ADP Based Fault-Tolerant Tracking Control for Underactuated AUV with Actuators Faults via Neural Network Observer

Gaofeng Che (chexuping@163.com)  Henan Normal University  https://orcid.org/0000-0001-6620-5066

Zhen Yu  Xiamen University

Research Article

Keywords: Adaptive dynamic programming (ADP), Fault-tolerant tracking control, Actuators faults, Neural network observer, Autonomous underwater vehicle (AUV)

Posted Date: December 2nd, 2021

DOI: https://doi.org/10.21203/rs.3.rs-807482/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License
ADP based output-feedback fault-tolerant tracking control for underactuated AUV with actuators faults via neural network observer

Gaofeng Che · Zhen Yu*

Received: date / Accepted: date

Abstract In this work, the fault-tolerant tracking control issue of underactuated autonomous underwater vehicle (AUV) with actuators faults is investigated. Firstly, an output-feedback error tracking system is constructed based on the theoretical model of underactuated AUV with actuators faults. Then, an adaptive dynamic programming (ADP) based fault-tolerant control controller is developed. In our proposed control scheme, a neural-network observer is designed to approximate the system states with actuators faults. A novel ADP scheme is constructed with critic neural network and action neural network in order to reduce the jitter in the control input and improve the tracking accuracy. Based on Lyapunov approach, the stability of the error tracking system is guaranteed by the proposed controller. At last, the simulation results show that the underactuated AUV achieves better tracking performance.

Keywords Adaptive dynamic programming (ADP) · Fault-tolerant tracking control · Actuators faults · Neural network observer · Autonomous underwater vehicle (AUV)

1 Introduction

Trajectory tracking is a complex motion control task for autonomous underwater vehicle (AUV) in an unknown underwater environment (Che et al (2019a); Qiao and Zhang (2017); Shen et al (2018); Che et al (2019b)). Many trajectory-tracking control methods have been developed for AUV without actuators faults, such as adaptive terminal-sliding-mode control method (Qiao and Zhang (2017); Zhang et al (2018a)), fuzzy control method (Liu et al (2019)).
Yu et al (2017)), robust control method (Li et al (2019)), cooperative control method (Wang et al (2018)) and so on.

Actuators are very important parts of underactuated AUV. The actuators faults may lead to performance degradation of underactuated AUV (Hao et al (2019); Kadiyam et al (2020)). In order to maintain system stability and the acceptable tracking accuracy, many fault-tolerant control strategies have been developed for AUV with actuators faults, such as adaptive fault-tolerant control method (Liu et al (2018a)), adaptive terminal sliding mode based fault-tolerant control method (Zhang et al (2015)), backstepping based adaptive region-tracking fault-tolerant control method (Zhang et al (2017)) and so on.

The adaptive dynamic programming (ADP) is introduced into this work to transform the trajectory-tracking control problem into optimal control problem for underactuated AUV with actuators faults. Policy iteration (PI) algorithm and value iteration (VI) algorithm are two important ADP procedures to solve the complex Hamilton-Jacobi-Bellman (HJB) equation (Gong et al (2019); Liu et al (2018b); Sun and Liu (2018)).

Many ADP algorithms have been developed to solve the tracking control problems for nonlinear systems in recent years. An infinite-time optimal tracking control problem is investigated based on greedy heuristic dynamic programming (HDP) iteration algorithm (Zhang et al (2008)). The output tracking control problem is solved based on event-driven ADP scheme (Zhang et al (2018b)). The time delays are considered and HDP is designed to solve the tracing control problem for a class of nonlinear systems (Zhang et al (2011)). The ADP based tracking control scheme is designed for coal gasification system (Zhang et al (2014)). The ADP algorithm is designed for tracking control with unknown system dynamics (Kiumarisi and Lewis (2015); Qin et al (2014)). The tracking controller based on ADP scheme is designed for fully-actuated AUV with current disturbances and rudders faults. The neural-network estimators are employed to approximate the current disturbances and rudder faults (Che and Yu (2020)). A PI algorithm is developed for online fault compensation control of a class of affine nonlinear systems with actuators failures (Zhao et al (2016)).

The main contribution of this work can be summarized as follows:

- An output-feedback error tracking system is constructed based on the theoretical model of underactuated AUV with actuators faults.
- The neural network observer is designed to approximate the actuators faults. The approximate actuators faults is introduced into an improved performance index function based on the performance index (Zhang et al (2008); Wei et al (2018); Zhao et al (2017)).
- Because AUV is a very complex nonlinear system, the critic-action neural networks are employed in order to reduce jitter in the control input which are different from the single critic neural network structure(Zhao et al (2017)).
The error tracking system can be guaranteed to be uniformly ultimately bounded (UUB) based on the Lyapunov stability theorem.

The rest of paper is organized as follows. The output feedback based error tracking system is constructed and problem formulation is described in Section 2. In Section 3, the fault-tolerant ADP tracking controller with neural network observer is designed. Simulation examples are provided to demonstrate the effectiveness of the proposed method in Section 4. The conclusion is drawn in Section 5.

2 Theoretical model of underactuated AUV and problem formulation

2.1 Theoretical model of underactuated AUV

Two coordinate systems are used in the theoretical model of underactuated AUV as shown in Fig. 1. One is the inertial coordinate system \( \{O_e\} \) and the other is the body-fixed coordinate system \( \{O_b\} \). The theoretical model of underactuated AUV without actuators faults is presented as follows:

\[
\begin{align*}
\dot{\eta} &= J(\eta)\xi \\
M\ddot{\xi} + C(\xi)\xi + D(\xi)\xi + g(\eta) &= \tau
\end{align*}
\]

(1)

where \( \eta = [x \ y \ z \ \phi \ \theta \ \psi]^T \) is the location vector with respect to the inertial coordinate system; \( \xi = [u \ v \ w \ p \ q \ r]^T \) is the velocity vector with respect to the body-fixed coordinate system; \( M \in \mathbb{R}^{6 \times 6} \) is the inertia matrix; \( C(\xi) \in \mathbb{R}^{6 \times 6} \) is the Coriolis and centripetal matrix; \( D(\xi) \in \mathbb{R}^{6 \times 6} \) is the hydrodynamic damping matrix; \( g(\eta) \in \mathbb{R}^{6 \times 1} \) is the gravitational forces and moment vector; \( \tau \in \mathbb{R}^{6 \times 1} \) is the control forces; \( J(\eta) \in \mathbb{R}^{6 \times 6} \) is the spatial transformation matrix between two coordinate systems.
The kinematics of underactuated AUV is described as follows:
\[
\begin{cases}
\xi = J^{-1}\dot{\eta} \\
\dot{\xi} = J^{-1}\dot{\eta} - J^{-1}\dot{J}J^{-1}\eta
\end{cases}
\] (2)

where \( J = J(\eta) \).

Combining equation (1) with equation (2), we can get
\[
\dot{\eta} = (MJ^{-1})^{-1}(MJ^{-1}\dot{J}J^{-1} - CJ^{-1} - DJ^{-1})\dot{\eta} - (MJ^{-1})^{-1}g + (MJ^{-1})^{-1}\tau
\] (3)

where \( C = C(\xi); D = D(\xi); g = g(\eta) \).

2.2 Problem formulation

The desired trajectory is given as follows:
\[
\dot{\eta}_d = (M_dJ_d^{-1})^{-1}(M_dJ_d^{-1}\dot{J}J_d^{-1} - C_dJ_d^{-1} - D_dJ_d^{-1})\dot{\eta}_d - (M_dJ_d^{-1})^{-1}g_d
\] + \((M_dJ_d^{-1})^{-1}\tau_d
\] (4)

The error vectors are defined as follows:
\[
\begin{cases}
\epsilon_\eta = \eta - \eta_d \\
\epsilon_\tau = \tau - \tau_d
\end{cases}
\] (5)

Then substituting equations (4), (5) into equation (3), the output feedback based error tracking system is given as follows:
\[
\dot{\epsilon}_\eta = (MJ^{-1})^{-1}(MJ^{-1}\dot{J}J^{-1} - C_dJ_d^{-1} - D_dJ_d^{-1})\epsilon_\eta + \Theta + (MJ^{-1})^{-1}\epsilon_\tau
\] (6)

where \( \Theta = (MJ^{-1})^{-1}(M_dJ_d^{-1} - MJ^{-1})\eta_d + (MJ^{-1})^{-1}(MJ^{-1}\dot{J}J^{-1} - CJ^{-1} - DJ^{-1} - M_dJ_d^{-1}\dot{J}J_d^{-1} + C_dJ_d^{-1} + D_dJ_d^{-1})\dot{\eta}_d + (MJ^{-1})^{-1}(g_d - g) \).

We define error vector \( x = [\epsilon_\eta \quad \epsilon_\tau]^T \), then the error tracking system (6) can be transformed as follows:
\[
\dot{x} = \begin{bmatrix} 0 & I \\ 0 & (MJ^{-1})^{-1}(MJ^{-1}\dot{J}J^{-1} - CJ^{-1} - DJ^{-1}) \end{bmatrix} x + \begin{bmatrix} 0 \\ \Theta \end{bmatrix}
\] (7)

where \( I \in \mathbb{R}^{6 \times 6} \) is the identity matrix.

Actuators faults are described as \( f \in \mathbb{R}^{m \times 1} \) and \( m \) is the number of actuators. The real output of actuators with actuators faults is given as follows:
\[
\mu' = \mu - f
\] (8)

where \( \mu \in \mathbb{R}^{m \times 1} \) is the output of controller.

The vector of control forces and control torque \( \epsilon'_\tau \) of underactuated AUV with actuators faults can be represented as follows:
\[
\epsilon'_\tau = \epsilon_\tau - \tau f = B\mu' = B\mu - Bf
\] (9)
where $B \in \mathbb{R}^{6 \times m}$ is the actuators configuration matrix; $e_r = B \mu$; $\tau_f = B f$.

The error tracking system with actuators faults is given as follows:
\[
\dot{x} = \varpi(x) + \rho(x)(\mu - f) \tag{10}
\]

where $\varpi(x) = \begin{bmatrix} 0 & I \\ 0 & (MJ^{-1})^{-1}(MJ^{-1}JJ^{-1} - CJ^{-1} - DJ^{-1}) \end{bmatrix} x + \begin{bmatrix} 0 \\ \Theta \end{bmatrix}$, $\rho(x) = B \mu$.

**Assumption 1** Because underactuated AUV does not have independent actuators in the sway and heave axes, the available controls are the surge force, pitch moment and the yaw moment. The actuators faults $f$ satisfies that $\|f\| = \|K \mu\| \leq \|\mu\| \leq \delta_1$. $K$ is a diagonal matrix and element $k_{ii}$ of diagonal matrix $K$ satisfies $0 \leq k_{ii} < 1$, $\varpi(x)$ and $\rho(x)$ are locally Lipschitz continuous, $\delta_1$ is a positive constant.

The performance index function is defined as follows:
\[
V_1(x, \mu) = \int_{t}^{\infty} e^{\gamma(t-\sigma)}(\beta \hat{f}^T(\sigma) \hat{f}(\sigma) + U(x(\sigma), \mu(\sigma))) d\sigma \tag{11}
\]

where $U(x, \mu) = x^TQx + \mu^TR\mu$ is the utility function and $U(0,0) = 0$; $Q \in \mathbb{R}^{12 \times 12}$ and $R \in \mathbb{R}^{m \times m}$ are positive definite matrices; $\hat{f}$ is the approximate actuators failures $f$; $\gamma$ is a discount factor and $0 \leq \gamma < 1$; $\beta$ is a positive constant.

**Definition 1** A control law $\mu$ is defined as an admissible control policy for the error tracking system (10) with $f = 0$, if $\mu$ is continuous on a set $\Omega \subset \mathbb{R}^{12}$ and can stabilize the error tracking system (10) with $f = 0$, $\mu(0) = 0$ and $V_1(x_0, 0)$ is finite for all $x_0 \in \Omega$.

Based on the optimal control theory, the performance index function (11) is a Lyapunov function and satisfies as follows:
\[
0 = \beta \hat{f}^T \hat{f} + U(x, \mu) + (\nabla V_1(x, \mu))^T(\varpi(x) + \rho(x)\mu) - \gamma V_1(x, \mu) \tag{12}
\]

where $V_1(0,0) = 0$ and $\nabla V_1(x, \mu)$ is the partial derivative of $V_1(x, \mu)$ with respect to $x$. $\nabla V_1(x, \mu) = \frac{\partial V_1(x, \mu)}{\partial x}$.

Then, the Hamiltonian function is defined as follows:
\[
H(x, \mu, \nabla V_1(x, \mu)) = \beta \hat{f}^T \hat{f} + U(x, \mu) + (\nabla V_1(x, \mu))^T(\varpi(x) + \rho(x)\mu) - \gamma V_1(x, \mu) \tag{13}
\]

The optimal cost function is defined as follows:
\[
V_1^*(x, \mu) = \min_{\mu \in \Psi(\Omega)} \int_{t}^{\infty} e^{\gamma(t-\sigma)}(\beta \hat{f}^T \hat{f} + U(x(\sigma), \mu(\sigma))) d\sigma \tag{14}
\]

where $\delta_2$ is a positive constant.
The optimal cost function (14) satisfies the HJB equation, then

$$0 = \min_{\mu} H(x, \mu, \nabla V_1^*(x, \mu))$$ (15)

The optimal control is expressed as follows:

$$\mu^*(x) = -\frac{1}{2} R^{-1} \rho^T(x) \nabla V_1^*(x, \mu)$$ (16)

The PI scheme is designed as shown in Algorithm 1.

### Algorithm 1 Online PI

**Step1:** Select an initial admissible control policy $\mu^{(0)}$ and a positive constant $\epsilon$ and an initial performance index function $\nabla V^{(0)}(x, \mu^{(0)}) = 0$;

**Step2:** Solve $V^{(i)}$ according to

$$0 = \beta f^T f + U(x, \mu^i) + (\nabla V^{(i)}(x, \mu^{(i)}))^T (\omega(x) + \rho(x) \mu^{(i)})$$

$$-\gamma V^{(i)}(x, \mu^{(i-1)}) ;$$

**Step3:** Update the control policy with

$$\mu^{(i+1)} = -\frac{1}{2} R^{-1} \rho^T(x) \nabla V^{(i)}(x, \mu^i);$$

**Step4:** if $\|V^{(i+1)}(x, \mu^{(i+1)}) - V^{(i)}(x, \mu^{(i)})\| \leq \epsilon$, stop the iterations; else return to Step2.

3 Fault-tolerant ADP tracking controller design via neural network observer

3.1 Problem transformation

The structural diagram of neural network observer based fault-tolerant ADP control scheme is shown in Fig. 2.
Assumption 2: The approximate error of actuators faults $e_f = f - \hat{f}$ satisfies that $\|e_f\| \leq \delta_3$, where $\delta_3$ is a positive constant.

Lemma 1 (Zhao et al. (2017, 2016)): With Assumption 1, 2 and the control policy (16) for error tracking system (10) with $f = 0$, the continuously differentiable function $V^*_t(x, \mu)$ is a Lyapunov function if the conditions $\beta \geq \gamma \delta^2 \delta_2 \lambda_{\max}(R^{-1}) + \lambda_{\max}(R)$ and $\|x\| \geq \sqrt{\frac{(\delta_1, \delta_2 \lambda_{\max}(R^{-1}) + \lambda_{\max}(R))(2\delta_1 + \delta_3)}{\lambda_{\min}(Q)}}$ hold. So, the optimal control law (16) is a solution to the error tracking system (10) with $f \neq 0$ and error tracking system (10) with $f \neq 0$ is UUB.

Proof: The derivative of $V^*_t(x, \mu^*)$ is given as follows:

$$
\dot{V}^*_t(x, \mu^*) = (\nabla V^*_t(x, \mu^*)^T x = (\nabla V^*_t(x, \mu^*)^T (x, \mu^*) + \rho(x)(\mu^* - f))
= (\nabla V^*_t(x, \mu^*)^T (x, \mu^*) - (\nabla V^*_t(x, \mu^*)^T \rho(x)f
$$

From equation (15) we have

$$
(\nabla V^*_t(x, \mu^*)^T (x, \mu^*) = -\beta (f\hat{f} - U(x, \mu^*) + \gamma V^*_t(x, \mu^*)
$$

Substituting equation (18) into equation (17), we can get

$$
\dot{V}^*_t(x, \mu^*) = -\beta \hat{f}^T \dot{f} - U(x, \mu^*) + \gamma V^*_t(x, \mu^*) - (\nabla V^*_t(x, \mu^*)^T \rho(x)f
$$

Hence, if the following conditions hold,

$$
\beta \geq \gamma \delta^2 \delta_2 \lambda_{\max}(R^{-1}) + \lambda_{\max}(R)
\|x\| \geq \sqrt{\frac{(\delta_1, \delta_2 \lambda_{\max}(R^{-1}) + \lambda_{\max}(R))(2\delta_1 + \delta_3)}{\lambda_{\min}(Q)}}
$$

Then $\dot{V}^*_t(x, \mu) \leq 0$ and $V^*_t(x, \mu)$ is a Lyapunov function.

The error tracking system (10) is UUB. This completes the proof.
3.2 Design of neural-network observer

For the error tracking system (10), we developed a radial basis function (RBF) neural network to approximate the actuators faults.

\[ f = - (W_0 \varphi(x) + \varepsilon_0) \]  \hspace{1cm} (21)

where \( W_0^T \in \mathbb{R}^{l_0} \) is the ideal weight; \( \varphi_0(x) \in \mathbb{R}^{l_0} \) is the activation function; \( l_0 \) is the neurons number of the hidden layer; \( \varepsilon_0 \) is the approximation error.

Substituting equation (21) into error tracking system (10), we can get

\[ \dot{x} = \varphi(x) + \rho(x)(\mu + W_0 \varphi_0(x) + \varepsilon_0) \]  \hspace{1cm} (22)

Then the neural-network faults observer is designed as follows:

\[ \dot{\hat{x}} = \varphi(\hat{x}) + \rho(\hat{x})(\mu + \hat{W}_0 \varphi_0(\hat{x})) + L(x - \hat{x}) \]  \hspace{1cm} (23)

where \( \hat{x} \) the approximation of \( x \); \( \hat{W}_0 \) is the approximation of \( W_0 \); \( L \in \mathbb{R}^{12 \times 12} \) is the positive matrix.

The weight vector \( \hat{W}_0 \) should be updated as

\[ \dot{\hat{W}}_0 = - \varrho_0 \varrho^T(\hat{x}) e_x \varphi_0^T(\hat{x}) \]  \hspace{1cm} (24)

where \( e_x = \hat{x} - x \) is the approximation error of \( x \), \( \varrho_0 \) is the learning rate and \( \varrho_0 > 0 \).

Combining equation (22) with equation (23), we can get

\[ \dot{e}_x = \dot{\hat{x}} - \dot{x} \]
\[ = \varphi(\hat{x}) + \rho(\hat{x})(\mu + \hat{W}_0 \varphi_0(\hat{x})) - L e_x - (\varphi(x) + \rho(x)(\mu + W_0 \varphi_0(x) + \varepsilon_0)) \]
\[ = -L e_x + (\varphi(\hat{x}) - \varphi(x)) + (\rho(\hat{x}) - \rho(x)) \mu + \rho(\hat{x}) \hat{W}_0 \varphi_0(\hat{x}) \]
\[ -\rho(x) W_0 \varphi_0(x) - \rho(x) \varepsilon_0 \]
\[ = -L e_x + \varphi - \rho \mu + \rho(\hat{x}) \hat{W}_0 \varphi_0(\hat{x}) - \rho(\hat{x}) \hat{W}_0 \varphi_0(\hat{x}) + \rho(\hat{x}) W_0 \varphi_0(\hat{x}) \]
\[ -\rho(x) W_0 \varphi_0(x) - \rho(x) \varepsilon_0 \]
\[ = -L e_x + \varphi - \rho \mu + \rho(\hat{x}) \hat{W}_0 \varphi_0(\hat{x}) + \rho(\hat{x}) W_0 \varphi_0(\hat{x}) - \rho(x) W_0 \varphi_0(x) - \rho(x) \varepsilon_0 \]
\[ = -L e_x + \varphi - \rho \mu + \rho(\hat{x}) \hat{W}_0 \varphi_0(\hat{x}) + \rho(\hat{x}) W_0 \varphi_0(\hat{x}) - \rho(x) \varepsilon_0 \]
\[ = -L e_x + \varphi - \rho \mu + \rho(\hat{x}) \hat{W}_0 \varphi_0(\hat{x}) + \rho(\hat{x}) W_0 \varphi_0(x) - \rho(x) \varepsilon_0 \]
\[ = -L e_x + \varphi - \rho \mu + \rho(\hat{x}) \hat{W}_0 \varphi_0(x) + \rho(\hat{x}) W_0 \varphi_0(x) - \rho(x) \varepsilon_0 \]
\[ = -L e_x + \varphi - \rho \mu + \rho(\hat{x}) \hat{W}_0 \varphi_0(\hat{x}) + \rho(\hat{x}) W_0 \varphi_0(x) - \rho(x) \varepsilon_0 \]
\[ = -L e_x + \varphi - \rho \mu + \rho(\hat{x}) \hat{W}_0 \varphi_0(x) + \rho(\hat{x}) W_0 \varphi_0(x) - \rho(x) \varepsilon_0 \]
\[ = -L e_x + \varphi - \rho \mu + \rho(\hat{x}) \hat{W}_0 \varphi_0(x) + \rho(\hat{x}) W_0 \varphi_0(x) - \rho(x) \varepsilon_0 \]
\[ = -L e_x + \varphi - \rho \mu + \rho(\hat{x}) \hat{W}_0 \varphi_0(x) + \rho(\hat{x}) W_0 \varphi_0(x) - \rho(x) \varepsilon_0 \]
\[ = -L e_x + \varphi - \rho \mu + \rho(\hat{x}) \hat{W}_0 \varphi_0(x) + \rho(\hat{x}) W_0 \varphi_0(x) - \rho(x) \varepsilon_0 \]
\[ \text{Assumption 3} \quad \varphi, \rho \mu, \rho \varphi_0(\hat{x}), \rho(x) W_0 \varphi_0(x) \text{ and } \rho(x) \varepsilon_0 \text{ are norm-bounded as } \| \varphi \| \leq \delta_4, \| \rho \mu \| \leq \delta_5, \| \rho \varphi_0(\hat{x}) \| \leq \delta_6, \| \rho(x) W_0 \varphi_0(x) \| \leq \delta_7 \text{ and } \| \rho(x) \varepsilon_0 \| \leq \delta_8 \text{; } \delta_4, \delta_5, \delta_6, \delta_7 \text{ and } \delta_8 \text{ are positive constants.} \]

Theorem 1 With Assumptions 1, 3, the updating law (24) of weight vector (23) can guarantee \( e_x \) to be UUB based on the neural network observer.
Proof Select an Lyapunov function as

\[ V_2 = \frac{1}{2}e_x^T e_x + \frac{1}{2\theta_0} tr[\hat{W}_0^T \hat{W}_0] \]  

Substituting equation (25) into the time derivative of equation (26), we can get

\[ \dot{V}_2 = e_x^T \dot{e}_x + \frac{1}{\theta_0} tr[\dot{\hat{W}}_0^T \hat{W}_0] \\
= e_x^T (-L e_x + \bar{z} + \bar{\mu} + \rho(\dot{x})\bar{W}_0 \varphi_0(\dot{x}) + \bar{\varphi}_0 \dot{W}_0 \varphi_0(\dot{x}) + \bar{\varphi}_0 \dot{W}_0 \varphi_0(\dot{x}) - \rho(x)\varepsilon_0) \\
- tr[(\rho^T(\dot{x}) e_x \varphi_0^T(\dot{x}))^T \hat{W}_0] \\
= e_x^T (-L e_x + \bar{z} + \bar{\mu} + \rho(\dot{x})\bar{W}_0 \varphi_0(\dot{x}) + \bar{\varphi}_0 \dot{W}_0 \varphi_0(\dot{x}) + \rho(x)\dot{\varphi}_0 - \rho(x)\varepsilon_0) \\
- tr[\varphi_0(\dot{x}) e_x^T \rho(\dot{x}) W_0] \\
= e_x^T (-L e_x + \bar{z} + \bar{\mu} + \rho W_0 \varphi_0(\dot{x}) + \rho(x)W_0 \varphi_0 - \rho(x)\varepsilon_0) \\
\leq -\lambda_{\text{min}}(L)e_x^T e_x + e_x^T (\bar{z} + \bar{\mu} + \rho W_0 \varphi_0(\dot{x}) + \rho(x)W_0 \varphi_0 - \rho(x)\varepsilon_0) \\
\leq -\frac{11\lambda_{\text{min}}(L)}{16} e_x^T e_x + \frac{1}{\lambda_{\text{min}}(L)}(\bar{\varphi}_0) \dot{W}_0 \varphi_0(\dot{x}) + (\rho(x)\varepsilon_0) + (\rho(x)\varepsilon_0)^T (\rho(x)\varepsilon_0) \\
\leq -\frac{11\lambda_{\text{min}}(L)}{16} e_x^T e_x + \frac{1}{\lambda_{\text{min}}(L)}(\bar{\varphi}_0) \dot{W}_0 \varphi_0(\dot{x}) + (\rho(x)\varepsilon_0)^2 + (\rho(x)\varepsilon_0)^2 \\
\leq -\frac{11\lambda_{\text{min}}(L)}{16} e_x^T e_x + \frac{1}{\lambda_{\text{min}}(L)}(\bar{\varphi}_0)^2 + (\rho(x)\varepsilon_0)^2 \\
\leq -\frac{11\lambda_{\text{min}}(L)}{16} e_x^T e_x + \frac{1}{\lambda_{\text{min}}(L)} (\delta_x^2 + \delta_e^2 + \delta_x^2 + \delta_e^2) \]  

We can conclude that \( \dot{V}_2 < 0 \) if \( e_x \) satisfies \( \|e_x\| > \frac{n}{\lambda_{\text{min}}} \sqrt{\delta_x^2 + \delta_e^2 + \delta_x^2 + \delta_e^2} \).

Based on the Lyapunov stability theorem, \( e_x \) is guaranteed to be UUB. This completes the proof.

3.3 Design of critic neural network

The ADP controller consists of critic neural network and action neural network. The critic neural network is utilized to approximate \( V^*_c(x, \mu) \).

\[ V_c(x, \mu) = W_c \varphi_c(x, \mu) + \varepsilon_c \]  

where \( W_c \in \mathbb{R}^{l_1} \) is the ideal weight; \( \varphi_c(x, \mu) \in \mathbb{R}^{l_1} \) is the activation function; \( l_1 \) is the neurons number of the hidden layer; \( \varepsilon_c \) is the approximation error.

The derivative of the cost function \( V_c(x, \mu) \) is given as follows

\[ \nabla V_c(x, \mu) = (\nabla \varphi_c(x, \mu))^T W_c^T + \nabla \varepsilon_c \]  

where \( \nabla \varphi_c(x, \mu) = \frac{\partial \varphi_c(x, \mu)}{\partial x} \) and \( \nabla \varepsilon_c = \frac{\partial \varepsilon_c}{\partial x} \).

Substituting equation (29) into equation (12), we can obtain

\[ 0 = \beta \bar{f}^T \bar{f} + U(x, \mu) + (W_c \nabla \varphi_c(x, \mu) + \nabla \varepsilon_c)(\varphi(x) + \rho(x)(\mu - \bar{f}) - \gamma (W_c \varphi_c(x, \mu) + \varepsilon_c) \]
Then the Hamiltonian function can be expressed as follows:

\[
H(x, \mu, W_c) = \beta f^T \hat{f} + U(x, \mu) + W_c \nabla \phi_c(x, \mu)(\varpi(x) + \rho(x) (\mu - \hat{f})) - \gamma W_c \phi_c(x, \mu)
\]

\[= e_c \] (31)

where \( e_c \) is the residual error.

Then, \( V_3(x, \mu) \) is approximated as follows:

\[
\hat{V}_3(x, \mu) = \hat{W}_c \phi_c(x, \mu)
\] (32)

where \( \hat{W}_c \) is the approximation of \( W_c \).

The derivative of \( \hat{V}_3(z) \) can be expressed as follows:

\[
\nabla \hat{V}_3(x, \mu) = (\nabla \phi_c(x, \mu))^T \hat{W}_c^T
\] (33)

Then, the approximate Hamiltonian function can be expressed as follows:

\[
\hat{H}(x, \mu, \hat{W}_c) = \beta \hat{f}^T \hat{f} + U(x, \mu) + \hat{W}_c \nabla \phi_c(x, \mu)(\varpi(x) + \rho(x) (\mu - \hat{f})) - \gamma \hat{W}_c \phi_c(x, \mu)
\]

\[= \hat{e}_c \] (34)

Given any admissible control policy \( \mu \), it is desired to select \( \hat{W}_c \) to minimize the squared residual error \( E_c(\hat{W}_c) \) as

\[
E_c(\hat{W}_c) = \frac{1}{2} \hat{e}_c^T \hat{e}_c
\] (35)

The weight update law for the critic neural network is given as follows

\[
\dot{\hat{W}}_c = -\frac{g_1 \hat{e}_c \varsigma_1}{(1 + \varsigma_1^2)}
\] (36)

where \( g_1 \) is the learning rate of critic neural network and \( g_1 \) satisfies that \( g_1 > 0; \varsigma_1 = \nabla \phi_c(x, \mu)(\varpi(x) + \rho(x) (\mu - \hat{f})) - \gamma \phi_c(x, \mu) \) and \( \varsigma_1 \in \mathbb{R}^{1} \).

The approximate weight error of critic neural network is defined as \( \tilde{W}_c = \hat{W}_c - W_c \). Then, equation (36) can be transformed as follows:

\[
\dot{\tilde{W}}_c = -\frac{g_1}{(1 + \varsigma_1^2)}(\tilde{W}_c \varsigma_1 - \nabla \varepsilon_c (\varpi(x) + \rho(x) (\mu - \hat{f})) - \gamma \varepsilon_c) \varsigma_1^T
\] (37)

**Assumption 4** \( \|\nabla \varepsilon_c (\varpi(x) + \rho(x) (\mu - \hat{f}))) + \gamma \varepsilon_c\| \leq \delta_9 \) and \( \varsigma_{\text{min}} \leq \|\varsigma_1\| \leq \varsigma_{\text{max}} \), where \( \delta_9 \), \( \varsigma_{\text{min}} \) and \( \varsigma_{\text{max}} \) are positive constants.

**Theorem 2** The approximate weight error is UUB, if the weight of the critic neural network is updated by (37).
Proof Select an Lyapunov function as

\[ V_4 = \frac{(1 + \varsigma_1^T\varsigma_1)^2}{2\varsigma_1} \dot{W}_c \dot{W}_c^T \]  

(38)

Then, the time derivative of \( V_4 \) is

\[ \dot{V}_4 = \frac{(1 + \varsigma_1^T\varsigma_1)^2}{2\varsigma_1} \ddot{W}_c \dot{W}_c^T \]

\[ = -(\dot{W}_{e}\varsigma_1 - \nabla \epsilon_c(x) + \rho(x)(\mu - \hat{f})) - \gamma \epsilon_c \varsigma_1^T \dot{W}_c \]

\[ = -W_{e}\varsigma_1^T \dot{W}_c + (\nabla \epsilon_c(x) + \rho(x)(\mu - \hat{f})) + \gamma \epsilon_c \varsigma_1^T \dot{W}_c \]

\[ \leq -\frac{1}{2} \|W_{e}\varsigma_1\|^2 + \frac{1}{2} \|\nabla \epsilon_c(x) + \rho(x)(\mu - \hat{f})\| + \gamma \epsilon_c \|

Hence, \( \dot{V}_4 < 0 \) if \( \|\dot{W}_c\| > \frac{\gamma \epsilon_c}{\|\nabla \epsilon_c\|} \). The approximate weight error is UUB, according to the Lyapunov stability theorem. This completes the proof.

3.4 Design of action neural network

The optimal control \( \mu^\ast \) is approximated by the action neural network as

\[ \mu = W_a \varphi_a(x) + \epsilon_a \]

(40)

where \( W_a \in \mathbb{R}^{l_2} \) is the ideal weight; \( \varphi_a(x) \in \mathbb{R}^{l_2} \) is the activation function; \( l_2 \) is the neurons number of the hidden layer; \( \epsilon_a \) is the approximation error.

Because the ideal weight \( W_a \) is unknown, \( \mu^\ast \) is approximated as follows:

\[ \dot{\mu} = \dot{W}_a \varphi_a(x) \]

(41)

where \( \dot{W}_a \) is the estimate of \( W_a \).

The approximate feedback error used for training action neural network is defined as the difference between the feedback control input applied to the error tracking system (10) and the optimal control \( \mu^\ast \) as

\[ \dot{\epsilon}_a = \dot{W}_a \varphi_a(x) + \frac{1}{2} R^{-1} \rho^T(x)(\nabla \varphi_a(x, \mu)) \dot{W}_c \]

(42)

The action neural network is defined to minimize the objective function as

\[ E_a(\dot{W}_a) = \frac{1}{2} \dot{\epsilon}_a^T \dot{\epsilon}_a \]

(43)

The weight updating law for the action neural network is given as follows

\[ \dot{W}_a = -\varphi_2 \dot{\epsilon}_a \varphi_a^T(x) \]

(44)

where \( \varphi_2 \) is the learning rate of action neural network and \( \varphi_2 > 0 \).

According to equations (16), (29) and (40), we have

\[ 0 = W_a \varphi_a(x) + \epsilon_a + \frac{1}{2} R^{-1} \rho^T(x)(\nabla \varphi_a(x, \mu)) \dot{W}_c + \nabla \epsilon_c \]

(45)
The approximate weight error of action neural network is defined as \( \hat{W}_a = \hat{W}_a - W_a \). Then, equation (44) can be transformed as follows:

\[
\dot{\hat{W}}_a = -\rho_2(W_a \varphi_a(x) + \frac{1}{2} R^{-1} \rho^T(x) (\nabla \varphi_c(x, \mu))^T \hat{W}_c^T - \varepsilon_a - \frac{1}{2} R^{-1} \rho^T(x) \nabla \varphi_c(x) \varphi_a^T(x) \tag{46}
\]

**Assumption 5** \( \|R^{-1} \rho^T(x) (\nabla \varphi_c(x, \mu))^T \| \leq \delta_{10}, \| \varepsilon_a + \frac{1}{2} R^{-1} \rho^T(x) \nabla \varphi_c(x) \| \leq \delta_{11} \) and \( \varphi_{a, \min} \leq \| \varphi_a(x) \| \leq \varphi_{a, \max} \), where \( \delta_{10}, \delta_{11} \) and \( \varphi_{a, \min} \) and \( \varphi_{a, \max} \) are positive constants.

**Theorem 3** The approximate weight error is UUB, if the weight of the action neural network is updated by (46).

**Proof** Select an Lyapunov function as

\[ V_5 = \frac{1}{2 \rho_2} \hat{W_a}^T \hat{W_a} \tag{47} \]

Then, the time derivative of \( V_5 \) is

\[
\dot{V}_5 = \frac{1}{2 \rho_2} \hat{W_a}^T \dot{\hat{W}_a} = -\rho_2(W_a \varphi_a(x) + \frac{1}{2} R^{-1} \rho^T(x) (\nabla \varphi_c(x, \mu))^T \hat{W}_c^T - \varepsilon_a - \frac{1}{2} R^{-1} \rho^T(x) \nabla \varphi_c(x) \varphi_a^T(x) \hat{W}_a^T
\]

\[
\dot{V}_5 = -\rho_2(W_a \varphi_a(x) - \frac{1}{2} R^{-1} \rho^T(x) (\nabla \varphi_c(x, \mu))^T \hat{W}_c^T \varphi_a^T(x) - \varepsilon_a + \frac{1}{2} R^{-1} \rho^T(x) \nabla \varphi_c(x) \varphi_a^T(x) \hat{W}_a^T
\]

\[
\leq -\frac{1}{4} \| \hat{W}_a \varphi_a(x) \|^2 + \frac{1}{2} R^{-1} \rho^T(x) (\nabla \varphi_c(x, \mu))^T \hat{W}_c^T \hat{W}_a^T
\]

\[
+ \frac{1}{2} \| \varepsilon_a + \frac{1}{2} R^{-1} \rho^T(x) \nabla \varphi_c(x) \|^2
\]

\[
- \frac{1}{4} \| \hat{W}_a \varphi_a(x) \|^2 + \frac{1}{2} \delta_{10}^2 \| W_c \|^2 + \frac{1}{2} \delta_{11}^2
\]

Hence, \( \dot{V}_5 < 0 \) if \( \| \hat{W}_a \| > \frac{\sqrt{\delta_{10}^2 \| W_c \|^2 + 2 \delta_{11}^2}}{\varphi_{a, \min}} \). The weight approximation error is UUB, according to the Lyapunov stability theorem. This completes the proof.

### 3.5 Stability analysis

**Assumption 6** \( \| \varphi(x) \| \leq \delta_{12} \) and \( \| \rho(x) \| \leq \delta_{13} \), where \( \delta_{12} \) and \( \delta_{13} \) are positive constants.

**Theorem 4** With the performance index function (11), the error tracking system (10) can be guaranteed to be UUB by the approximate fault-tolerant tracking control policy (41).

**Proof** Select an Lyapunov function as

\[ V_6 = \frac{1}{2} x^T x + V_1^* \tag{49} \]
Then, the time derivative of $V_6$ is

$$
\dot{V}_6 = x^T \dot{x} + (\nabla V_1)^T \dot{x}
= x^T (\varpi(x) + \rho(x)(\mu - f)) + (\nabla V_1)^T(\varpi(x) + \rho(x)(\mu - f))
= x^T \varpi(x) + x^T \rho(x)\mu - x^T \rho(x)f - (\nabla V_1)^T \rho(x)f
+ (\nabla V_1)^T(\varpi(x) + \rho(x)\mu)
$$

(50)

According to equations (13), (15), the equation (50) can be transformed as follows:

$$
\dot{V}_6 = x^T \varpi(x) + x^T \rho(x)\mu - x^T \rho(x)f - (\nabla V_1)^T \rho(x)f - x^T Qx - \mu^T R\mu
- \beta \dot{f}^T f + \gamma \dot{V}_4^* \\
= x^T \varpi(x) + x^T \rho(x)\mu - x^T \rho(x)f + 2\mu^T Rf - x^T Qx - \mu^T R\mu
- \beta \dot{f}^T f + \gamma \dot{V}_4^* \\
\leq \frac{5}{2} x^T \varpi(x) + \frac{1}{2} x^T \rho(x)\mu - x^T \rho(x)f + \frac{1}{2} \mu^T R^T \rho(x)\mu + \frac{1}{2} \rho(x)^T \rho(x)f
+ \mu^T \mu + \dot{f}^T R^T f - x^T Qx - \mu^T R\mu - \beta \dot{f}^T f + \gamma \delta_2 \\
\leq -\left(\lambda_{\text{min}}(Q) - \frac{3}{2}\right)\|x\|^2 - \left(\lambda_{\text{min}}(Q) - \frac{1}{2}\delta_{13} - 1\right)\|\mu\|^2 - \left(\beta - \frac{1}{2}\delta_{13}^2 - \lambda_{\text{max}}(R')\right)\|f\|^2
+ \left(\frac{1}{2}\delta_{13}^2 + \lambda_{\text{max}}(R')\right)(2\delta_1 + \delta_3)\delta_3 + \frac{1}{2}\delta_{12}^2 + \gamma \delta_2
$$

(51)

where $R' = R^T R$.

Hence, $\dot{V}_6 < 0$ if $(\lambda_{\text{min}}(Q) - \frac{3}{2}) > 0$, $(\lambda_{\text{min}}(Q) - \frac{1}{2}\delta_{13} - 1) \geq 0$, $(\beta - \frac{1}{2}\delta_{13}^2 - \lambda_{\text{max}}(R')) > 0$ and $\|x\| > \sqrt{\frac{(1/2}\delta_{13}^2 + \lambda_{\text{max}}(R')(2\delta_1 + \delta_3)\delta_3 + \frac{1}{2}\delta_{12}^2 + \gamma \delta_2}{\lambda_{\text{min}}(Q) - \frac{3}{2}}}$. The error tracking system (10) is UUB, according to the Lyapunov stability theorem. This completes the proof.

4 Simulation results

In order to show the effectiveness of the proposed fault-tolerant tracking control based on ADP, two simulation examples are given in this section. According to the kinematic and dynamic model of underactuated AUV (1) with the conditions that are $\eta(4) = 0$ and $\xi(4) = 0$, the matrices $M$, $C(\xi)$, $D(\xi)$ and $g(\eta)$, $J$ are described as follows:

$$
M = \begin{bmatrix}
215 & 0 & 0 & 0 & 0 & 0 \\
0 & 265 & 0 & 0 & 0 & 0 \\
0 & 0 & 265 & 0 & 0 & 0 \\
0 & 0 & 0 & 80 & 0 & 0 \\
0 & 0 & 0 & 0 & 80 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
$$

(52)

$$
C(\xi) = \begin{bmatrix}
0 & 0 & 0 & 0 & 265 \nu & -265 \nu \\
0 & 0 & 0 & 0 & 0 & 215 \mu \\
0 & 0 & 0 & -215 \mu & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
-50 \nu & 0 & 0 & 0 & 0 & 0 \\
50 \nu & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
$$

(53)
\[
D(\xi) = \begin{bmatrix}
D_1 & 0 & 0 & 0 & 0 & 0 \\
0 & D_2 & 0 & 0 & 0 & 0 \\
0 & 0 & D_3 & 0 & 0 & 0 \\
0 & 0 & 0 & D_4 & 0 & 0 \\
0 & 0 & 0 & 0 & D_5 & 0 \\
0 & 0 & 0 & 0 & 0 & D_6
\end{bmatrix}
\] (54)

where \( D_1 = 70 + 100|u|; \ D_2 = 100 + 200|v|; \ D_3 = 100 + 200|w|; \ D_4 = 0; \)
\( D_5 = 50 + 100|q|; \) and \( D_6 = 50 + 100|r|. \)

\[
g(\eta) = \begin{bmatrix} 0 & 0 & - (1822.25 - G) \cos(\theta) & 0 & (18.22225 - 0.01G) \sin(\theta) \end{bmatrix}^T
\] (55)

where \( G \) is the gravity.

\[
J = \begin{bmatrix}
c\theta c\psi & -s\psi & s\theta c\psi & 0 & 0 & 0 \\
c\theta s\psi & c\psi & s\theta s\psi & 0 & 0 & 0 \\
-s\theta & 0 & c\theta & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \sec \theta
\end{bmatrix}
\] (56)

where \( c\theta = \cos \theta; \ s\theta = \sin \theta; \ s\psi = \sin \psi; \ c\psi = \cos \psi \)

4.1 Example one without actuators faults

Given \( \tau_f = 0, \ \varphi_0 = 0.1, \ \varphi_1 = 0.02, \ \varphi_2 = 0.04, \ \gamma = 0.3, \ B = 1822.25, \ \beta = 0.15 \) and \( \tau_d = [500 \ 0 \ 0 \ 0 \ 200 \ 10]^T \), the simulation results are given as follows.

Fig. 3 Tracking error of desired position
Fig. 3 and Fig. 4 show the tracking error of desired position and the tracking error of desired velocity with respect to inertial coordinate system compared with the existing method (Zhao et al. (2017, 2016)) respectively. The tracking trajectory is shown in Fig. 5. The method proposed in this work has received almost the same results with the existing method (Zhao et al. (2017, 2016)). From Fig. 3 and Fig. 4, we can know that the error tracking system (10) is bounded stable. The absolute value of the tracking error of desired position is no more than the threshold value 0.2. The absolute value of the tracking error of desired velocity is no more than the threshold value 0.05. From Fig. 5, we know that the value of the error trajectory between the desired trajectory and the simulation trajectory with the method proposed in this work is no bigger than 0.1m.

4.2 Example two with actuators faults

In this simulation example, we used the parameters values of example one except for $\tau_f = 0.1(\tau_c + \tau_d)$ and $\tau_f = 0.2(\tau_c + \tau_d)$ respectively. The simulation results compared with the existing method (Zhao et al. (2017, 2016)) are given as follows.
Fig. 5 AUV trajectory

(a) Tracking error of desired position with $\tau_f = 0.1(\tau_e + \tau_d)$
(b) Tracking error of desired position with $\tau_f = 0.2(\tau_e + \tau_d)$

Fig. 6 Tracking error of desired position

Fig.6 and Fig.7 show the tracking error of desired position and the tracking error of desired velocity with respect to inertial coordinate system respectively compared with the existing method(Zhao et al (2017, 2016)). From Fig.7, we know that the jitter happens with the existing method(Zhao et al (2017, 2016)) from 0 second to 15 second, and when actuators faults $\tau_f$ is $0.2(\tau_e + \tau_d)$, the jitter is bigger. The method proposed in this paper has received better results without jitter. The tracking error of desired position and the tracking error of desired velocity are tracking system are bounded.

Fig.8 and Fig.9 give the estimated actuators faults based on the RBF neural network when the values of $\tau_f$ are $0.1(\tau_e + \tau_d)$ and $0.2(\tau_e + \tau_d)$ respectively. With the actuators faults, the jitter happened in the estimation of actuators.
faults with existing method (Zhao et al. (2017, 2016)). When the actuators faults became bigger, the jitter became bigger.

Fig. 10 shows the tracking trajectories with \( \tau_f = 0.1(\tau_e + \tau_d) \) and \( \tau_f = 0.2(\tau_e + \tau_d) \). From the simulation results, we know that the value of the error trajectory between the desired trajectory and the simulation trajectory with
(a) Estimated actuators faults with $\tau_f = 0.1(\tau_e + \tau_d)$
(b) Estimated actuators faults with $\tau_f = 0.2(\tau_e + \tau_d)$

Fig. 9 Estimated actuators faults based on RBF neural network with the existing method (Zhao et al. 2017, 2016)

(a) AUV trajectory with $f = 0.1(\tau_e + \tau_d)$
(b) AUV trajectory with $f = 0.2(\tau_e + \tau_d)$

Fig. 10 AUV trajectory

the method proposed in this work is no more than 0.3 m. The trajectory with the method proposed in this work is more close to the desired trajectory.

5 Conclusion

ADP tracking technique via neural network observer based on the error tracking system with actuators faults has been designed. The simulation examples are developed without actuators faults and with actuators faults respectively for the development of the fault-tolerant tracking control scheme to achieve better tracking performance. When the actuators faults happened, the jitter appeared with the existing method (Zhao et al. 2017, 2016)). The jitter will become bigger when the actuators faults become bigger and make the system
unstable. However, the method proposed in this work make the error tracking system bounded stable with the actuators faults based on the Lyapunov stability. Simulation results have shown excellent performance of the closed-loop system compared with the existing method (Zhao et al. (2017, 2016)).

References

Che G, Yu Z (2020) Neural-network estimators based fault-tolerant tracking control for auv via adp with rudders faults and ocean current disturbance. Neurocomputing 411:442–454
Che G, Liu L, Yu Z (2019a) An improved ant colony optimization algorithm based on particle swarm optimization algorithm for path planning of autonomous underwater vehicle. J Amb Intel Hum Comp 11:3349–3354
Che G, Liu L, Yu Z (2019b) Nonlinear trajectory-tracking control for autonomous underwater vehicle based on iterative adaptive dynamic programming. J Intell Fuzzy Syst 37(3):4205–4215
Gong L, Wang Q, Dong C (2019) Spacecraft output feedback attitude control based on extended state observer and adaptive dynamic programming. J Frankl Inst 356:4971–5000
Hao LY, Yu Y, Li H (2019) Fault tolerant control of unv based on sliding mode output feedback. Appl Math Comput 359:433–455
Kadiyam J, Parashar A, Mohan S, Deshmukh D (2020) Actuator fault-tolerant control study of an underwater robot with four rotatable thrusters. Ocean Eng 197, DOI 10.1016/j.oceaneng.2020.106929
Kiumarsi B, Lewis F (2015) Actor-critic-based optimal tracking for partially unknown nonlinear discrete-time systems. IEEE T Neural Netw Learn Syst 26(1):140–151
Li J, Du J, Sun Y, Lewis F (2019) Robust adaptive trajectory tracking control of underactuated autonomous underwater vehicles with prescribed performance. Int J Robust Nonlin 29(14):4629–4643
Liu X, Zhang M, Yao F (2018a) Adaptive fault tolerant control and thruster fault reconstruction for autonomous underwater vehicle. Ocean Eng 155:10–23
Liu X, Zhang M, Rogers E (2019) Trajectory tracking control for autonomous underwater vehicles based on fuzzy re-planning of a local desired trajectory. IEEE T Veh Technol 68(12):11657–11667
Liu Y, Zhang H, Yu R, Qu Q (2018b) Data-driven optimal tracking control for discrete-time system with delays using adaptive dynamic programming. J Frankl Inst 355:5649–5666
Qiao L, Zhang W (2017) Adaptive non-singular integral terminal sliding mode tracking control for autonomous underwater vehicles. IET Control Theory A 11(8):1293–1306
Qin C, Zhang H, Luo Y (2014) Online optimal tracking control of continuous-time linear systems with unknown dynamics by using adaptive dynamic programming. Int J Control 87(5):1000–1009
Shen C, Shi Y, Buckham B (2018) Trajectory tracking control of an autonomous underwater vehicle using Lyapunov-based model predictive control. IEEE T Indust Electron 65(7):5796–5805

Sun J, Liu C (2018) Backstepping-based adaptive dynamic programming for missile-target guidance systems with state and input constraints. J Frankl Inst 355:8412–8440

Wang H, Liu K, Li S (2018) Command filter based globally stable adaptive neural control for cooperative path following of multiple underactuated autonomous underwater vehicles with partial knowledge of the reference speed. Neurocomputing 275:1478–1489

Wei Q, Liu D, Lin Q, Song R (2018) Adaptive dynamic programming for discrete-time zero-sum games. IEEE T Neural Netw Learn Syst 29(4):957–969

Yu C, Xiang X, Lapierre L, Zhang Q (2017) Nonlinear guidance and fuzzy control for three-dimensional path following of an underactuated autonomous underwater vehicle. Ocean Eng 146:457–467

Zhang G, Huang H, Qin H, Wan L, Li Y, Cao J, Su Y (2018a) A novel adaptive second order sliding mode path following control for a portable AUV. Ocean Eng 151:82–92

Zhang H, Wei Q, Luo Y (2008) A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm. IEEE T Syst Man Cy B 38(4):937–942

Zhang H, Song R, Wei Q, Zhang T (2011) Optimal tracking control for a class of nonlinear discrete-time systems with time delays based on heuristic dynamic programming. IEEE T Neural Netw 22(12):1851–1862

Zhang H, Song R, Wei Q, Zhang T (2014) Adaptive dynamic programming for optimal tracking control of unknown nonlinear systems with application to coal gasification. IEEE T Autom Sci Eng 11(4):1020–1036

Zhang K, Zhang H, Jiang H, Y Wang Y (2018b) Near-optimal output tracking controller design for nonlinear systems using an event-driven ADP approach. Neurocomputing 309:168–178

Zhang M, Liu X, Yin B, Liu W (2015) Adaptive terminal sliding mode based thruster fault tolerant control for underwater vehicle in time-varying ocean currents. J Frankl Inst 352:4935–4961

Zhang M, Liu X, Wang F (2017) Backstepping based adaptive region tracking fault tolerant control for autonomous underwater vehicles. J Navigation 70:184–204

Zhao B, Liu D, Li Y (2016) Online fault compensation control based on policy iteration algorithm for a class of affine non-linear systems with actuator failures. IET Control Theory A 10(15):1816–1823

Zhao B, Liu D, Li Y (2017) Observer based adaptive dynamic programming for fault tolerant control of a class of nonlinear systems. Inform Sciences 384:21–33