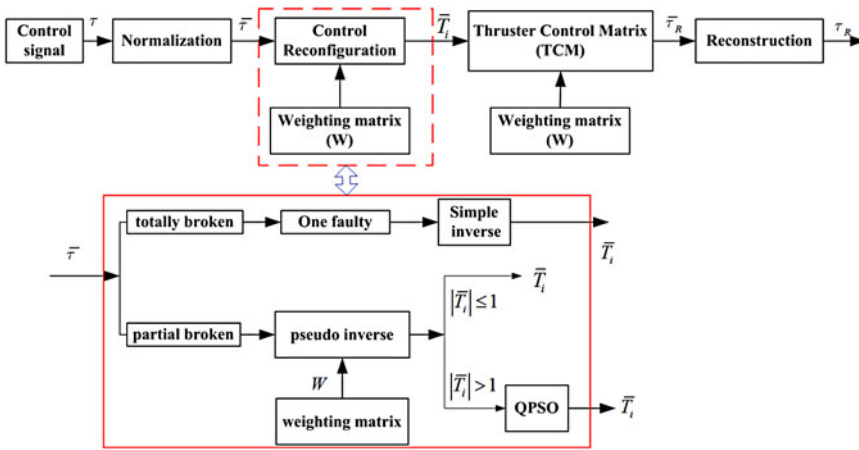


Figure 6. The block diagram of thruster fault accommodation control.

4.2.2. *Weighted pseudo-inverse method.* In the former section, the case of one thruster totally broken is considered. But a large proportion of the cases will involve a thruster in a partially malfunctioning condition. Here we assume that the fault diagnosis processes have already been completed (Zhu and Sun, 2013). Accordingly, this paper focuses on the thruster fault accommodation and discusses the tracking control for continuing the mission.

In order to account for the faulty thrusters, a diagonal weighting matrix can be used for the later thruster accommodation. The diagonal weighting matrix can be defined as

$$W = \text{diag}(w_1 \ w_2 \ w_3 \ w_4) \in R^{4 \times 4}$$

$$w_i = \begin{cases} 0 < w_i \leq 1, & \text{if the } i\text{th thruster is in partial failure, } i = 1 \sim 4 \\ 1, & \text{if the } i\text{th thruster is not in failure} \end{cases} \quad (14)$$

where the weighting coefficient w_i represents the fault degree of the thruster. The bigger the w_i , the smaller the fault and vice versa. “1” and “0” in the weighting matrix represent the upper and lower saturation limits of faulty thrusters to meet the available thrust capacity. Generally speaking, propeller thrust and rotational speed is a complex nonlinear relationship. Here, for simple consideration, it can be regarded as a linear relation, so the weight coefficient can be simply described as

$$w_i = 1 - \frac{n_i}{n_{\max}} \quad (15)$$

where n_i is the actual rotational speed and n_{\max} is the ideal rotational speed.

The block diagram of the control reconfiguration is shown in Figure 5. It is clear that the control allocation matrix can be rewritten as:

$$\bar{\tau} = \bar{B}W\bar{T} \quad (16)$$

If the thruster is partly malfunctioning with the given weighting matrix in Equation (14), the horizontal thruster forces for the desired horizontal motions can be obtained by

weighted pseudo-inverse:

$$\bar{T} = W^{-1} \bar{B}^T (\bar{B} W^{-1} \bar{B}^T)^{-1} \bar{\tau} \quad (17)$$

The weighted pseudo-inverse method is a commonly used control allocation method to deal with the thruster fault accommodation problem of AUV systems. Two approximation methods (truncation or scaling) are used to ensure that the normalised thruster forces are able to satisfy the constraints ($-1 \leq \bar{T} \leq 1$). However, by truncation or scaling, the reconstruction control forces may not equal the desired forces $\tau_R \neq \tau$. Generally, the number of thrusters in the *Odin* AUV is more than the minimum required to produce the desired motion, so the thruster solution is not unique. In order to extract a most appropriate solution, QPSO with constraints is used to find the optimum control vector in this paper.

4.2.3. *Quantum-behaved Particle Swarm Optimisation (QPSO)*. In the QPSO model, the state of a particle is depicted by wave function $\psi_{(x,t)}$ instead of position and velocity. The dynamic behaviour of the particle is widely divergent from the particle in traditional PSO systems since the exact values of x and V cannot be determined simultaneously. It can only learn the probability of the particle's appearance in position x from the probability density function $|\psi_{(x,t)}|^2$. By using this function, the potential field where the particle lies can be determined. The particles move according to the following iterative equation:

$$\begin{aligned} X_{(t+1)} &= P_i - \beta^*(mbest - X_t) * \ln(1/u) \quad \text{if } k \geq 0.5 \\ X_{(t+1)} &= P_i + \beta^*(mbest - X_t) * \ln(1/u) \quad \text{if } k < 0.5 \end{aligned} \quad (18)$$

where

$$P_i = \varphi * pbest_i + (1 - \varphi) * gbest_i \quad (19)$$

$$mbest = \frac{1}{N} \sum_{i=1}^N pbest_i \quad (20)$$

$mbest$ is the mean best position defined as the mean of all the best positions of the population, k , u and φ are random numbers distributed uniformly on $[0, 1]$ respectively. Considering that the iteration number and population size are common parameters in every evolutionary algorithm, the contraction-expansion coefficient β is the only parameter in the QPSO algorithm. It can be tuned to control the convergence speed of QPSO method.

The objective function is defined by the l_∞ norm of the thruster force. The l_∞ norm of the thruster force manifold $\bar{T} = [\bar{T}_1 \quad \bar{T}_2 \quad \bar{T}_3 \quad \bar{T}_4]^T$ is defined as

$$\|\bar{T}\|_\infty = \max\{|\bar{T}_1| \quad |\bar{T}_2| \quad |\bar{T}_3| \quad |\bar{T}_4|\} \quad (21)$$

For equation $\bar{\tau} = \bar{B} W \bar{T}$, it can be written as

$$\begin{aligned} \bar{\tau}_X &= -\frac{1}{4} w_1 \bar{T}_1 + \frac{1}{4} w_2 \bar{T}_2 + \frac{1}{4} w_3 \bar{T}_3 - \frac{1}{4} w_4 \bar{T}_4 \\ \bar{\tau}_Y &= \frac{1}{4} w_1 \bar{T}_1 + \frac{1}{4} w_2 \bar{T}_2 - \frac{1}{4} w_3 \bar{T}_3 - \frac{1}{4} w_4 \bar{T}_4 \\ \bar{\tau}_N &= \frac{1}{4} w_1 \bar{T}_1 - \frac{1}{4} w_2 \bar{T}_2 + \frac{1}{4} w_3 \bar{T}_3 - \frac{1}{4} w_4 \bar{T}_4 \end{aligned} \quad (22)$$

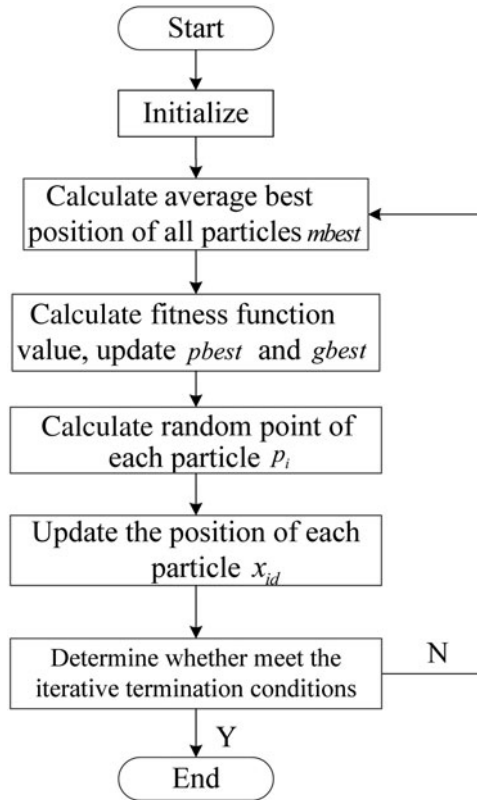


Figure 7. The flow chart of QPSO.

One inverse function can be given as

$$\begin{aligned}
 \bar{T}_1 &= \bar{T}_1 \\
 \bar{T}_2 &= (w_1 \bar{T}_1 - 2(\bar{\tau}_N - \bar{\tau}_X))/w_2 \\
 \bar{T}_3 &= (w_1 \bar{T}_1 - 2(\bar{\tau}_Y - \bar{\tau}_X))/w_3 \\
 \bar{T}_4 &= (w_1 \bar{T}_1 - 2(\bar{\tau}_Y + \bar{\tau}_N))/w_4
 \end{aligned}
 \tag{23}$$

where $w_1, w_3, w_4 \neq 0$. Then based on the new Equation (23), the optimisation problem can be simplified into one single parameter tuning under the objective criterion Equation (21). This solution can reduce the computation cost and select the most suitable control force in the solution space. While in the case of more thrusters, multi parameter tuning is favourable, this is out of the scope of this paper.

After thruster normalisation, $[-1, 1]$ are the thruster saturation limits, so the constrained optimisation problem becomes to acquire the smallest l_∞ norm thrust value as the feasible solution. The formula's expression can be given as follows:

$$\text{minimise } \|\bar{T}\|_\infty, \text{ subject to } \bar{\tau} = \bar{B}W\bar{T}, -1 \leq \bar{T}_i \leq 1, i = 1 \sim 4.$$

The flow chart of QPSO is shown in Figure 7.

Table 1. The results of first thruster fault case.

		Weight pseudo-inverse				QPSO			
		T-approximation (truncation)		S-approximation (scaling)					
\bar{T}	[0.24 0.4 0.24 -1]	[0.2143	0.3571	0.2143	-1]	[0.8267	0.6933	0.5333	-0.8267]
$\bar{\tau}$	[0.38 0.32 0.24]	[0.3661	0.3125	0.2411]	[0.41	0.35	0.27]		

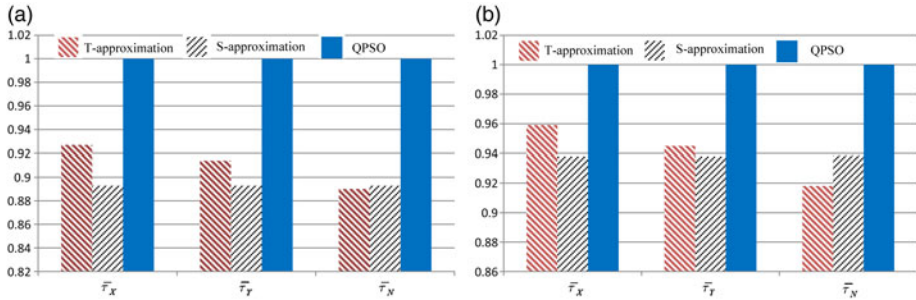


Figure 8. Comparison of different types of fault case.

4.3. *Simulation study.* In order to show the superior performance when QPSO is applied to the hybrid thruster reconfiguration strategy, two simple comparison studies are conducted in this section. This will show the efficiency of this control allocation strategy.

4.3.1. *The One Thruster Fault Case.* Assume that the first thruster (1HT thruster) is detected as faulty and its corresponding weight value of fault state w_1 is 1/2, so $W = \text{diag}(1/2 \ 1 \ 1 \ 1)$.

Let the given motion control forces/moments $\bar{\tau} = [0.41 \ 0.35 \ 0.27]$ for thruster configuration of *Odin* AUV. The weighted pseudo-inverse solution $\bar{T} = [0.24 \ 0.4 \ 0.24 \ -1.12]$ is unfeasible because $\bar{T}_4 > 1$. Then T-approximation is given by $\bar{T} = [0.24 \ 0.4 \ 0.24 \ -1]$, according to Equation (16), the reconfiguration forces are $\bar{\tau}_R = [0.38 \ 0.32 \ 0.24]$. The S-approximation is given by $\bar{T} = [0.2143 \ 0.3571 \ 0.2143 \ -1]$, then $\bar{\tau}_R = [0.3661 \ 0.3125 \ 0.2411]$. In contrast, the solution obtained by the QPSO algorithm $\bar{T} = 0.8267 \ 0.6933 \ 0.5333 \ -0.8267$ is feasible without any approximation because \bar{T} is limited in $[-1, 1]$ during the process of searching for a solution in the QPSO algorithm and the corresponding $\bar{\tau}_R = \bar{B}W\bar{T} = [0.41 \ 0.35 \ 0.27] = \bar{\tau}$.

From Table 1, it can be seen that the control vector obtained by the QPSO algorithm can reallocate the thruster forces reasonably compared with the method of weighted pseudo-inverse and can precisely meet the desired control forces/moments $\bar{\tau} = [0.41 \ 0.35 \ 0.27]$ as the original state. By normalisation of the reconfiguration forces and moments in Table 1, the comparison result is illustrated in Figure 8(a) which clearly shows the reconfiguration effect.

4.3.2. *The Two Thruster Fault Case.* Assume that the second and third thrusters (2HT and 3HT thruster) are detected as faulty, and the corresponding weight value of

Table 2. The results of second and third thruster fault case.

	Weight pseudo-inverse								
	T-approximation (truncation)				S-approximation (scaling)				QPSO
\bar{T}	[-0.0657	0.4457	0.2686	-1]	[-0.0617	0.4182	0.2520	-1]	[0.2 0.8 0.8 -0.8]
$\bar{\tau}$	[0.3836	0.2836	0.1836]		[0.3753	0.2815	0.1877]		[0.4 0.3 0.2]

fault state $w_2 = 3/4, w_3 = 1/2$ and weight matrix $W = diag(1 \ 3/4 \ 1/2 \ 1)$. Let the original state $\bar{\tau} = [0.4 \ 0.3 \ 0.2]$. The experimental results are shown in Table 2 and are mostly the same result as the one fault case. By normalisation of the reconfiguration forces and moments in Table 2, the comparison result is illustrated in Figure 8(b) which clearly shows the reconfiguration effect.

The solutions obtained by different methods could perform the appropriate configuration, but the solutions obtained by weighted pseudo-inverse still have some error in magnitude, while the QPSO algorithm can preserve the original magnitude within thruster limitations. It should be noted that l_∞ norm thrust allocation is not energy minimisation but a solution to minimise all the thruster forces in the solution space so that the AUV can have more manoeuvring ability.

5. SIMULATION AND DISCUSSION. In this section, simulation studies are performed to demonstrate the effectiveness of the proposed tracking control method with thruster allocation approaches. The AUV system considered in the simulations is the *Odin* model discussed in Section 3.

5.1. *Circular tracking.* In this study, the filter-designed method with thruster faults is applied to the trajectory tracking. Assume that the first thruster is totally broken in the first 10 s, then in the 10 s–30 s assume that second and third thrusters (2HT and 3HT thruster) are detected as partially faulty, and the corresponding weight value of fault state $w_2 = 3/4, w_3 = 1/2$. The weight matrix can be written as:

$$W = \begin{cases} diag(0 \ 1 \ 1 \ 1) & 0 \leq t \leq 10 \\ diag(1 \ 3/4 \ 1/2 \ 1) & 10 < t \leq 30 \end{cases}$$

The AUV starts at posture $(-0.5, -2, 0)$, while the desired initial AUV posture is $(0, -1, 0)$. Thus the initial posture error is $(-0.5, -1, 0)$. Time varies from 0 to 30 s. The desired state of AUV is $x_d(t) = \sin(0.5t), y_d(t) = -\cos(0.5t), \psi_d(t) = 0.5t$.

In order to show the efficiency of the fault accommodation method, the filter-designed tracking method without fault accommodation is conducted as a comparison study. Figures 9–11 show the simulation results of the circle tracking. The red solid lines indicate the filter-designed tracking method with fault accommodation results, and the blue solid lines are the filter-designed method without fault accommodation. Figure 9 gives the tracking control results of the filter-designed method with and without fault accommodation on the same condition. Figure 10 gives the tracking errors and the normalised thruster control force \bar{T} of trajectory tracking is shown in Figure 11.

In the first 10 s, the first thruster is assumed to be totally broken, and then the thruster reallocation method is given in Equation (13). The T-approximation method is used to ensure the limitation if the solution of Equation (13) is out of the

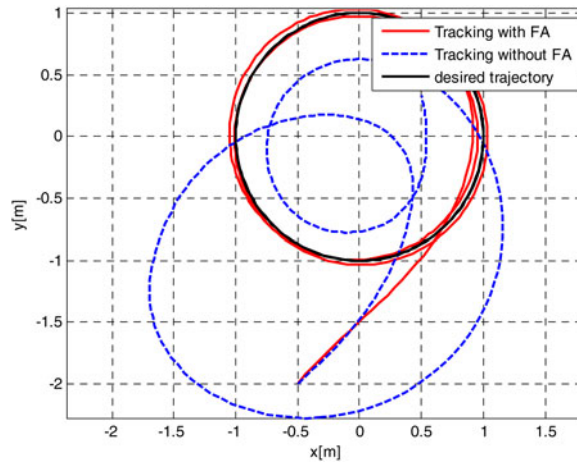


Figure 9. The tracking result using filter-designed method with and without fault accommodation.

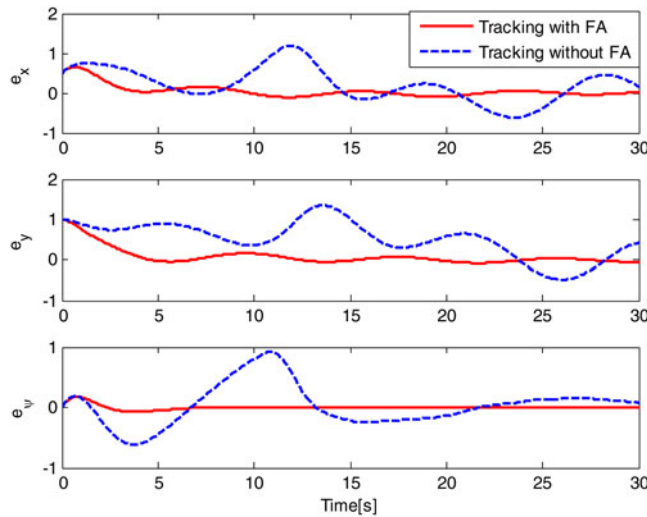


Figure 10. The tracking errors using filter-designed method with and without fault accommodation.

limitation boundary $-1 \leq \bar{T} \leq 1$. In 10 s–30 s, the second and third thrusters (2HT and 3HT thruster) are detected as partially faulty. Then the weighted pseudo-inverse method given in Equation (17) is used to generate the thruster control signal. If any of the thruster control signal \bar{T}_i is out of the limitation range $[-1, 1]$, the QPSO method is used to reconfigure the thruster control forces.

From Figures 9 and 10, it can be clearly seen that the filter-designed tracking method with fault accommodation can quickly achieve the desired trajectory while the tracking method without fault accommodation cannot accomplish the tracking mission since the reconfigured control forces/moments cannot meet the desired

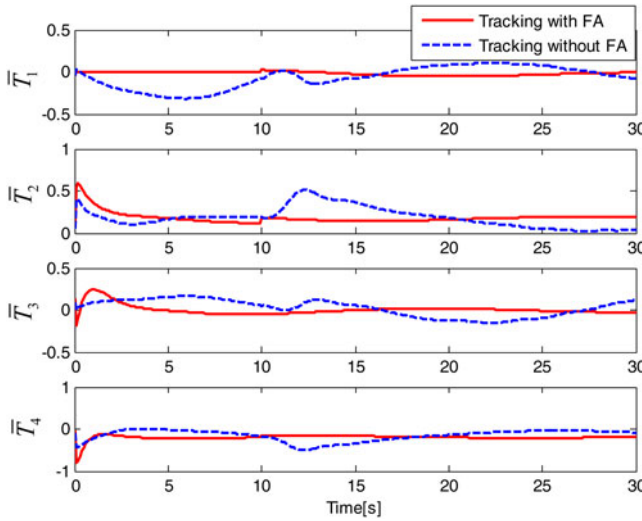


Figure 11. The normalised thruster forces using filter designed method with and without fault accommodation.

control forces/moments $\bar{\tau} \neq \bar{\tau}_d$. The fault accommodation method can achieve perfect tracking after the first 5 s. After 10 s, the fault thruster situation changes, therefore it needs an adjustment process while the filter-designed tracking method with fault accommodation can still achieve smooth tracking control. Where there is no fault accommodation method, since no fault tolerant control strategy is added, the tracking performance is quite poor so that it cannot achieve the assigned tracking tasks. The change of thruster fault case can also be reflected in the normalised thruster control forces in Figure 11. At time 10 s, the fault accommodation method can reallocate the control forces rapidly and achieve precise tracking while not having a fault accommodation method does not achieve the desired control forces.

5.2. *Sinusoid tracking.* Circular tracking is a traditional research scenario where the desired trajectory is generated with constant velocity, but sometimes the desired trajectory is generated with variable velocities and it will be more difficult to solve the tracking control problem. In order to show the superiority of the fault accommodation method, in this section, the sinusoid tracking will be studied and the parameter setting is the same as the former section. The AUV starts at posture $(-0.5, -2, 0)$, while the desired initial robot posture is $(0, 0, 1.3734)$. Thus the initial posture error is $(0.5, 2, 1.3734)$. Time varies from 0 to 60 s. The desired state of the AUV is $x_d(t) = 0.2t$, $y_d(t) = 2\sin(0.5t)$, $\psi_d(t) = a \tan 2(\frac{\dot{y}_d}{x_d})$. The fault tolerant control strategy is the same as Section 4. The weight matrix can be written as:

$$W = \begin{cases} \text{diag}(0 & 1 & 1 & 1) & 0 \leq t \leq 10 \\ \text{diag}(1 & 3/4 & 1/2 & 1) & 10 < t \leq 30 \end{cases}$$

Figures 12–14 show the simulation results of the sinusoid tracking. The red solid lines indicate the filter designed tracking method with fault accommodation results, and the

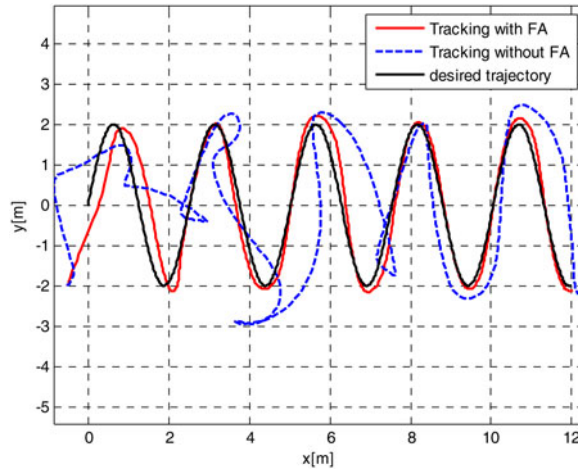


Figure 12. The tracking result using filter-designed method with and without fault accommodation.

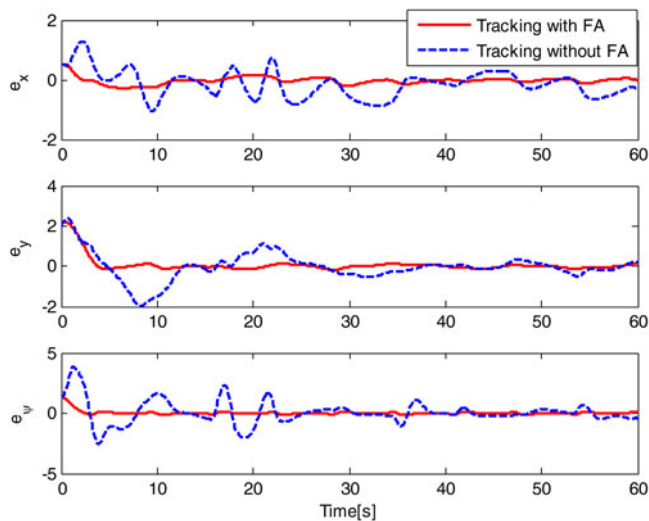


Figure 13. The tracking errors using filter-designed method with and without fault accommodation.

blue solid lines are the filter-designed method without fault accommodation. Figure 12 gives the tracking control results of filter-designed method with and without fault accommodation on the same condition. Figure 13 shows the tracking errors of the filter-designed method with and without fault accommodation. The normalised thruster control force of trajectory tracking is shown in Figure 14.

From Figures 13 and 14 it can be clearly seen that the proposed method with fault accommodation can quickly achieve the desired trajectory while the tracking method

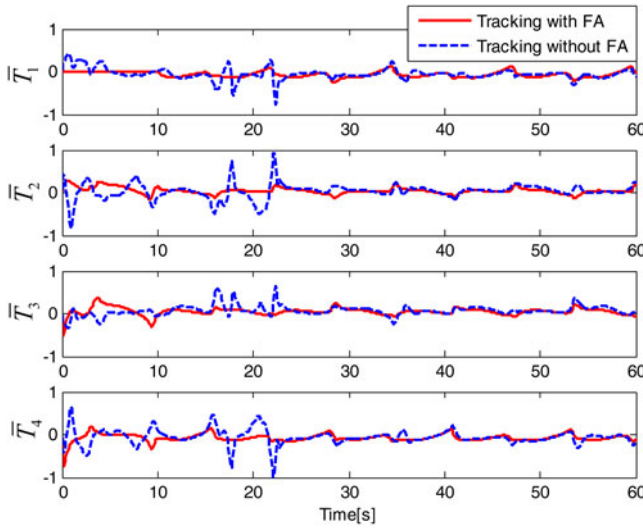


Figure 14. Normalised thruster control force using filter-designed method with and without fault accommodation.

without fault accommodation cannot reach the trajectory and move along it in a limited time. For the fault accommodation method, it can achieve quite good tracking results after the first 10 s. Due to the varying velocities, there still exist small tracking errors, but they remain within an acceptable range. When the fault thruster situation changes after 10 s, the filter-designed tracking method with fault accommodation can still have a smooth and continuous control without performance loss. The change of thruster fault case can also be reflected in the normalised thruster forces. After 10 s, fault accommodation method can reallocate the control forces rapidly and achieve precise tracking while the method with no fault accommodation has to catch the desired control forces smoothly.

5.3. Comparison with other methods. From the former simulation results, it can be concluded that the proposed method is an efficient tool for fault tolerant tracking control. In this part, in order to show the reason to choose the cascaded control method for tracking control, a comparison study between the proposed method and some benchmark methods are given. Here, an adaptive method and a neural network method are selected since they are commonly used in robotic control and a simple introduction is given as follows:

5.3.1. Adaptive method. A classical adaptive algorithm (hereinafter referred to as adaptive method) first proposed by Slotine and Li (1991) was used as a comparison study. The control law is given as:

$$\tau = \Phi(\dot{v}_r, v_r, v, \eta)\hat{\theta} - \mathbf{J}^T(\boldsymbol{\eta})K_d\mathbf{s} \tag{24}$$

The adaptive law of $\hat{\theta}$ can be chosen as:

$$\dot{\hat{\theta}} = -\Gamma\Phi^T(\dot{v}_r, v_r, v, \eta)\mathbf{J}^{-1}(\boldsymbol{\eta})\mathbf{s} \tag{25}$$

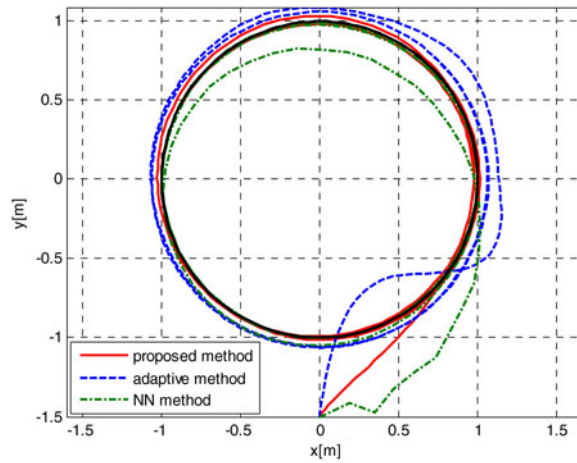


Figure 15. Comparison study with other control methods.

where $\hat{\theta}$ is the estimated parameter vector and K_d , Γ are positive constants. Detailed description of this method can be found in Slotine and Li (1991).

5.3.2. *Neural network (NN) method.* Neural network control for robotic system is a popular design and here a typical neural network design (Radial Basis Function) was used as a comparison study. The control law is given as:

$$\tau_\eta = \hat{f}(h) + K_v s - \alpha \quad (26)$$

where the neural network function estimate in the controller as $\hat{f}(h) = \hat{W}^T \beta(h)$ and a detailed description of this method be found in Jagannathan and Galan (2003).

The AUV starts at posture $(0, -1.5, 0)$, while the desired initial robot posture is $(0, -1, 0)$. Thus the initial posture error is $(0, 0.5, 0)$. Time varies from 0 to 30 s. The desired state of AUV is $x_d(t) = \sin(0.5t)$, $y_d(t) = -\cos(0.5t)$, $\psi_d(t) = 0.5t$. The controller parameter setting for the three methods are given as follows:

1. The proposed method: $K = 20$, $k = 2$, $k_\psi = 2$, $A = 2$, $B = 1$, $D = 1$
2. The adaptive method: $K_d = 20$, $\Gamma = 1$
3. The neural network method: $K_p = 9$, $K_v = 6$, the NN was not trained offline and the weights were initialised at zero.

It can be concluded that in order to give a satisfactory tracking result, the parameters cannot be too small, or the convergence rate will be too slow. The parameters cannot be too large, or control inputs will be big enough to exceed the thruster limits. The parameter values are selected experimentally and have the same order of magnitude in a general way.

Figure 15 shows the final tracking control results under the different methods: the proposed method, adaptive method and NN method. From the simulation results, it is clear to see that the proposed method has superior performance over the other two methods. Since the adaptive method and NN design have both updated a process, it will cause a computation burden and the weight tuning in the very beginning is a major problem in causing slow convergence.

6. **CONCLUSION.** In this paper, a filter-designed cascaded controller for the *Odin* Autonomous Underwater Vehicle (AUV) with thruster fault tolerant control is proposed. Firstly, a filter-designed cascaded tracking control is proposed to deal with robust tracking control. Secondly, a different control strategy for various thruster fault cases is presented while the major contribution is a mixed control method of weighted pseudo-inverse and quantum-behaved particle swarm optimisation for fault thruster reallocation. Simulation experiments were implemented to show the efficiency of the filter-designed cascaded tracking control with thruster fault accommodation. Future work will involve experimentally validating the proposed controller-thrust allocation scheme on the existing *Odin* AUV system including computer programming.

ACKNOWLEDGEMENT

This project is supported by the National Natural Science Foundation of China (51279098, 51575336, 61503239) and the Creative Activity Plan for Science and Technology Commission of Shanghai (14JC1402800).

REFERENCES

- Andonian, M., Cazzaro, D. and Invernizzi, L. (2010). Geometric control for autonomous underwater vehicles: overcoming a thruster failure. *IEEE Conference on Decision and Control (CDC)*, IEEE, 7051–7056.
- Benosman, M. and Kai-Yew, L. (2009). Online references reshaping and control reallocation for nonlinear fault tolerant control. *IEEE Transactions on Control Systems Technology*, **17**, 366–379.
- Chen, X. and Tan, W.W. (2013). Tracking control of surface vessels via fault-tolerant adaptive backstepping interval type-2 fuzzy control, *Ocean Engineering*, **70**, 97–109.
- Das Sharma, K., Chatterjee, A and Rakshit, A. (2012). A PSO–Lyapunov Hybrid Stable Adaptive Fuzzy Tracking Control Approach for Vision-Based Robot Navigation, *IEEE Transactions on Instrumentation and Measurement*, **61**, 1908–1914.
- Do, K.D., Jiang, Z.P., Pan, J. and Nijmeijer, H. (2004). A global output-feedback controller for stabilization and tracking of underactuated ODIN: A spherical underwater vehicle. *Automatica*, **40**, 117–124.
- Fossen, T.I. (2011). Handbook of marine craft hydrodynamics and motion control. John Wiley & Sons.
- Fossen, T.I. and Jojansen, T.A. (2006). A survey of control allocation methods for ship and underwater vehicles. *14th IEEE Med. Conf. on Control and Automation*, Ancona, 1–6.
- Ge, S.S. and Li, Z. (2014). Robust adaptive control for a class of MIMO Nonlinear Systems by State and Output Feedback, *IEEE Transactions on Automatic Control*, **59**, 1624–1629.
- Isermann, R. (2011). Perspectives of automatic control. *Control Engineering Practice*, **19**, 1399–1407.
- Jagannathan, S. and Galan, G. (2003). One-layer neural-network controller with preprocessed inputs for autonomous underwater vehicles. *IEEE Transactions on Vehicular Technology*, **2003**, **52**, 1342–1355.
- Lapierre, L. and Jouvencel, B. (2008). Robust nonlinear path-following control of an AUV. *IEEE Journal of Oceanic Engineering*, **33**, 89–102.
- Liu, J., Wu, Q. and Zhu, D.Q. (2009). Thruster fault-tolerant for UUVs based on quantum-behaved particle swarm optimization opportunities and challenges for next-generation applied intelligence. *Studies in Computational Intelligence*, **214**, 159–165.
- Liu, Y., Fang, P., Bian, D., Zhang, H. and Wang, S. (2014). Fuzzy comprehensive evaluation for the motion performance of autonomous underwater vehicles, *Ocean Engineering*, **88**, 568–577.
- Luo, C. and Yang, S.X. (2008). A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments. *IEEE Transactions on Neural Networks*, **19**, 1279–1298.
- Omerdic, E. and Roberts, G. (2004). Thruster fault diagnosis and accommodation for open-frame underwater vehicles. *Control Engineering Practice*, **12**, 1575–1598.
- Pan, C., Lai, X., Yang, S.X. and Wu, M. (2013). An efficient neural network approach to tracking control of an autonomous surface vehicle with unknown dynamics, *Expert Systems with Applications*, **40**, 1629–1635.

- Sankaranarayanan, V. and Mahindrakar, A.D. (2009). Control of a class of underactuated mechanical systems using sliding modes. *IEEE Transactions on Robotics*, **25**, 459–467.
- Sarkar, N., Podder, T.K. and Antonelli, G. (2002). Fault-accommodating thruster force allocation of an AUV considering thruster redundancy and saturation. *IEEE Transactions on Robotics and Automation*, **18**, 223–233.
- Slotine, J.J. and Li, W. (1991). *Applied Nonlinear Control*. Prentice-Hall, Englewood Cliffs, NJ.
- Soylu, S., Buckham, B.J. and Podhorodeski, R.P. (2008). A chattering-free sliding-mode controller for underwater vehicles with fault-tolerant infinity-norm thrust allocation. *Ocean Engineering*, **35**, 1647–1659.
- Sui, S., Tong, S. and Li, Y. (2014). Adaptive fuzzy backstepping output feedback tracking control of MIMO stochastic pure-feedback nonlinear systems with input saturation. *Fuzzy Sets and Systems*, **254**, 26–46.
- Sun, B., Zhu, D. and Yang, S.X. (2014). A bio-inspired cascaded tracking Control of 7000 m manned submarine vehicle. *IEEE Transactions on Industrial Electronics*, **61**, 3682–3693.
- Woods, S.A., Bauer, R.J. and Seto, M.L. (2012). Automated ballast tank control system for autonomous underwater vehicles. *IEEE Journal of Oceanic Engineering*, **37**, 727–739.
- Yang, S.X., Zhu, A.M., Yuan, G.F. and Meng, M.Q. (2012). A bioinspired neurodynamics-based approach to tracking control of mobile robots. *IEEE Transactions on Industrial Electronics*, **59**, 3211–3220.
- Zhang, Y. and Jiang, J. (2008). Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, **32**, 229–252.
- Zheng, Y. and Jun, W. (2012). Model Predictive Control for Tracking of Underactuated Vessels Based on Recurrent Neural Networks. *IEEE Journal of Oceanic Engineering*, **37**, 717–726.
- Zhu, D. and Sun, B. (2013). Information fusion fault diagnosis method for unmanned underwater vehicle thrusters. *IET Electrical Systems in Transportation*, **3**, 102–111.
- Zhu, D.Q., Liu, Q. and Hu, Z. (2011). Fault-tolerant control algorithm of the manned submarine with multi-thruster based on quantum-behaved particle swarm optimization. *International Journal of Control*, **84**, 1817–1829.
- Zhu, D.Q., Liu, Q. and Yang, S.X. (2008). An active fault-tolerant control method of unmanned underwater vehicles. *International Journal of Advanced Robotic Systems*, **5**, 411–418.
- Zhu, D.Q., Yue, Z. and Yan, M.Z. (2012). A bio-inspired neurodynamics based backstepping path-following control of an AUV with ocean current. *International Journal of Robotics and Automation*, **27**, 280–287.