ORIGINAL ARTICLE

Robust-adaptive dynamic programming-based time-delay control of autonomous ships under stochastic disturbances using an actor-critic learning algorithm

Hossein Nejatbakhsh Esfahani[1] · Rafal Szlapczynski[1]

Abstract

This paper proposes a hybrid robust-adaptive learning-based control scheme based on Approximate Dynamic Programming (ADP) for the tracking control of autonomous ship maneuvering. We adopt a Time-Delay Control (TDC) approach, which is known as a simple, practical, model free and roughly robust strategy, combined with an Actor-Critic Approximate Dynamic Programming (ACADP) algorithm as an adaptive part in the proposed hybrid control algorithm. Based on this integration, Actor-Critic Time-Delay Control (AC-TDC) is proposed. It offers a high-performance robust-adaptive control approach for path following of autonomous ships under deterministic and stochastic disturbances induced by the winds, waves, and ocean currents. Computer simulations have been conducted under two different conditions in terms of the deterministic and stochastic disturbances and all simulation results indicate an acceptable performance in tracking of paths for the proposed control algorithm in comparison with the conventional TDC approach.

Keywords Autonomous ships · Robust adaptive · Actor-critic learning · Approximate dynamic programming · Time-delay control

1 Introduction

Marine autonomous vehicles have been impressively improved during the last decade. Their applications are very extensive ranging from inspection tasks to accomplishing complex underwater explorations. Nowadays, intelligent transportation using the autonomous platforms has become a new research area as state of the art for researcher. In a few years from now, autonomous ships are expected to be used in the sea transportation to carry the goods without assistance of human operator in their navigation. The International Maritime Organization (IMO) specifies four degrees of autonomy for autonomous ships:

ship with automated processes and decision support: seafarers are on board to operate and control shipboard systems and functions. Some operations may be automated.

Remotely controlled ship with seafarers on board: the ship is controlled and operated from another location, but seafarers are on board.

Remotely controlled ship without seafarers on board: the ship is controlled and operated from another location. There are no seafarers on board.

Fully autonomous ship: the operating system of the ship is able to make decisions and determine actions by itself.

As can be seen, the role of human operator is expected to be reduced throughout the development of autonomous ships. It can also be expected that during this process, the exploited ships may fuse two or more degrees of autonomy. The main difference between manned and unmanned ships in terms of control systems is that autonomous ship control system relies on three dependent subsystems which are known as navigation, guidance and control (NGC) systems, while there is no a full decision-making system for the manned ships in particular for the collision avoidance and steering goals.

Designing a high-performance motion control system for these platforms is most significant challenge due to the complexity and nonlinearity of the ship dynamics as well as time-varying external disturbances. There are two groups of

✉ Hossein Nejatbakhsh Esfahani
hossein.esfahani@pg.edu.pl

[1] Faculty of Ocean Engineering and Ship Technology, Gdansk University of Technology, Gdansk, Poland

perturbations, which affect the autonomous ship's motion. The first group stems from the uncertainties in autonomous ship's dynamics caused by parameter variation in the inertia, particularly when there is a variable load carried by ship. Another group is a result of external perturbations including deterministic and time-varying stochastic disturbances, which are induced by the wind, wave and ocean currents. To address these two kinds of perturbations simultaneously a robust-adaptive control approach should be adopted. A brief literature review of related solutions in the marine robotics control area is provided below.

The main author of this paper proposed a perturbation compensating algorithm addressing accurate path tracking of a marine autonomous vehicle in the presence of environmental disturbances—a robust Model Predictive Control (MPC) [1]. In [2], the author presented a robust strategy based on Time-Delay Control (TDC) algorithm and Terminal Sliding Mode (TSM) designed to control an Underwater Vehicle-Manipulator System (UVMS). In this work, TDC was used to estimate some perturbations using a Time-Delay Estimation (TDE) part. In addition, SMC was adopted for withstanding other perturbations, which cannot be estimated by TDC. As for other robust control strategies, which were adopted to address trajectory tracking control of marine robots, some researches upon the underwater robotics were done in [3–6]. A robust and adaptive controller based on the H-infinity approach was proposed in [7], where some of ship's parameters were unknown. A related research conducted in [8] by the main author of this paper proposed to use an approximate dynamic programming to optimally adjust the control gains in a super-twisting sliding mode controller for a Maritime Autonomous Surface Ship (MASS). A nonlinear MPC was used for an Unmanned Surface Vehicle (USV) in [9]; however, a conventional MPC was used there and it could not guarantee sufficient robustness. In addition, the environmental disturbances in ocean or sea have not been considered there. In [10], a backstepping control approach combined with sliding mode control algorithm was designed to address the trajectory tracking problem of an underactuated USV. An autonomous robotic boat was presented in [11], and for its controller, a nonlinear model predictive control (NMPC) was adopted. However, the uncertainty effect, resulting from the changes in inertia, was not taken into account there. Another weakness of this work was neglecting the influences due to deterministic and stochastic disturbances. In [12], a nonlinear disturbance observer was designed for an autonomous vessel to estimate unknown parameters and external disturbances. However, this observer was able to estimate only constant disturbances. A control law for path tracking of marine autonomous vessel was adopted in [13], where a singular perturbation method was used to decompose the system into two Lyapunov-based control subsystems. For tuning of

gains in a PID controller of USV, a self-regulator PID was designed in [14] in which coefficients have been regulated by the fuzzy rules.

In general, despite significant progress, none of the above-mentioned nonlinear conventional robust control algorithms managed to successfully address internal uncertainties in autonomous ships and time-varying stochastic disturbances upon them with sufficient accuracy. To successfully deal with these perturbations, other algorithms have been proposed as adaptive robust approaches. For this case, a disturbance observer was proposed in [15] for the trajectory tracking of Unmanned Marine Vehicles (UMVs). Researchers in this work adopted an adaptive law to estimate and compensate the disturbance observer error. Following this, by combining a nonlinear disturbance observer, dynamic surface control and adaptive robust backstepping together, they developed a dynamic surface adaptive robust controller. In another robust-adaptive approach [16], a nonlinear tracking differentiator combined with a reduced-order extended state observer (ESO) was adopted to estimate external disturbances and uncertainties with a small estimation error. Another interesting work has been accomplished using the finite-time extended state observer-based distributed formation control for marine surface vehicles with input saturation and external disturbances. The time-varying external disturbances induced by winds, waves, and ocean currents are considered there in the computer simulation [17]. Authors in [18] proposed an MPC approach combined with an adaptive line-of-sight (LOS) guidance for path following of Autonomous Surface Vehicles (ASVs). However, the conventional MPC algorithm is not a robust algorithm inherently. Hence, the robustness analysis should have discussed in aforementioned research. By combining the adaptive continuous sliding mode control with backstepping technique, an adaptive-robust control algorithm was adopted in [19]. Another backstepping-based control algorithm was presented in [20], in which a sliding mode disturbance observer is designed and applied to the path-following system of a container ship to estimate the time-varying environmental disturbances. Concerning some artificial intelligence-based control algorithms for the autonomous marine platforms, an adaptive neural network (NN) controller for fine trajectory tracking of surface vessels with uncertain environmental disturbances was presented in [21]. A novel robust-adaptive formation control scheme based on the Minimal Learning Parameter (MLP) algorithm and the Disturbance Observer (DOB) was presented in [22]. Using the MLP algorithm, a remarkable descent in tunable parameters for the controller and DOB has been achieved, which in turn led to great reduction of the online computational time.

An artificial intelligence-based robust control strategy in the form of a multi-layer neural network was combined with adaptive robust techniques in [23]. This resulted in

the control system of great robustness against uncertain nonlinearities and environmental disturbances. The proposed approach efficiently compensated both parametric and non-parametric uncertainties including time-varying disturbances induced by waves and ocean currents.

In this study, we propose an adaptive-robust strategy for path following control of an autonomous ship under deterministic and stochastic disturbances induced by winds, waves and ocean currents. This strategy includes a learning-based algorithm, which is based on the Actor-Critic Approximate Dynamic Programming (ACADP), as an adaptive part and a TDC control algorithm as a robust part. Furthermore, the proposed control algorithm brings an optimal characteristic, which results from solving the Hamilton–Jacobi–Bellman (HJB) equation using Approximate Dynamic Programming (ADP) procedure, for the conventional robust TDC algorithm.

The rest of this paper is organized as follows: an autonomous ship dynamical model associated with all of the time-varying external disturbances including deterministic and stochastic parts as well as uncertainties is described in the next section. The third section is dedicated to describe a time-delay control algorithm. In Sect. 4, the proposed hybrid Actor-Critic Time-Delay Control (AC-TDC) as an adaptive-robust approach is explained. In Sect. 5, the computer simulation results and comparison of them are shown and discussed to confirm high performance of the proposed hybrid control. Finally, the conclusion of this research is provided in Sect. 6.

2 Nonlinear dynamics of autonomous ships

In the control systems designed for the maneuvering control of ships, it is common to consider a 3 DOFs model as a coupled surge–sway–yaw model and thus neglect heave, roll and pitch motions. Therefore, the nonlinear dynamics of an autonomous ship can be described in form of 3 DOFs as follows:

$$\begin{cases} M\dot{\vartheta} + C(\vartheta)\vartheta + D(\vartheta)\vartheta = U_c + MR^T(\psi)\Gamma \\ \eta = R(\psi)\vartheta \end{cases} \quad (1)$$

where velocity and position vectors are defined as $\vartheta(t) = [u(t), v(t), r(t)]^T$ and $\eta(t) = [x(t), y(t), \psi(t)]^T$, respectively. $u(t)$ and $v(t)$ are speed signals of autonomous ship along the x- and y-axes, respectively. In addition, $r(t)$ is an angular speed of ship around the y-axis. The respective frames of surge, sway and yaw motions are shown in Fig. 1. In the aforementioned equation, the rotation matrix
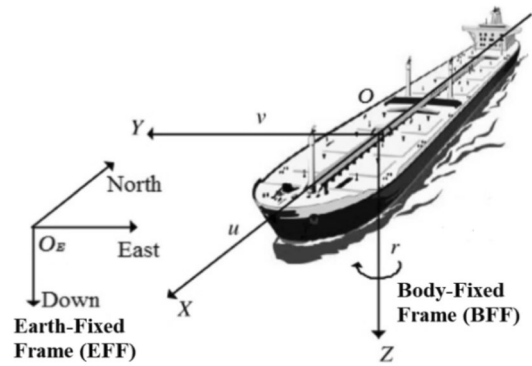

Fig. 1 Autonomous ship with three degrees of freedom [31]

$R(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$ is adopted to transfer coordinates from the body-fixed frame to the inertial frame.

For presenting Eq. (1) in a standard form based on position vector, we can rewrite it using the property of the rotation matrix $\dot{\eta} = R(\psi)\vartheta$ as

$$M(\eta)\ddot{\eta} + C(\eta, \dot{\eta})\dot{\eta} + D(\eta, \dot{\eta})\dot{\eta} = U_c + \Gamma \quad (2)$$

where all the related components of each matrix in Eq. (2) were presented in [24] analogous to the dynamics of robotic manipulators. $U_c = [F_x, F_y, \tau_\psi]^T$ and Γ are control inputs and external disturbances including deterministic and stochastic disturbances, respectively. These time-varying stochastic disturbances are induced by the winds, waves and ocean currents, which are shown by the vector of $\Gamma = [\Gamma_u, \Gamma_v, \Gamma_r]^T$. In this research, we consider these disturbances for applying upon the autonomous ship with 3 DOFs, in which each term of Γ_u, Γ_v and Γ_r consists of the constant deterministic components and the time-varying stochastic parts [25]. Therefore, we can rewrite them as follows:

$$\begin{cases} \Gamma_u = \bar{\Gamma}_u + \tilde{\Gamma}_u \\ \Gamma_v = \bar{\Gamma}_v + \tilde{\Gamma}_v \\ \Gamma_r = \bar{\Gamma}_r + \tilde{\Gamma}_r \end{cases} \quad (3)$$

where $\bar{*}$ and $\tilde{*}$ denote deterministic and stochastic components of external disturbances, respectively. Regarding stochastic components, a Wiener process is used to model them as follows [25]:

$$\begin{cases} \tilde{\Gamma}_u = \Delta_1(t)\dot{w}_1 \\ \tilde{\Gamma}_v = \Delta_2(t)\dot{w}_2 \\ \tilde{\Gamma}_r = \Delta_3(t)\dot{w}_3 \end{cases} \quad (4)$$

where Δ_1, Δ_2, and Δ_3 denote time-varying covariance and w_i is a Wiener process:

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \in \mathbb{R}^3, M(\eta) \in \mathbb{R}^3,$$

where $C(\eta, \dot{\eta}) \in \mathbb{R}^3$, $D(\eta, \dot{\eta}) \in \mathbb{R}^3$ are constant inertia matrix, time-varying inertia matrix, time-varying Coriolis matrix, and time-varying hydrodynamic damping matrix including both linear and nonlinear parts, respectively. Indeed, with taking into account the nonlinear damping parts, the matrix $D(\eta)$ is changed to the form $D(\eta, \dot{\eta})$ and nonlinear coefficients (d_{ij}) are dependent on velocity vector $\dot{\eta} = R(\psi)\vartheta$.

2.1 Time-delay control

Let us assume that the desired paths, which must be followed by the autonomous ship, are denoted by the vector η_d. A control strategy of the autonomous ship path following is the adopted to provide a position vector η for following the η_d. Let us define the tracking error and its first and second derivatives as

$$e(t) = \eta_d(t) - \eta(t) \tag{5}$$

$$\dot{e}(t) = \dot{\eta}_d(t) - \dot{\eta}(t) \tag{6}$$

$$\ddot{e}(t) = \ddot{\eta}_d(t) - \ddot{\eta}(t) \tag{7}$$

We regard parameters of e_s, e_{sw} and e_y as components of the tracking error vector $e(t)$ for an autonomous ship, which are surge, sway and yaw tracking errors, respectively:

$$e_s(t) = x_d(t) - x(t) \tag{8}$$

$$e_{sw}(t) = y_d(t) - y(t) \tag{9}$$

$$e_y(t) = \psi_d(t) - \psi(t) \tag{10}$$

where η, η_d are supposed to be measurable. In Eq. (2) from the modeling section let $M(\eta)$ be approximated with $\bar{M}$ as a constant matrix. Then, Eq. (2) can be rewritten as

$$\bar{M} \ddot{\eta} + N(\eta, \dot{\eta}) + H(\eta, \dot{\eta}, \ddot{\eta}) = U_c \tag{11}$$

where nominal term $N(\eta, \dot{\eta})$ and perturbation term $H(\eta, \dot{\eta}, \ddot{\eta})$ including uncertainties C_h, D_h and external deterministic and time-varying stochastic disturbances Γ are expressed as follows:

$$\begin{cases} N(\eta, \dot{\eta}) = C_n(\eta, \dot{\eta})\dot{\eta} + D_n(\eta, \dot{\eta})\dot{\eta} \\ \\ H(\eta, \dot{\eta}, \ddot{\eta}) = \left[M(\eta) - \bar{M} \right] \ddot{\eta} + \dots \\ \quad \dots C_h(\eta, \dot{\eta})\dot{\eta} + D_h(\eta, \dot{\eta})\dot{\eta} - \Gamma \end{cases} \tag{12}$$

Following the control scheme of the Computed Torque Control (CTC) method, the control reads as

$$U_c = \bar{M} u_{CTC} + \tilde{H}(\eta, \dot{\eta}, \ddot{\eta}) \tag{13}$$

$$u_{CTC} = \ddot{\eta}_d(t) + k_d \dot{e} + k_p e \tag{14}$$

where $\tilde{H}$, k_d and k_p are the estimated value of perturbation of H using Time-Delay Estimation (TDE) part, and the derivative and the proportional gain matrices, respectively. Substituting Eqs. (13) and (14) into Eq. (11), final error dynamics is expressed as

$$\ddot{e}(t) + k_d \dot{e}(t) + k_p e(t) = 0 \tag{15}$$

In the TDE part of TDC, under the assumption that the time delay (L) is very small and equal to the simulation step size ($L = \Delta t$), $H(t)$ can be approximated as

$$\tilde{H}(t) \cong H_{t-\Delta t} = U_c(t - \Delta t) - \bar{M} \ddot{\eta}(t - \Delta t) \tag{16}$$

Combining Eqs. (13), (14) and (16), the Time-Delay Control (TDC) law is obtained as

$$u_{TDC} = u_{TDC}(t - \Delta t) - \bar{M} \ddot{\eta}(t - \Delta t) + \bar{M}(\ddot{\eta}_d(t) + k_d \dot{e} + k_p e) \tag{17}$$

In addition, the second derivative for the delayed positions is implemented using the following equation [2]:

$$\ddot{\eta}(t - L) = \frac{\eta(t) - 2\eta(t - \Delta t) + \eta(t - 2\Delta t)}{\Delta t^2} \tag{18}$$

To guarantee that the control system is stable, the constant matrix $\bar{M}$ should be adopted as follows:

$$\bar{M} = \gamma I, 0 < \gamma < 2\Theta \tag{19}$$

where Θ is the minimum bound of eigenvalues λ_i of $M(\eta)$ for all η.

2.2 Actor-critic TDC and approximate dynamic programming

Although the induced perturbations can be partially estimated by the TDE utility in Eq. (16), this estimation will be inaccurate and bring in large estimation errors when the autonomous ship is encountered with the deterministic and time-varying stochastic disturbances with high amplitudes.

Furthermore, if we improperly choose the constant components of the matrix M, the estimation error of TDC will increase. To address this problem in TDC algorithm, we propose a backpropagation policy algorithm based on approximate dynamic programming and actor-critic learning for compensating of TDE error, particularly when M has not been truly tuned.

Let us define TDE error as

$$\varepsilon = \bar{M}^{-1}\left[\tilde{H} - H\right] \tag{20}$$

Substituting Eqs. (16) into (17) and considering the above equation, Eq. (17) is rewritten as follows:

$$U_c = \tilde{H} + \bar{M}\left(\ddot{\eta}_d(t) + k_d \dot{e} + k_p e\right) +$$

$$\bar{M}^{-1}\left[\tilde{H} - H\right] \tag{21}$$

Then, we adopt an optimal guaranteed cost control part u_{AC} to incessantly compensate TDE errors and increase convergence speed.

Let us define u_{AC} as a compensator signal as

$$u_{AC} = \tilde{H} - H \tag{22}$$

where it is optimally generated using an approximate dynamic programming algorithm based on a propagation actor-critic learning approach.

Therefore, combining the TDC technique with optimal control theory, a robust TDC scheme is proposed for path following control of autonomous ships with an inherent uncertain dynamics and under deterministic and time-varying stochastic disturbances. According to Eqs. (16), (21) and (22), the hybrid AC-TDC is designed as

$$u_{AC-TDC} = u_{TDC} + \bar{M}^{-1}u_{AC} \tag{23}$$

where u_{TDC} is a part of the proposed control law, which is used as a main part of the proposed robust strategy and u_{AC} is an optimal guaranteed cost control part. The latter is used to partially compensate time-delay estimation error and attenuate stochastic disturbance effects.

As for the actor-critic network (which will be described in the next section), its reward function and the weights are updated when the autonomous ship encounters deterministic and stochastic disturbances that cannot be suppressed solely by TDC (Inaccurate Matrix M).

The learning-based part of the proposed control approach can be brought to designing an optimal control strategy. It is responsible for solving the HJB equations using Actor-Critic Approximate Dynamic Programming

(ACADP) as a Policy Iteration (PI) algorithm. Figure 2 shows the scheme of this hybrid control approach.

2.3 ACADP strategy

The proposed ACADP algorithm is based on a Heuristic Dynamic Programming (HDP), which is analogous to the formulation of reinforcement learning (RL) using the Temporal Difference (TD) learning methods [26–32].

Let us define a continuous utility or reward-to-go function as follows:

$$r(t) = g_r(t)pg_r^T(t) \tag{24}$$

where the state-action vector $g_r(t)$ is regarded:

$$g_r(t) = \left[\, q^T(t) \; u_{AC}^T(t) \,\right],$$

and $u_{AC}(t) = \left[\, u_{AC}^{surge} \; u_{AC}^{sway} \; u_{AC}^{yaw} \,\right]^T$

is the control law as a compensator part, which is delivered by an actor-critic approximate dynamic programming approach; $\omega_1(t) = \left[\, e_s(t) \; \dot{e}_s(t) \; e_{sw}(t) \; \dot{e}_{sw}(t) \; e_y(t) \; \dot{e}_y(t) \,\right]$, p is a positive-definite diagonal weighting matrix with a proper dimensions. $q(t)$ is applied to the backpropagation actor neural network as an input vector in the input layer and its components are as

$$q(t) = \left[\, \omega_1(t - \Delta t) \; \omega_1(t) \,\right]^T \tag{25}$$

where Δt is simulation step size for implementing of ACADP iteration algorithm. We adopt a total cost-to-go function as
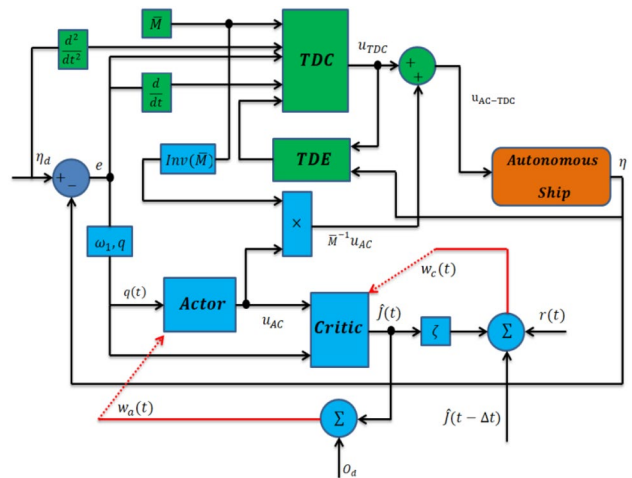
$$J\big(q(t), u_{AC}(t)\big)$$

Fig. 2 Hybrid control scheme AC-TDC

$$= \sum_{i=0}^{\infty} \zeta r\big((q(t + i\Delta t), u_{AC}(t + i\Delta t)\big) \qquad (26)$$

where $r\big(q(t), u_{AC}(t)\big)$ is a positive-definite function if $q(t) \neq 0$ and $u_{AC}(t) \neq 0$ and just equal to zero when $q(t) = 0$ and $u_{AC}(t) = 0$. In above equation, ζ, $0 < \zeta < 1$ is a discount factor. We adopt $J^*(q(t))$ as a solution of (26). The result of minimization of the mentioned cost-to-go function, which satisfies HJB equation as an optimal solution, can be expressed as

$$J^*(q(t)) = \min_{u_{AC}(t)} \big\{ r\big(q(t), u_{AC}(t)\big) + \zeta J^*(q(t + \Delta t)) \big\} \qquad (27)$$

For solving Eq. (27), a policy iteration as an iterative method of reinforcement learning is used. In this research, the ACADP approach is used as a policy iteration to approximately solve HJB equation. In other words, $J^*(q(t))$ is approximated by $\widehat{J}(q(t))$, which is the output of the critic neural network. It also means that if estimation error of the TDE part (ε) is largely compensated and tracking errors including $[e_s(t), \dot{e}_s(t), e_{sw}(t), \dot{e}_{sw}(t), e_y(t),$ and $\dot{e}_y(t)]$ all converge to zero, the ACADP control policy $u_{AC}(t) = 0$, $J^*(q(t)) = \widehat{J}(q(t)) = 0$, and three positions of autonomous ship absolutely track their desired paths with an acceptable deviation in presence of deterministic and stochastic disturbances. In ACADP structure, the critic neural network is trained to appraise the performance of the current control policy, which is composited of actor control policy and TDC control policy.

In our design, the proposed control signal (AC-TDC) is generated by the sum of the TDC and the ACADP control laws. The outputs of the action network (u_{AC}) are multiplied by the inverse of constant matrix ($\overline{M}$) and then adaptively compensate perturbation effects, while output of TDC controller (u_{TDC}) has a large estimation error and is not capable to overcome these stochastic disturbances. The related propagation algorithm of the ACADP strategy is implemented using MATLAB programming environment.

2.4 Critic neural network structure

This network is a function of $(q(t), u_{AC}(t), w_c(t))$, where $w_c(t)$ is the critic weight vector and the control law $u_{AC}(t)$ generated by the actor neural network is also the input to the critic neural network [33]. Therefore, critic's input and output vectors read as

$$\begin{cases} c_i(t) = \big[q(t)^T \; u_{AC}(t)^T \big]^T \\ c_o(t) = \widehat{J}(t) \end{cases} \qquad (28)$$

Using standard algebra, one can derive from Eq. (26): $\zeta J(t) + r(t) - J(t - \Delta t) = 0$, where $J(t) = J\big(q(t), u_{AC}(t)\big)$. Then, we can define an evaluation factor as an error function for the backpropagation critic neural network as follows:

$$e_c(t) = \zeta \widehat{J}(t) - [\widehat{J}(t - \Delta t) - r(t)] \qquad (29)$$

Regarding weights updating of the critic network, an objective function is minimized in the following equation:

$$\min_{w_c(t)} E_c(t) = \min_{w_c(t)} \frac{1}{2} e_c^T(t) e_c(t) \qquad (30)$$

Let us consider N_{ci} and N_{ch} as number of inputs and number of neurons in hidden layer of critic network, respectively. $N_{ci} = n + m$, n is number of inputs in actor network and m is number of outputs in actor network. We adopt a hyperbolic tangent threshold activation function for both actor and critic networks as follows:

$$\sigma(t) = \frac{1 - e^{-t}}{1 + e^{-t}} \qquad (31)$$

where $\sigma_c(t) = \sigma_a(t) = \sigma(t)$.

Let us define w_{c1} and w_{c2} as weights of the critic network for its input layer and its output layer, respectively. In addition, two new variables $\varnothing_{cj}(t)$ and $\theta_{cj}(t)$ for the jth neuron in hidden layer are expressed as

$$\varnothing_{cj}(t) = \sum_{i=1}^{n} q_i(t) w_{c1,ij}(t) + \sum_{i=1}^{m} u_{ACi}(t) w_{c1,ij}(t), j = 1, \ldots, N_{ch} \qquad (32)$$

$$\theta_{cj}(t) = \sigma_c\big(\varnothing_{cj}(t)\big) = \frac{1 - e^{-\varnothing_{cj}(t)}}{1 + e^{-\varnothing_{cj}(t)}}, j = 1, \ldots, N_{ch} \qquad (33)$$

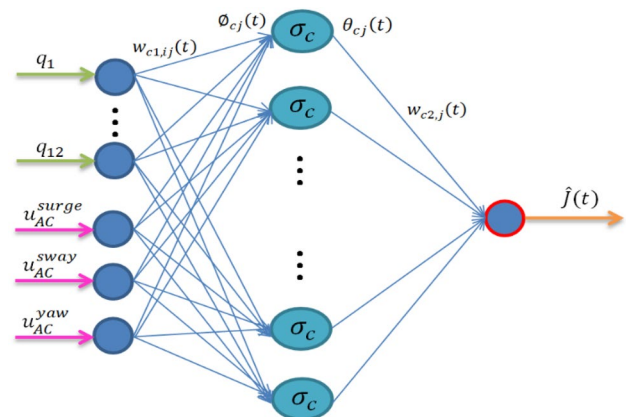$$\widehat{J}(t) = \sum_{j=1}^{N_{ch}} w_{c2,j}(t) \theta_{cj}(t) \qquad (34)$$

Fig. 3 Critic neural network

We use a gradient-based adaption algorithm for updating the weights of the actor and critic neural networks. Therefore, the weights of the critic network can be calculated using this algorithm as follows:

$$\Delta w_{c2,j}(t) = -\rho_c(t)\frac{\partial E_c(t)}{\partial \widehat{J}(t)} \cdot \frac{\partial \widehat{J}(t)}{\partial w_{c2,j}(t)} \tag{35}$$

$$\Delta w_{c1,ij}(t) = -\rho_c(t)\frac{\partial E_c(t)}{\partial \widehat{J}(t)} \cdot \frac{\partial \widehat{J}(t)}{\partial \theta_{cj}(t)} \cdot \frac{\partial \theta_{cj}(t)}{\partial \varnothing_{cj}(t)} \cdot \frac{\partial \varnothing_{cj}(t)}{\partial w_{c1,ij}(t)} \tag{36}$$

$$w_{c2,j}(t + \Delta t) = w_{c2,j}(t) + \Delta w_{c2,j}(t) \tag{37}$$

$$w_{c1,ij}(t + \Delta t) = w_{c1,ij}(t) + \Delta w_{c1,ij}(t) \tag{38}$$

where $\rho_c(t)$ is an adaptive learning rate of the critic neural network.

Suppose that the weights of the input-to-hidden layer are randomly initialized and retained fixed for both the action network ($w_{a1,ij}$) and the critic network ($w_{c1,ij}$). Based on the universal approximation theorem of neural networks, if the number of hidden-layer neurons is ample enough, then the approximation error can be regarded small. Therefore, we choose the updating strategy for the critic output weights (w_c) as the normalized gradient descent algorithm as in [29]:

us label w_{a1} and w_{a2} as weights of the actor network for its input layer and its output layer, respectively. Then, the output of actor network $u_{ACk}(t)$ can be computed as follows:

$$\varnothing_{aj}(t) = \sum_{i=1}^{N_{ai}} q_i(t)w_{a1,ij}(t), j = 1, \dots, N_{ah} \tag{41}$$

$$\theta_{aj}(t) = \sigma_a(\varnothing_{aj}(t)) = \frac{1 - e^{-\varnothing_{aj}(t)}}{1 + e^{-\varnothing_{aj}(t)}}, j = 1, \dots, N_{ah} \tag{42}$$

$$\gamma_{ak}(t) = \sum_{j=1}^{N_{ah}} w_{a2,jk}(t)\theta_{aj}(t), k = 1, \dots, N_{ao} \tag{43}$$

$$u_{ACk}(t) = \sigma_a(\gamma_{ak}(t)) = \frac{1 - e^{-\gamma_{ak}(t)}}{1 + e^{-\gamma_{ak}(t)}}, k = 1, \dots, N_{ao} \tag{44}$$

The action neural network is trained using an iterative evaluation loop, which appraises the error between the desired objective (O_d) and the approximated cost-to-go function provided by the critic network. Regarding desired objective, O_d is the objective value of the total cost-to-go and is usually set to 0:

$$e_a(t) = \widehat{J}(t) - O_d \tag{45}$$

$$w_c(t + \Delta t) = w_c(t) - \rho_c(t)\sigma_c(t)\left[\zeta w_c^T(t)\rho_c(t) + r(t) - w_c^T(t - \Delta t)\sigma_c(t - \Delta t)\right]^T \tag{39}$$

A critic neural network is depicted in Fig. 3, which consists of 3 layers, 15 inputs $c_i(t) = \begin{bmatrix} q(t)^T & u_{AC}(t)^T \end{bmatrix}^T$ in input layer, 12 neurons in hidden layer and 1 output $c_o(t) = \widehat{J}(t)$ in output layer. As the simulation studies will show, this simple network structure turned out to result in a sufficient accuracy combined with a short update time.

2.5 Actor neural network structure

An actor neural network is a function of $(q(t), w_a(t))$, whose input and output vectors are defined as follows:

$$\begin{cases} a_i(t) = q(t) \\ a_o(t) = u_{AC}(t) \end{cases} \tag{40}$$

Let us define N_{ai}, N_{ah} and N_{ao} as number of inputs, number of neurons in hidden layer and number of outputs of an actor network, respectively. $u_{ACk}, k = 1 \dots N_{ao}$. Analogically to the last section (procedure for the critic network design), we adopt three variables $\varnothing_{aj}(t)$, $\theta_{aj}(t)$ and $\gamma_{ak}(t)$. Let

We can express an objective function for the actor neural network as

$$\min_{w_a(t)} E_a(t) = \min_{w_a(t)} \frac{1}{2} e_a^T(t) e_a(t) \tag{46}$$

Following this, we then use a gradient-based adaption approach to update the actor weights adaptively. It is done similarly as for critic network, so we do not repeat this procedure description and the accompanying equations here. Based on mentioned assumption for the critic network in which the weights of the input-to-hidden layer are randomly initialized and retained fixed for both networks, the output weights of the actor neural network are obtained as follows:

$$w_a(t + \Delta t) = w_a(t) - \rho_a(t)\sigma_a(t)[w_c^T(t)\alpha(t)][w_c^T(t)\sigma_c(t)]^T \tag{47}$$

where $\alpha(t)$ is a matrix with dimension $N_{ch} \times m$ and its elements can be calculated as

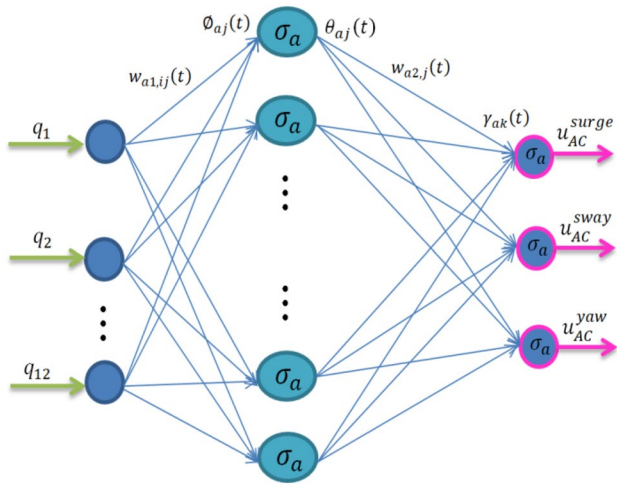$$\alpha_{jk}(t) = \frac{1}{2}\left(1 - \sigma_{cj}^2(t)\right)w_{cj,n+k} \tag{48}$$

Fig. 4 Actor neural network

$$, \begin{cases} j = 1, \dots, N_{ch} \\ k = 1, \dots, m \end{cases}$$

An actor neural network is shown in Fig. 4, which consists of 3 layers, 12 inputs $a_i(t) = \begin{bmatrix} q_1(t - \Delta t) & q_1(t) \end{bmatrix}^T$ in input layer, 10 neurons in hidden layer and 3 outputs $a_o(t) = \begin{bmatrix} u_{AC}^{surge} & u_{AC}^{sway} & u_{AC}^{yaw} \end{bmatrix}^T$ in output layer.

2.6 Concept of uniformly ultimately bounded

Typically, the presence of bounded deterministic and stochastic disturbances as well as neural network approximation errors lead to a Uniformly Ultimately Bounded (UUB) [32]. The UUB is used in the stability analysis of dynamical systems—particularly those that are based on approximate dynamic programming algorithm. The stability property is developed to guarantee that each of the iterative control laws can make the cost-to-go index function, adaptive weights and tracking errors uniformly ultimately bounded. Based on Eqs. (39) and (47), a dynamical system of estimation errors reads as

$$\tilde{w}_{a,c}(t + \Delta t) = \tilde{w}_{a,c}(t) - \qquad (49)$$

Table 2 Setting of the neural networks

Parameter	Actor Network	Critic Network
Number of Learning Iterations	$n_a = 70$	$n_c = 100$
Initial Learning Rates	$\rho_a = 0.2$	$\rho_c = 0.2$
Number of Hidden Neurons	$N_{ah} = 10$	$N_{ch} = 12$
Number of input Neurons	12	15
Number of output Neurons	3	1
Tolerance Errors	$T_a = 10^{-6}$	$T_c = 10^{-6}$
Discount Factor	−	$\zeta = 0.92$

Table 1 Parameters for Cybership II

m	23.8	X_u	-0.72	Y_r	-7.2
I_z	1.76	$X_{\|u\|u}$	-1.32	$Y_{\|v\|r}$	-0.84
x_g	0.046	X_{uuu}	-5.86	$Y_{\|r\|r}$	-3.45
$X_{\dot{u}}$	-2	Y_v	-0.89	$N_{\|r\|v}$	0.1
$Y_{\dot{v}}$	-10	$Y_{\|v\|v}$	-36.4	N_r	-1.9
$Y_{\dot{r}}$	0	N_v	0.03	$N_{\|v\|r}$	0.08
$N_{\dot{v}}$	0	$N_{\|v\|v}$	3.95	$N_{\|r\|r}$	-0.75
$N_{\dot{r}}$	-1	$Y_{\|r\|v}$	-0.8		

$$f(w_{a,c}(t - \Delta t), w_{a,c}(t), \sigma_{a,c}(t - \Delta t), \sigma_{a,c}(t))$$

where $\tilde{w}_{a,c}(t) = w_{a,c} - w_{a,c}^*$. The optimal weights of critic and actor networks are shown by w_c^* and w_a^*, respectively. All the weights and activation functions are assumed to be bounded as

$$\begin{cases} \|w_c(t)\| \leq w_{cm}(t) \\ \|w_c^*(t)\| \leq w_{cm}(t) \\ \|w_a(t)\| \leq w_{am}(t) \\ \|w_a^*(t)\| \leq w_{am}(t) \\ \|\sigma_c(t)\| \leq \sigma_{cm}(t) \\ \|\sigma_a(t)\| \leq \sigma_{am}(t) \end{cases} \qquad (50)$$

Definition 1. A dynamical system is said to be UUB with ultimate bound $b > 0$, if for any $a > 0$ and $t_0 > 0$, there exists a positive number $N = N(a, b)$ independent of t_0, such that $\| \tilde{w}(t)\| \leq b$ for all $t \geq N + t_0$ whenever $\| \tilde{w}(t_0)\| \leq a$.

Theorem 1. *If, for system* (49), *there exists a function* $L(\tilde{w}(t), t)$, *such that for all* $\tilde{w}(t_0)$ *in a compact set*
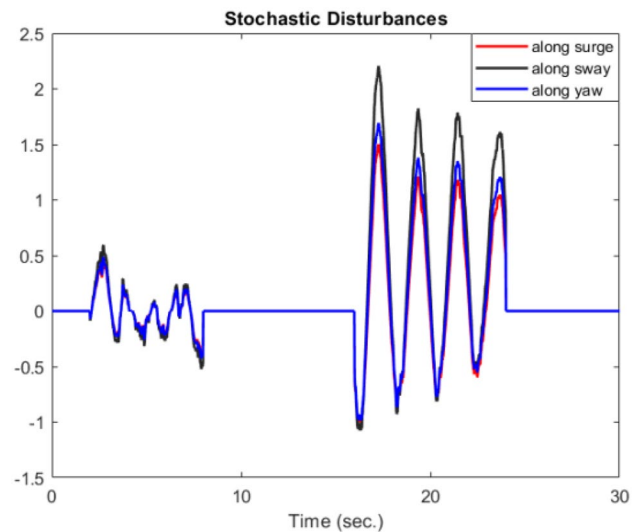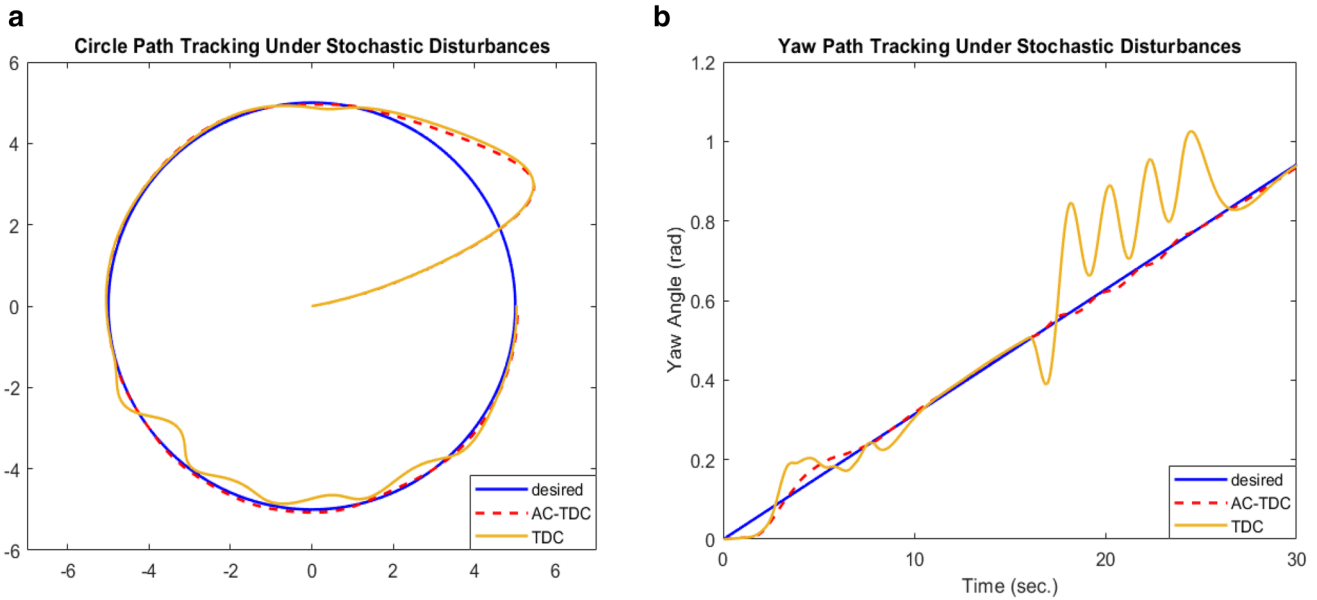

Fig. 5 Stochastic disturbances

a

b

Fig. 6 Path following of autonomous ship under stochastic disturbances

K, L(w̃ (t), t) is positive definite and the first difference, L(w̃ (t), t) < 0 for ‖ w̃ (t₀)‖ > b, for some b > 0, such that b -neighborhood of w̃ (t) is contained in K, then the system is UUB and the norm of state is bounded to within a neighborhood of b.

Based on the above theorem and definition, a Lyapunov function can be adopted as follows:

$$\begin{cases} L(t) = L_1(t) + L_2(t) \\ L_1(t) = \left(\frac{1}{\rho_c}\right) tr(\tilde{w}_c^T \tilde{w}_c) \\ L_2(t) = \left(\frac{1}{\rho_a}\right) tr\left(\tilde{w}_a^T \tilde{w}_a\right) \end{cases} \quad (51)$$

With applying the above mentioned theorem and Lyapunov function, presented analysis in [29] showed that errors between the optimal weights of the actor and critic neural networks w_a^*, w_c^* and their estimates w_a, w_c, which are approximated using the ACADP algorithm, can be all UUB. The UUB concept is obviously realized in the depicted results of our proposed hybrid algorithm, which are presented in the next section (Table 1).

3 Computer simulations and discussion

In this section, the proposed AC-TDC control algorithm is tasted on a small-scale autonomous surface ship (Cybership II) with the following mass-related and identified

parameters, which were calculated using the system identification procedure in [36]:

In the simulation, we consider all the components of matrices $M(\eta), C(\eta, \dot{\eta}), D(\eta, \dot{\eta})$ given in [24]. Damping matrix $D(\eta, \dot{\eta})$ includes some nonlinear damping coefficients, which are described as follows [17]:

$$d_{11} = 0.72 + 1.33|u| + 5.87u^2$$

$$d_{22} = 0.8896 + 36.5|v| + 0.805|r| \quad (52)$$

$$d_{23} = 7.25 + 0.8451|v| + 3.45|r|$$

$$d_{32} = 0.0313 + 3.96|v| + 0.13|r|$$

$$d_{33} = 1.9 - 0.08|v| + 0.75|r|$$

Other damping coefficients in the damping matrix 3×3 are assigned to zero. In addition, the constants in the inertia matrix are assigned as follows:

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} = \begin{bmatrix} 25.8 & 0 & 0 \\ 0 & 33.8 & 1.0115 \\ 0 & 1.0115 & 2.76 \end{bmatrix}$$

Furthermore, disturbances are considered for applying upon the Cybership II, in which their respective vectors $\mathbf{\Gamma} = [\mathbf{\Gamma}_u, \mathbf{\Gamma}_v, \mathbf{\Gamma}_r]^T$ in Eq. (3) can be expressed as follows [17]:
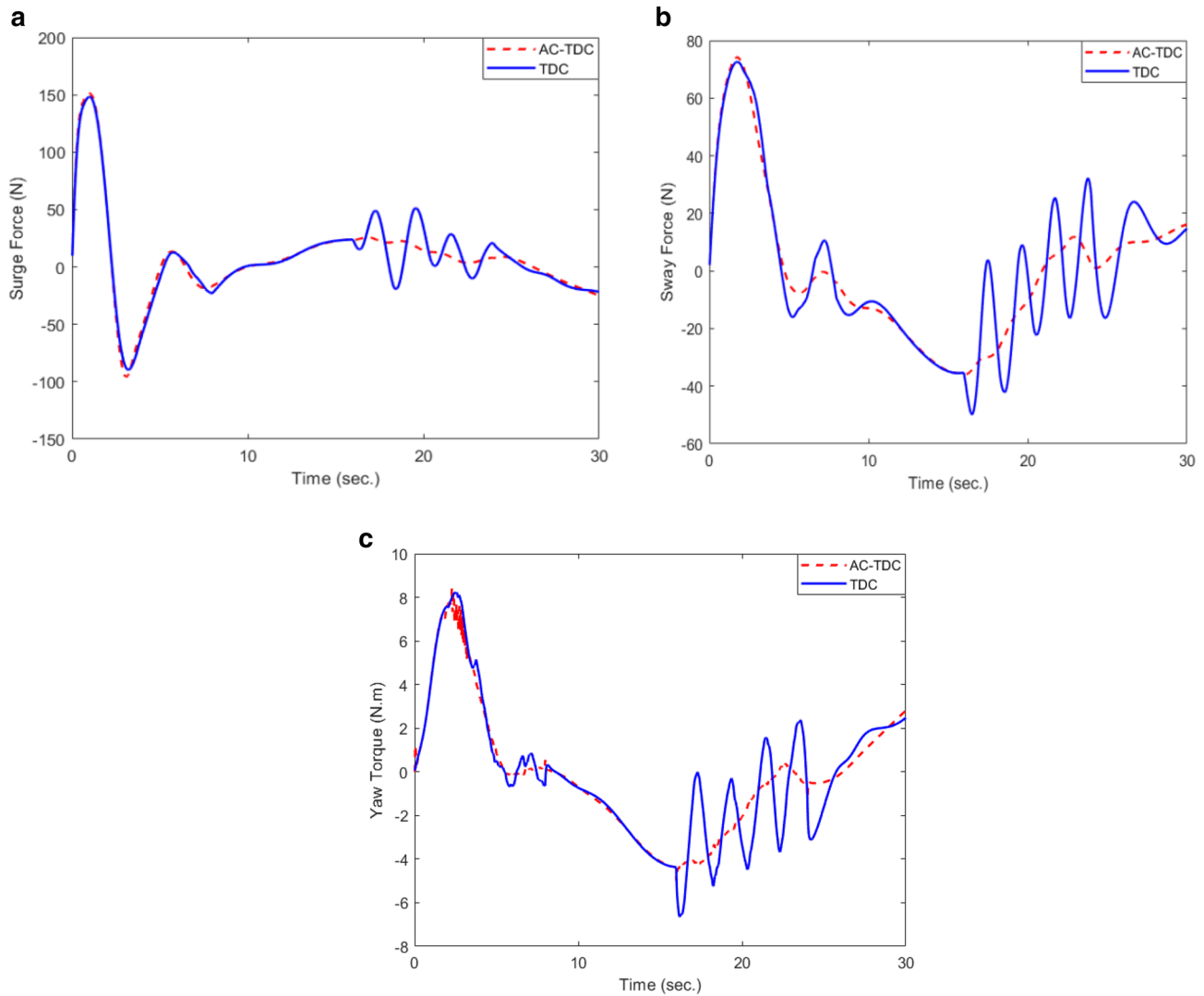
a

b

c

Fig. 7 Control input signals

$$\begin{cases} \Gamma_u = 0.1v^3 + 0.06u + 0.01\sin(t) \\ \Gamma_v = ur + 0.1u + 0.01\sin(t) \\ \Gamma_r = 0.4ur + v^2 + 0.01\sin(t) \end{cases} \qquad (53)$$

In both case studies, we regard the following setting of the coefficients in Table 2 for the critic and actor neural networks:

The weighting matrix in Eq. (24) is initialized by $p = 0.1I_{nn}$, where I_{nn} is an identity matrix with dimension 15×15 (The state-action dimension is 15). In the proposed algorithm, we consider an adaptive step size for both neural networks, which are decreased by 0.01 in each of the 5 steps $\Delta t = 0.02$. The weights of the critic and actor networks are randomly initialized with values from the range $[-0.2, 0.2]$.

4 Case study 1:

4.1 Tracking of desired paths under stochastic disturbances

The desired paths from which must be followed by the autonomous ship under stochastic disturbances are given by (54). The disturbances themselves are shown in Fig. 5. The simulation time $t = 30sec.$ is adopted for this tracking:

$$\begin{cases} x = 5cos(\frac{1}{15}\pi t) \\ y = 5sin(\frac{1}{15}\pi t) \\ \psi = 0.01\pi t \end{cases} \qquad (54)$$

Regarding parameters of TDC, in the first case study we set them as follows:
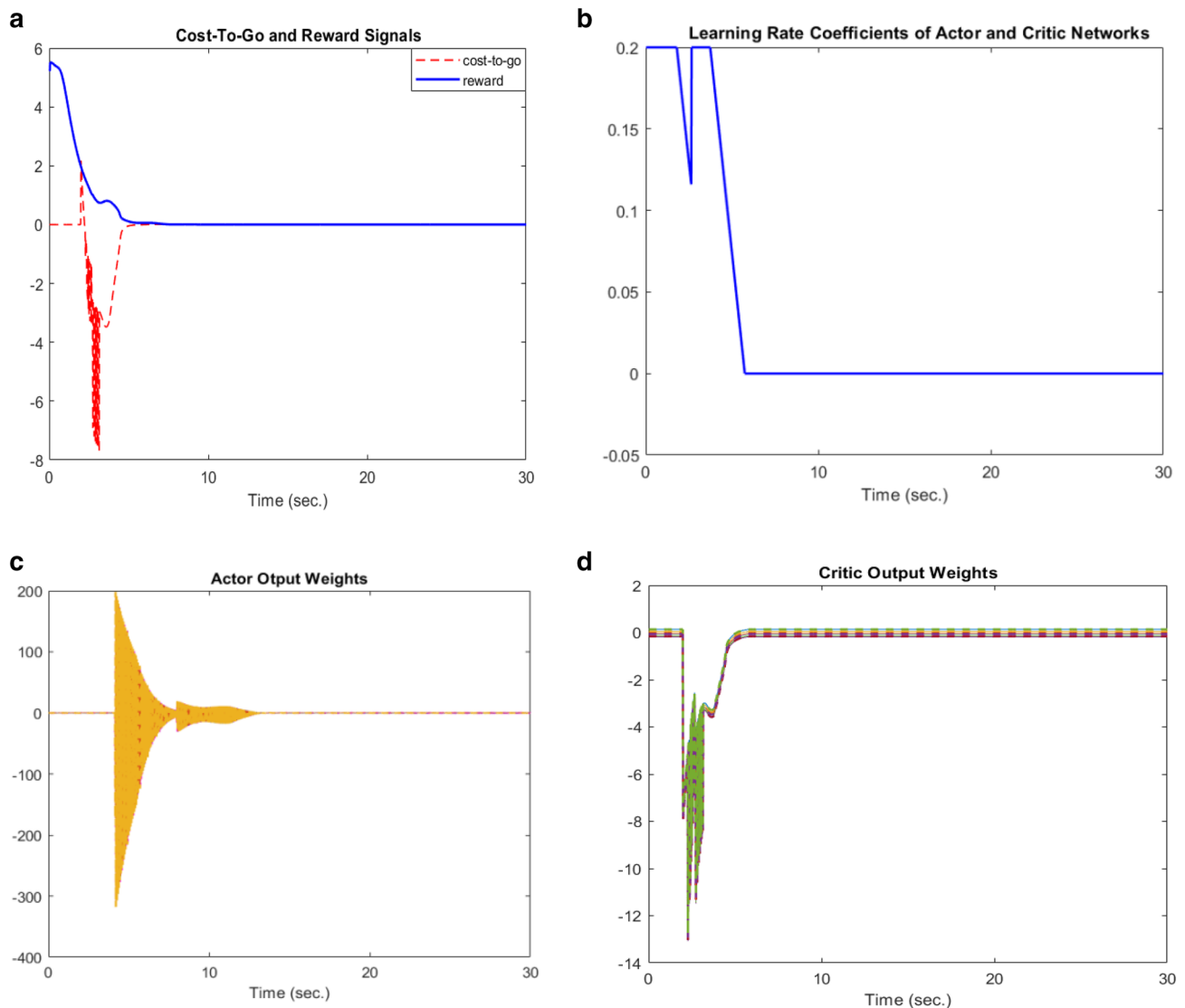
Fig. 8 Adaptive weights and parameters of actor-critic learning algorithm

$(\bar{M} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} I_{nn}, n = 3), k_d = k_p = 2$ where I_{nn} is an identity matrix.

In this case study, we have applied only stochastic disturbances in two stages during the tracking. The results of path following under this condition are shown in Fig. 6, where it is observed that the proposed AC-TDC algorithm is capable of handling disturbances significantly better than standard TDC algorithm. The deviations of TDC from the given path are particularly visible in the bottom part of Fig. 6 and upper part of Fig. 6b, where the TDC algorithm clearly has problems with handling the increased

amplitude of stochastic disturbance. In comparison, AC-TDC was able to deal with that and avoid those deviations in both cases (circle path tracking and yaw path tracking).

The control input signals are depicted in Fig.7, which are in an optimal effort status for the hybrid algorithm toward the conventional TDC. Concerning performance of actor-critic neural network, respective results for the implemented algorithm of ACADP are illustrated in Fig. 8.

To clearly observe the responses of network weights, we provide some figures with zoom in view depicted in Figs. 9 and dedicated to the actor and critic neural
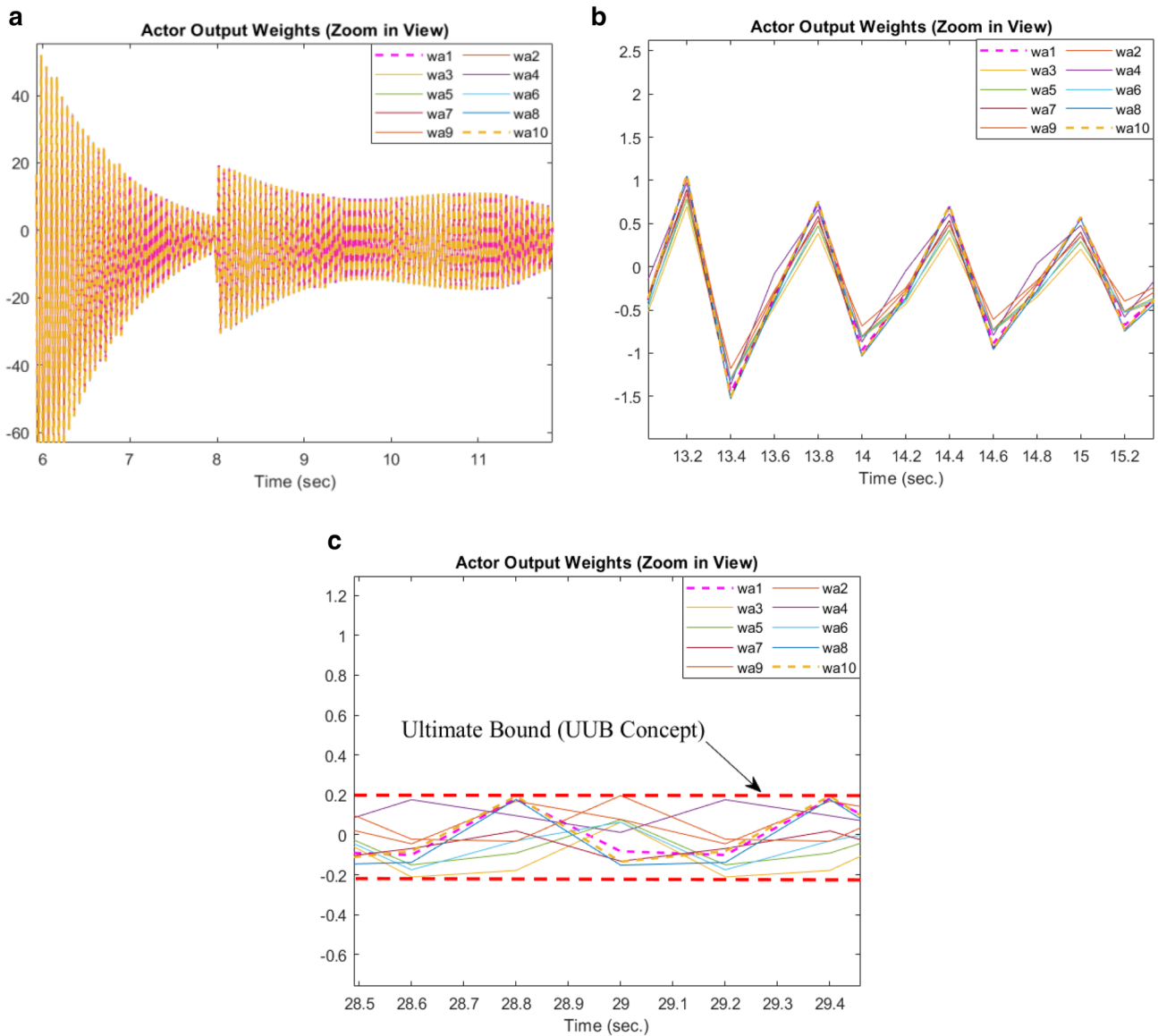
Fig. 9 Zoom in view of actor output adaptive weights

networks, respectively. As it is observed in Fig. 9c, we have uniformly ultimately bounded convergence for the output weights of actor neural network. It means that the actor weights, after their learning, are kept in a variable mode but with bounded amplitudes. Therefore, this variable mode of actor weights leads to the suppression of unexpected stochastic disturbances. Similarly, an exponential convergence is shown in Fig. 10b, c for the output weights of critic neural network.

5 Case study 2

5.1 Tracking of desired paths under deterministic and stochastic disturbances

The desired paths, which must be followed by autonomous ship under stochastic disturbances, are given by (55). As for disturbances, they are shown in Fig. 11. The simulation time $t = 30\,sec.$ is adopted for this tracking.

Fig. 10 Zoom in view of critic output adaptive weights
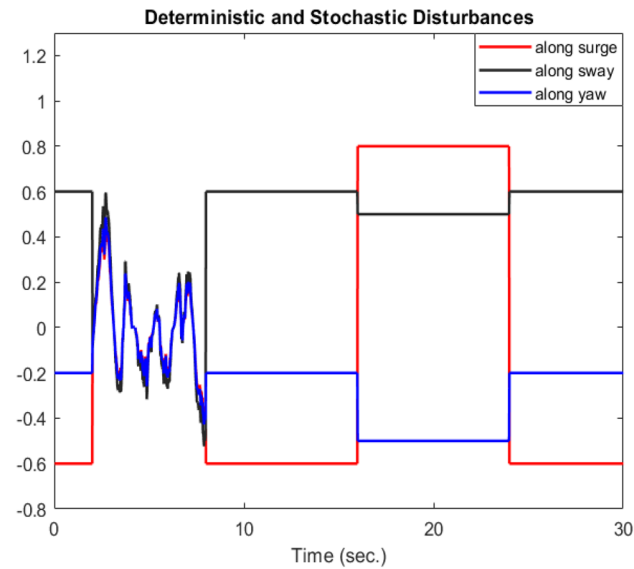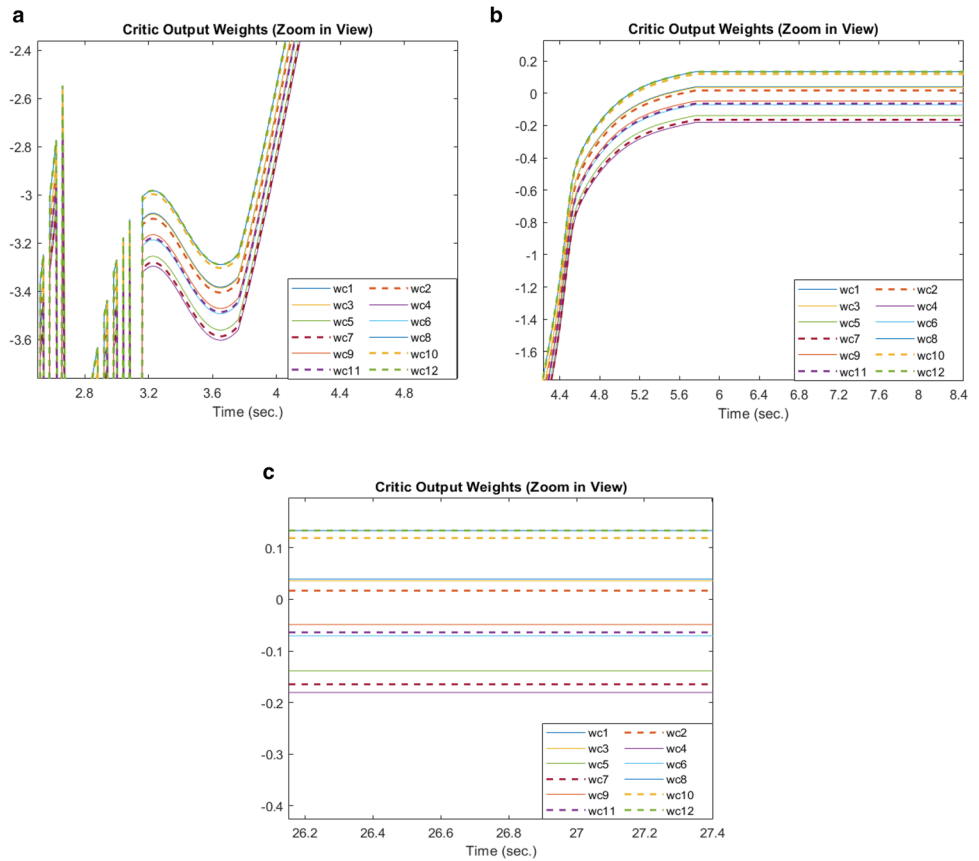
Fig. 11 Deterministic and stochastic disturbances

$$\begin{cases} x = 3(m) \\ y = 5sin(\frac{1}{15}\pi t) \\ \psi = 0.01\pi t \end{cases} \tag{53}$$

Regarding parameters of TDC in the second case study, we set them as follows:

$(M = \begin{bmatrix} 0.5 & 0.4 & 0.6 \end{bmatrix} I_{nn}, n = 3)$, $k_d = k_p = 2$, where I_{nn} is an identity matrix.

In the second case study, we have applied both kinds of disturbances upon the autonomous ships including deterministic and time-varying stochastic disturbances. As shown in Fig. 11, the deterministic disturbances are applied four times, whereas there is for a single 30-s phase of applying the stochastic disturbance. The results of path following under this highly disturbed condition are shown in Fig. 12. It can be observed that the conventional TDC is unable to keep the yaw motion of autonomous ship on the reference values, which is depicted in Fig. 12c. In contrast to the TDC,
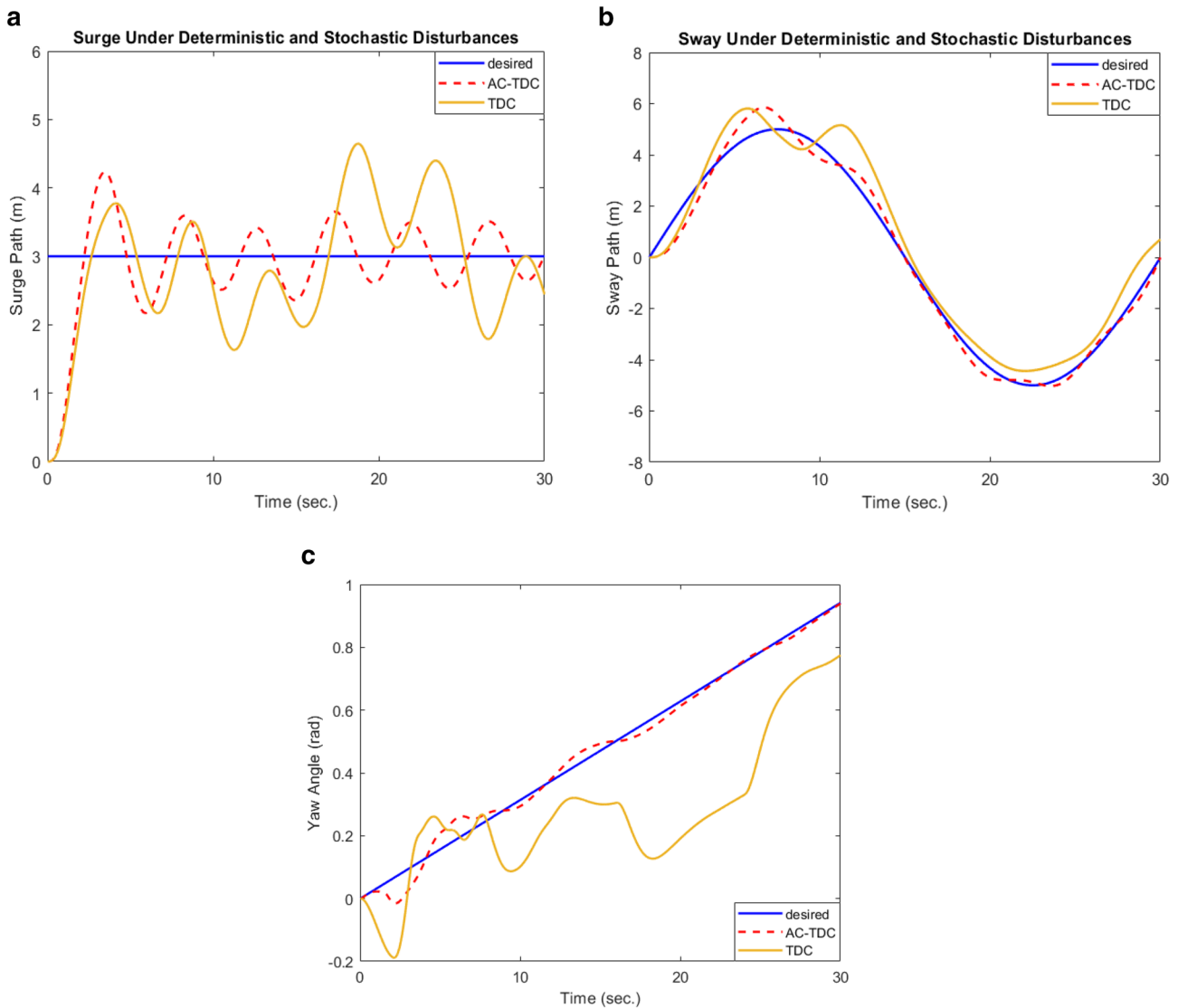
Fig. 12 Path following of autonomous ship under deterministic and stochastic disturbances

we have a high-performance responses of the AC-TDC in following the desired paths of surge, sway, and yaw motion with much lower deviations in comparison to the TDC.

The optimal control efforts are depicted in Fig.13. Concerning performance of actor-critic neural network, respective results for the implemented algorithm of ACADP are illustrated in Fig. 14 and some zoom-in-view figures of output weights of actor and critic networks are shown in Figs. 15 and 16.

In this case study, we have chosen smaller values for the components of constant inertia matrix M toward the first case study. It leads to an increasing in TDE error. Indeed, with applying the deterministic and stochastic disturbances simultaneously, the perturbation magnitude is increased. Therefore, for overcoming this compelled perturbation, we have to increase the values of M's components in the TDC

algorithm. Otherwise, with a decrease in values of $\bar{M}$'s components, the error of estimating the perturbation by the TDE part is increased. In such a situation (improper values of M), the TDC algorithm is no longer capable to follow the path accurately. For instance it is observed in Fig. 12c. For tackling with this problem due to the improper assigning of values to the inertia matrix M, we have proposed a supplementary control algorithm combined with the TDC. In this proposed AC-TDC control scheme, the TDE error is compensated using an online learning-based part in the proposed control algorithm ACADP. This algorithm generates optimal efforts using an actor neural network as supplementary control inputs in which the output weights of this network are instantly updated, in case of occurring the deterministic disturbances, as shown in Fig. 14d. As it has been depicted
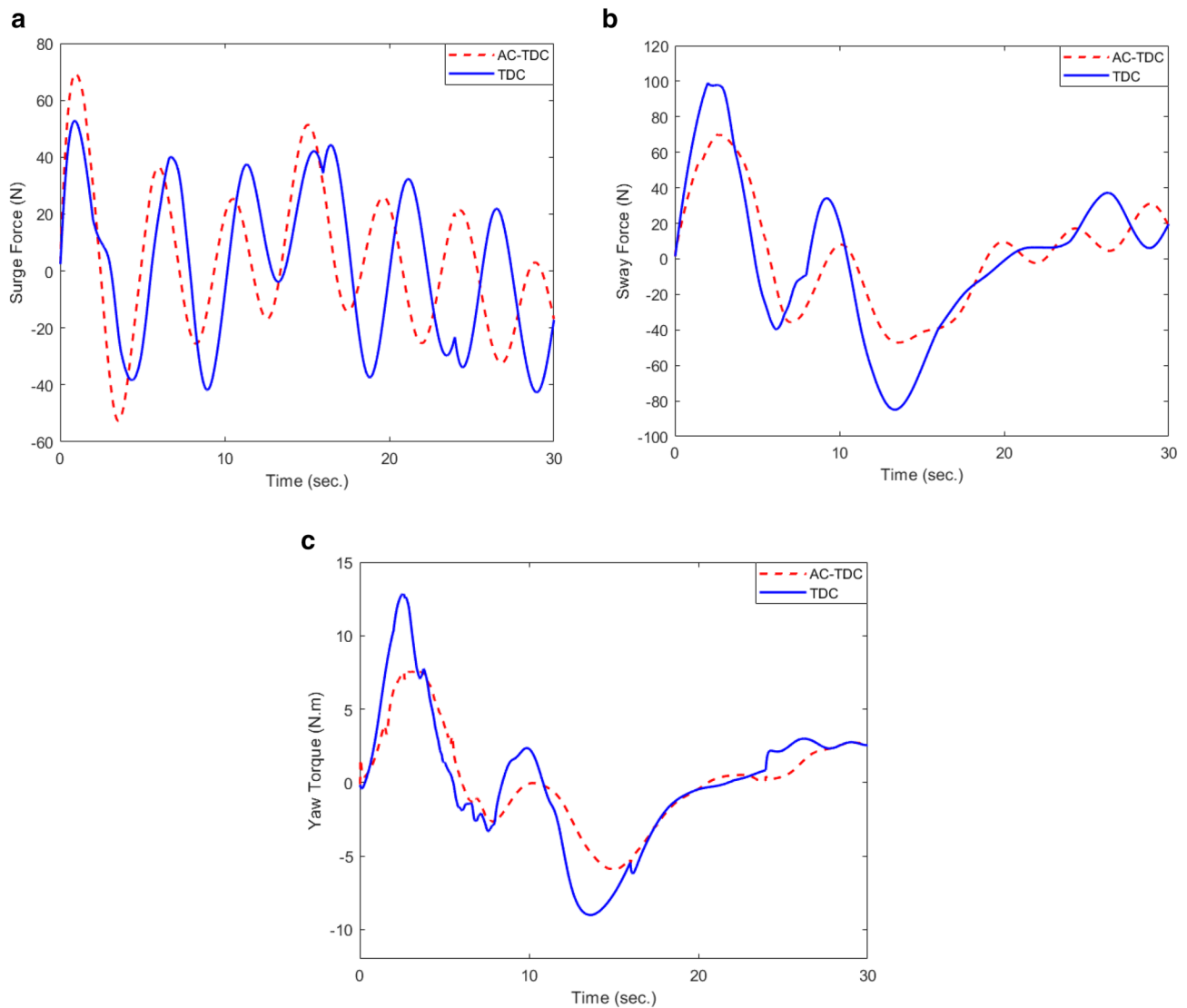
Fig. 13 Control input signals

in Fig. 14c, with occurring each new deterministic disturbance during of path following, a new learning process is commenced to assign proper weights to the actor network.

Concerning the UUB convergence, the output weights of the actor neural network move toward a bounded variable mode, analogically to the first case study. It is shown in Fig. 15c.

6 Conclusion

In this paper, a hybrid robust-adaptive optimal control structure was developed to address the path following control problem for autonomous ships under deterministic and stochastic disturbances. In the proposed structure, a TDC robust algorithm is combined with an ADP-based algorithm as an adaptive part of the proposed AC-TDC algorithm. This part includes an actor-critic neural network, which is used as a main subsystem to implement an ADP algorithm. Based on the capability of the heuristic ADP procedure in solving the HJB equation with a curse of dimensionality, a supplementary optimal control law is a result of this cost-to-go approximation procedure. This optimal control law was added to the TDC law to address high-performance path following control algorithm for the autonomous ships, particularly when the conventional TDC is unable to track the desired paths in case of facing the deterministic and stochastic disturbances, simultaneously. The robustness of new algorithm has been tested in presence of above-mentioned perturbations for two different case studies. The simulation results depicted that this procedure has significant properties in trajectory tracking with acceptable precision for autonomous ships under deterministic
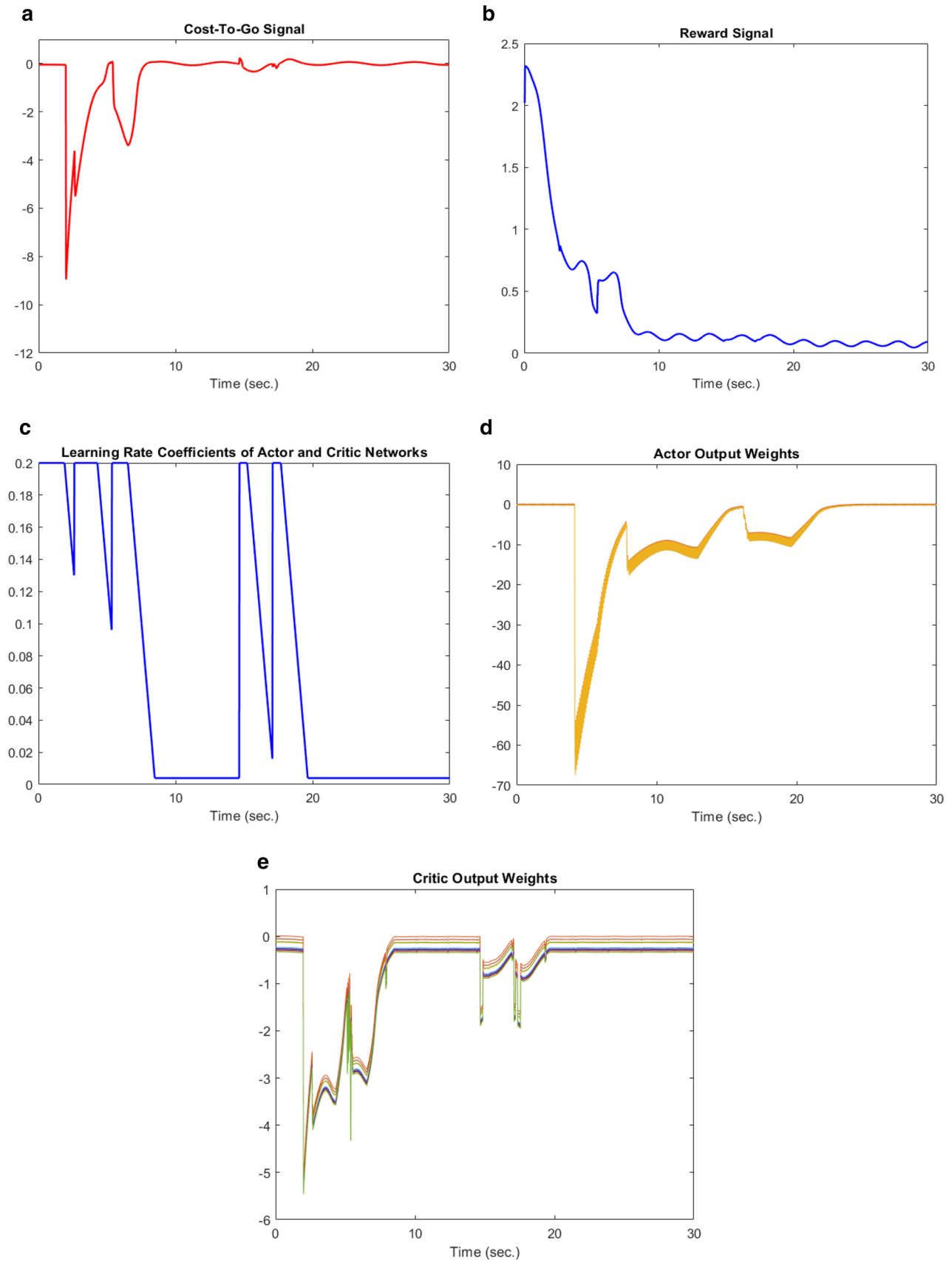
a

b

c

d

e

Fig. 14 Adaptive weights and parameters of actor-critic learning algorithm

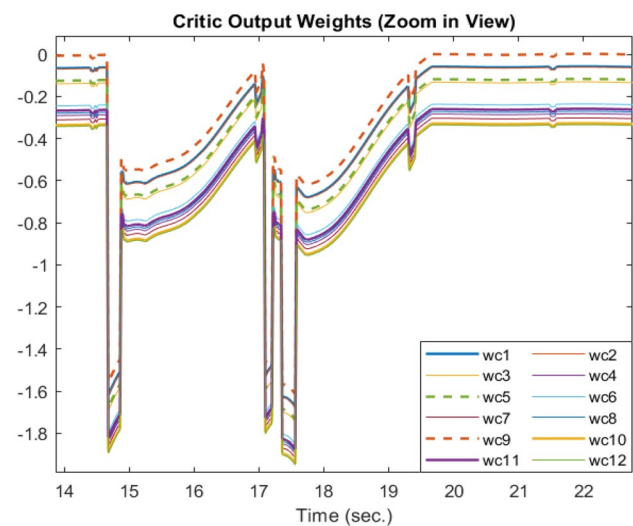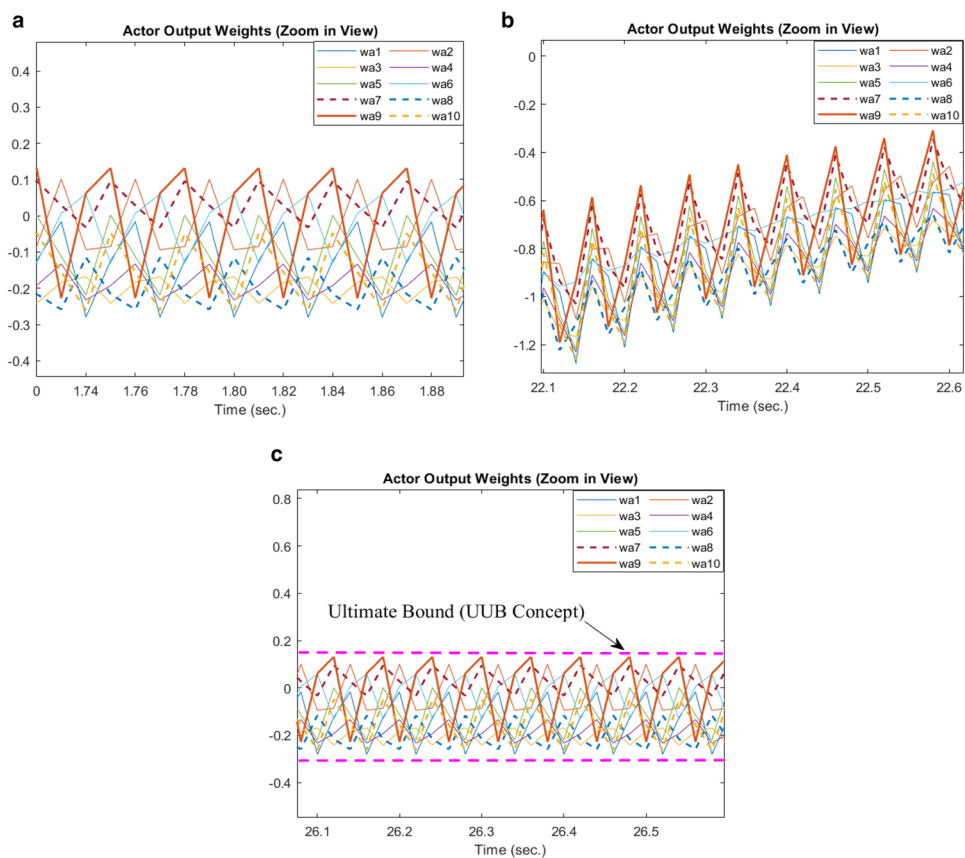**Fig. 15** Zoom in view of actor output adaptive weights

Fig. 16 Zoom in view of critic output adaptive weights

and stochastic disturbances. Concerning the future works, the next step is to implement the presented algorithm upon an underactuated autonomous surface ship equipped with a rudder navigation system. Moreover, the TDC part of the hybrid control algorithm will be enhanced by two structures. First, a Super-Twisting Sliding Mode Control (ST-SMC) strategy will be added to fulfil even greater robustness. Second, the method will feature optimal gain tuning of ST-SMC by means of the ADP part of the proposed control approach.

References

1. Esfahani HN (2019) Robust model predictive control for autonomous underwater vehicle-manipulator system with fuzzy compensator. Polish Maritime Res (forthcoming). https://doi.org/10.2478/pomr-2019-00139
2. Esfahani HN, Azimirad V, Danesh M (2015) A time delay controller included terminal sliding mode and fuzzy gain tuning for underwater vehicle-manipulator systems. Ocean Eng 107:97–107

3. Esfahani HN, Azimirad V, Eslami A, Asadi S (2013) An optimal sliding mode control based on immune-wavelet algorithm for underwater robotic manipulator. In: Proceedings of the 21st Iranian Conference on Electrical Engineering (ICEE), Mashhad, Iran

4. Esfahani HN, Azimirad V, Zakeri M (2014) Sliding Mode-PID Fuzzy controller with a new reaching mode for underwater robotic manipulators. Lat Am Appl Res 44(3):253–258

5. You S-S, Lim T-W, Kim J-Y, Choi H-S (2012) Dynamics and robust control of underwater vehicles for dpth trajectory. J Eng Maritime Envion 227(2):107–113

6. Lakhekar GV, Waghmare LM (2017) Robust manevering of autonomous underwater vehicle: an adaptive fuzzy PI sliding mode control. Intel Serv Robot 10(3):195–212

7. Zwierzewicz Z (2020) Robust and adaptive path-following control of an underactuated ship. IEEE Access 8:120198–120207. https://doi.org/10.1109/ACCESS.2020.3004928

8. Esfahani HN, Szlapczynski R, Ghaemi H (2019) High performance super-twisting sliding mode control for a maritime autonomous surface ship (MASS) using ADP-Based adaptive gains and time delay estimation. Ocean Eng 191:106526

9. Liu C, Zheng H, Negenborn RR, Chu X, Wang L (2015) Trajectory tracking control for underactuated surface vessels based on nonlinear Model Predictive Control. In: Corman F., Voß S., Negenborn R. (eds) Computational Logistics. ICCL 2015. Lecture Notes in Computer Science, vol 9335, pp. 166–180. Springer, Cham. (Proceedings of the 6th International Conference, ICCL 2015, Delft, The Netherlands)

10. Liu J, Luo J, Cui J, Peng Y (2016) Trajectory Tracking Control of Underactuated USV with Model Perturbation and External Interference. In: Procedings of the 3rd International Conference on Mechanics and Mechatronics Research (ICMMR 2016) Chongqing, China

11. Wang W, Mateos LA, Park S, Leoni P, Gheneti B, Duarte F, Ratti C, Rus D (2018) Design , Modeling , and Nonlinear Model Predictive Tracking Control of a Novel Autonomous Surface Vehicle. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 6189–6196. Brisbane, Australia

12. Abdelaal M, Fr M, Hahn A (2018) Nonlinear Model Predictive Control for trajectory tracking and collision avoidance of underactuated vessels with disturbances. Ocean Eng 160:168–180

13. Yi B, Qiao L, Zhang W (2016) Two-time scale path following of underactuated marine surface vessels : Design and stability analysis using singular perturbation methods. Ocean Eng 124:287–297

14. Jamalzade MS, Koofigar HR, Ataei M (2016) Adaptive fuzzy control for a class of constrained nonlinear systems with application to a surface vessel. J Theor Appl Mech 54(3):987–1000

15. Zhang P (2018) Dynamic surface adaptive robust control of unmanned marine. J Robot 2018:1–6

16. Huang H, Gong M, Zhuang Y, Sharma S, Xu D (2019) A new guidance law for trajectory tracking of an underactuated unmanned surface vehicle with parameter perturbations. Ocean Eng 175:217–222

17. Fu M, Yu L (2018) Finite-time extended state observer-based distributed formation control for marine surface vehicles with input saturation and disturbances. Ocean Eng 159:219–227

18. Liu C, R. Negenborn R, Chu X, Zheng H, (2018) Predictive path following based on adaptive line-of-sight for underactuated autonomous surface vessels. J Mar Sci Technol 23(3):483–494

19. Sun Z, Zhang G, Qiao L, Zhang W (2018) Robust adaptive trajectory tracking control of underactuated surface vessel in fields of marine practice. J Mar Sci Technol 23(4):950–957

20. Zhao Y, Dong L (2019) Robust path-following control of a container ship based on Serret-Frenet frame transformation. J Mar Sci Technol. https://doi.org/10.1007/s00773-019-00631-6

21. Li G, Li W, Hildre HP, Zhang H (2016) Online learning control of surface vessels for fine trajectory tracking. J Mar Sci Technol 21(2):251–260

22. Lu Y, Zhang G, Sun Z, Zhang W (2018) Robust adaptive formation control of underactuated autonomous surface vessels based on MLP and DOB. Nonlinear Dyn 94:503–519

23. Shojaei K (2016) Observer-based neural adaptive formation control of autonomous surface vessels with limited torque. Rob Auton Syst 78:83–96

24. Fang Y (2004) Global output feedback control of dynamically positioned surface vessels: an adaptive control approach. Mechatronics 14:341–356

25. Do KD (2016) Global robust adaptive path-tracking control of underactuated ships under stochastic disturbances. Ocean Eng 111:267–278

26. Barto A, Sutton R, Anderson, (1983) Neuronlike adaptive elements that can solve difficult learning control problems. IEEE Trans Syst Man Cybern 5:835–846

27. Sutton RS, Barto AG (1990) Time-derivative models of pavlovian reinforcement. In: Moore JW, Gabriel M (eds) Learning and computational neuroscience. MIT Press, Cambridge, pp 497–537

28. Si J, Yang L, Lu C.S, Sun J, Mei S (2009) Approximate dynamic programming for continuous state and control problems. In: 17th Mediterranean Conference on Control and Automation, pp 1415–1420

29. Si J, Yu-Tsung W (2001) On-line learning control by association and reinforcement. IEEE Trans Neural Networks 12(2):264–276

30. Guo W, Liu F, Si J, Mei S (2014) Online and model-free supplementary learning control based on approximate dynamic programming. In: 26th Chinese Control and Decision Conference, CCDC 2014:1316–1321

31. Sokolov Y, Kozma R, D. Werbos L, J. Werbos P, (2015) Complete stability analysis of a heuristic approximate dynamic programming control design. Automatica 59:9–8

32. Liu F, Sun J, Si J, Guo W, Mei S (2012) A boundedness result for the direct heuristic dynamic programming. Neural Netw 32:229–235

33. Szuster M, Gierlak P (2015) Approximate dynamic programming in tracking control of a robotic manipulator. Int J Adv Rob Syst 13(1):16

34. Bhasin S (2011) Reinforcement learning and optimal control methods for uncertainnonlinear systems. University of Florida, Ph.D.

35. Fossen TI (2011) Handbook of marine craft hydrodynamics and motion control. Wiley, Chichester

36. Skjetne R, Smogeli Ø, Fossen TI (2004) Modeling, identification, and adaptive maneuvering of CyberShip II: a complete design with experiments. IFAC Proceedings 37(10):203–208

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.