e-Navi

Available online at http://www.e-navigation.kr/
e-Navigation Journal

Original article

Adaptive control of ship heading based on DDPG algorithm

LI Weihong[a*], LI Xinyi[b], ZHAO Zijun[c], DING Shengda[d]

[a*]School of Navigation and Shipping, Shandong Jiaotong University, China, 15964513537@163.com

[b]School of Navigation and Shipping, Shandong Jiaotong University, China, 13015950887@163.com

[c]School of Navigation and Shipping, Shandong Jiaotong University, China, 1481725038@qq.com

[d]School of Navigation and Shipping, Shandong Jiaotong University, China, 1517026939@qq.com

Abstract

This paper investigates the problem of ship course control in the presence of model uncertainties, external disturbances, and actuator saturation. A high-performance autopilot is developed based on a direct neural network adaptive dynamic surface control (DSC) framework integrated with deep reinforcement learning. To compensate for lumped uncertainties arising from unmodeled dynamics and disturbances, a radial basis function (RBF) neural network is employed to provide online approximation within the control design. Moreover, the actuator saturation constraint is explicitly incorporated into the controller, avoiding performance degradation commonly encountered in conventional DSC schemes.To alleviate the reliance on manual parameter tuning, the controller parameter adaptation is formulated as a continuous-action optimization problem and solved using a deep deterministic policy gradient (DDPG) algorithm. The DDPG agent learns an optimal tuning policy by maximizing a reward function that penalizes course tracking errors, excessive control variations, and energy consumption. Simulation results demonstrate that the proposed method achieves improved tracking accuracy, smoother control inputs, and enhanced robustness under complex operating conditions, thereby validating the effectiveness of the DDPG-based adaptive tuning strategy for autonomous ship navigation.

Keywords: Ship Heading Control; Deep Deterministic Policy Gradient; Input Saturation; Dynamic Surface Control; Neural Network

Copyright ⓒ 2017, International Association of e-Navigation and Ocean Economy.
This article is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/3.0/).
Peer review under responsibility of Korea Advanced Institute for International Association of e-Navigation and Ocean Economy
https://doi.org/10.52820/j.enavi.2025.25.046

1. Introduction

As the maritime industry's demand for vessel automation and autonomous navigation continues to grow, ship heading control has emerged as a key research area for improving navigational safety, reducing energy consumption, and mitigating human operational error. In the 1920s, Anschutz and Sperry achieved pivotal breakthroughs in mechanical heading-control systems based on onboard gyrocompasses, establishing the engineering foundation of modern autopilot technology (2012). With the introduction of Proportional-Integral-Derivative (PID) control, heading-control accuracy improved substantially; however, robustness remained constrained by modeling uncertainties in ship dynamics (Sun, 2020). To address model mismatch and inherently nonlinear dynamics under complex sea states, adaptive autopilots grounded in adaptive control theory were subsequently developed (Du, 2020). Notably, the adaptive trajectory linearization control (TLC) algorithm proposed in demonstrates markedly superior heading-tracking performance compared with conventional PID and standard TLC methods, even in the presence of unmodeled dynamics and input saturation (Mu and Wang, 2020).

Liu et al. addressed actuator saturation constraints by designing a heading controller that integrates backstepping with Lyapunov-based stability analysis (2024). However, classical backstepping is often hindered by the well-known "explosion of complexity". To overcome this limitation, Liu S. and Zhang G. et al. (2022) introduced DSC and further reduced computational burden by incorporating fuzzy-logic strategies. In parallel, neural networks-owing to their multilayer nonlinear architectures-can automatically extract vessel-motion features and construct approximate models of nonlinear dynamics under wind–wave–current disturbances, thereby accelerating advances in marine control (Le, 2021). Xie et al. (2020) applied a DSC-based robust adaptive neural-network framework to nonlinear ship heading systems, while Li (2012) developed a direct adaptive neural-network heading controller that explicitly accounts for input saturation.

More recently, deep reinforcement learning (DRL) has been increasingly adopted to optimize control and planning for marine vehicles. Ma L. , Qi M. Z. et al. (2025) proposed a deep deterministic policy gradient (DDPG) driven optimization scheme for a sliding-mode controller in underactuated vessels. DRL has also been leveraged for higher-level autonomy: Ref. (Qu and Xie, 2024) introduced a trajectory-history-aware DRL approach to ship path planning, and Ref. (Zhang, 2025) proposed a DRL-based method for nonlinear roll stabilization.

Several studies further advance DRL-enabled heading control toward engineering realism. Gao X. and Hu X. et al. (2025) combined a finite-time disturbance observer with a fuzzy system for composite disturbance rejection, introduced reinforcement-learning-based optimal compensation on the backstepping error surface, and explicitly handled rudder angle saturation-thus aligning heading-tracking design with practical constraints. For a single outboard-propelled unmanned surface vessel (USV), Cui B. and Chen Y. et al. (2024) proposed a hybrid architecture in which DDPG provides guidance while model predictive control (MPC) executes control; full-scale experiments, benchmarked against Adaptive Line-of-Sight (ALOS)–PID, showed substantial reductions in both cross-track and heading-angle errors and improved robustness to wind–wave–current disturbances. Sivaraj S. and Dubey A. et al. (1997) conducted a systematic comparison of DDPG, Twin Delayed Deep Deterministic Policy Gradient (TD3), Proximal Policy Optimization (PPO), and SAC for trajectory tracking and heading control (KRISO Very Large Crude Carrier 2 case study), concluding that Soft Actor-Critic (SAC) yields smoother control actions, where as DDPG achieves smaller cross-track errors. Wang, X. et al. (2024) proposed an improved DDPG-based PID tuning strategy for unmanned surface vehicle trajectory tracking, demonstrating enhanced robustness and tracking accuracy under environmental disturbances compared with conventional PID control. Finally, Emphasizing sample efficiency, Greep J. and Bayezit A. B. et al. (2025) investigated heading-keeping in open water with waves, comparing multiple reinforcement-learning training strategies and demonstrating stable steering even under limited data regimes.

Collectively, these advances indicate a clear shift in recent years toward deep reinforcement learning–based control paradigms. Where the DDPG algorithm (Ou, 2024) has emerged as a representative approach, capable of iteratively refining control policies through sustained interaction with the environment to enhance both performance and robustness. This capability is particularly compelling for marine systems, where dynamic uncertainty and strong nonlinearities are intrinsic and often coupled. Consequently, DDPG-driven adaptive heading control constitutes a promising direction with substantial practical potential for next-generation ship autopilot systems.

Here we propose a DDPG-based adaptive heading-control strategy for ships. The method integrates deep reinforcement learning with a direct adaptive DSC framework to explicitly address core challenges in heading regulation, including actuator (rudder) saturation, external disturbances, and model uncertainty. First, starting from a nonlinear vessel dynamics model, we construct an adaptive DSC architecture that enforces input-saturation constraints and employs an online radial basis function (RBF) neural network to approximate lumped uncertainties

in real time. Second, we recast controller-parameter tuning as a continuous-action optimization problem: through interaction with the closed-loop control environment, the DDPG agent learns and refines the control parameters by maximizing a task-specific reward function that encodes heading-tracking performance and robustness.

Simulation studies demonstrate that the proposed approach delivers strong dynamic performance and robustness under challenging operating conditions, yielding improved heading-tracking accuracy and closed-loop stability. These results offer a new solution paradigm for autonomous ship-navigation systems and highlight the substantial potential of DDPG for optimizing adaptive-controller parameters. More broadly, the study provides a forward-looking pathway for advancing next-generation ship heading control.

2. Problem Analysis

2.1. Mathematical Model of Ship Heading Control System

This study develops a second-order response model based on the Nomoto model, incorporating the specific requirements and dynamic characteristics of ship motion control. Taking into account the impact of external disturbances such as wind, waves, and currents, as well as the rudder angle input, we derive the state-space representation of the ship heading system. This formulation serves as the foundational theoretical framework for subsequent research into heading control strategies.

The ship heading system model, as proposed by Nomoto, is expressed as follows (Jia and Yang, 1997):

$$\ddot{\phi}+\frac{1}{T}H\left(\dot{\phi}\right)=\frac{K}{T}\delta \tag{1}$$

Let δ denote the rudder angle, ϕ the heading angle, K the turning index, and T the course-keeping index. Notably, $H\left(\dot{\phi}\right)$ is a nonlinear function of $\dot{\phi}$, which can be approximately expressed as follows:

$$H\left(\dot{\phi}\right)=a_1\dot{\phi}+a_2\dot{\phi}^3+a_3\dot{\phi}^5+\cdots \tag{2}$$

Where a_i and $i=1,2,3\cdots i$ are real-valued constants.

During algorithmic simulation, physical constraints can often be neglected; however, in real-world operations, the controller must strictly respect onboard actuator limits and navigational safety requirements. Specifically, the rudder command is subject to a feasible set $[-35^\circ,+35^\circ]$, where port rudder angles are defined as negative and starboard angles as positive. In addition, the yaw-rate state is bounded within an admissible interval $[-3^\circ/s,+3^\circ/s]$.

In light of the aforementioned constraints, this study incorporates input saturation limits into the controller design. During heading control, when the heading error is small, linearization techniques can be applied to approximate the control system, effectively simplifying nonlinear effects. However, in actual steering maneuvers, the nonlinearities induced by rudder angle variations cannot be neglected, and assuming them to be negligible is clearly insufficient for accurate modeling. Therefore, to establish a more universally applicable control framework, this research adopts a nonlinear model with broader applicability, better suited to meet the complexities of real-world operational requirements.

2.2. Intra-controller Compensation Auxiliary System

Due to the physical limitations of the steering gear, the rudder-angle command is inherently bounded. When a control signal exceeds this allowable range, the resulting saturation can degrade control performance and even compromise closed-loop stability. It is therefore essential to incorporate an auxiliary mechanism that compensates for these input constraints. By continuously monitoring and adjusting the control signal in real time, the auxiliary system modifies the commanded rudder input whenever saturation occurs, thereby preserving effective actuation, enhancing overall control performance, and improving the robustness and stability of the ship's heading-control system.

Consider the rudder angle limits for the inputs of maximum value ($u_{\max}$) and minimum value ($-u_{\min}$). The maximum amplitude of the ship's rudder angle is $35°$.

$$-u_{\min}\le u\le u_{\max} \tag{3}$$

$$u=sat\left(v\right)=\begin{cases}u_{\max}, & v>u_{\max}\\ v, & -u_{\min}\le v\le u_{\max}\\ -u_{\min}, & v<-u_{\min}\end{cases} \tag{4}$$

Where v represents the control input that the system needs to design.

Considering the influence of input saturation, the system is designed in the following form:

$$\dot{e}=\begin{cases}-c_{21}e-\dfrac{f\left(\cdot\right)}{e^2}e+\left(u-v\right), & |e|\ge\varepsilon\\ 0, & |e|<\varepsilon\end{cases} \tag{5}$$

Where e is an auxiliary system introduced to compensate for the error term caused by saturation, and $c_{21}>0$ is an adjustable parameter. And, $f\left(\cdot\right)=f\left(z_2,(u-v)\right)=\left|z_2\cdot b\cdot(u-v)\right|+\frac{1}{2}\left(u-v\right)^2\left(\Delta u=u-v\right)$. The undefined symbols will be introduced in Section 3.1.

2.3. RBF Neural Network

Artificial Neural Networks (ANNs) (Tian, 2025) are composed of a large number of neurons arranged in specific connection patterns, designed to simulate the neural structure and information processing mechanisms of the human brain.

In this study, RBF is employed to approximate continuous functions.

Where the input vector $Z\in\Omega_Z\subset R^q$, weight vector $\theta=\left[\theta_1,\theta_2,\ldots\theta_l\right]^{\mathrm{T}}\in R^l$, number of NN nodes $l>1$, and

$\xi(Z)=[\xi_1(Z),\xi_2(Z),\ldots,\xi_l(Z)]^{\mathrm{T}}$, and $\xi_i(Z)$ usually adopts the Gaussian function, as follows:

$$\xi_i(Z)=\exp\left[\frac{-(Z-\mu_i)^T(Z-\mu_i)}{\eta_i^2}\right], i=1,2,\ldots,l \tag{7}$$

Where $\mu_i=[\mu_{i1},\mu_{i2}\ldots,\mu_{iq}]^{\mathrm{T}}$ is the center position of the Gaussian function, and η_i is its width.

$$h(Z)=\theta^{*\mathrm{T}}\xi(Z)+\delta^*, \forall Z\in\Omega_Z \tag{8}$$

Where θ^* is the ideal constant weight, and δ^* is the approximation error.

For all $Z\in\Omega_Z$, when $\delta_m>0$, there exists an ideal constant weight θ^* such that $|\delta^*|\le\delta_m$.

$$\theta^*\triangleq\arg\min_{\theta\in R^l}\left\{\sup_{Z\in\Omega_Z}\left|h(Z)-\theta^T\xi(Z)\right|\right\} \tag{9}$$

Where θ^* is an artificially defined quantity. θ^* is the value of θ, which minimizes $Z\in\Omega_Z\subset R^q$ for all $|\delta^*|$.

2.4. Introduction of the DDPG Algorithm

The DDPG algorithm is a control method that combines deep learning with reinforcement learning principles, specifically designed for optimal control problems involving continuous state and action spaces. The core concept is to approximate both the policy function and value function using deep neural networks within a deterministic policy framework, thereby enabling end-to-end learning of continuous control strategies (Lillicrap, 2016).

In the standard reinforcement learning framework, the agent and the environment are modeled through a Markov Decision Process (MDP). The state space is denoted as S, the action space as A, the transition probabilities as $P(s_{t+1}\mid s_t,a_t)$, and the immediate reward function as $r(s_t,a_t)$. The objective is to find the optimal policy $\mu^*(s)$ that maximizes the expected cumulative discounted reward, represented as:

$$J=\mathrm{E}_{s_t\sim\rho^\mu}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,\mu(s_t))\right] \tag{10}$$

Where $\gamma\in(0,1)$ is the discount factor, and ρ^μ is the state distribution under policy μ.

The policy gradient of the DDPG algorithm can be expressed as:

$$\nabla_\theta J(\mu_\theta)=\mathrm{E}_{s_t\sim\rho^\mu}\left[\nabla_a Q^\mu(s_t,a)|_{a=\mu_\theta(s_t)}\nabla_\theta\mu_\theta(s_t)\right] \tag{11}$$

Where $\mu_\theta(s_t)$ denotes the deterministic policy network controlled by parameter θ, and $Q^\mu(s_t,a_t)$ is the evaluation of the state-action pair by the action-value function (Critic network).

In implementation, DDPG adopts an Actor–Critic architecture. The Actor network is responsible for generating a continuous control action based on the current state, and is formally defined as:

$$a_t=\mu_\theta(s_t) \tag{12}$$

On the other hand, the Critic network approximates the optimal Q-function by minimizing the Bellman error:

$$L(\phi)=\mathrm{E}_{(s_t,a_t,r_t,s_{t+1})\sim D}\left[\left(Q_\phi(s_t,a_t)-y_t\right)^2\right] \tag{13}$$

$$y_t=r_t+\gamma Q_{\phi'}(s_{t+1},\mu_{\theta'}(s_{t+1})) \tag{14}$$

Where ϕ' and θ' respectively denote the parameters of the Target Network, which are used to stabilize the training process, and D is the Replay Buffer.

To enhance training stability, the DDPG algorithm incorporates experience replay and a soft-update mechanism for the target networks. Experience replay mitigates sample correlation by randomly sampling from a buffer of past transitions, while soft updates ensure that the target networks evolve smoothly, preventing divergence during learning.

The soft-update rule is defined as:

$$\theta'\leftarrow\tau\theta+(1-\tau)\theta' \tag{15}$$

$$\phi'\leftarrow\tau\phi+(1-\tau)\phi' \tag{16}$$

Where $\tau\in(0,1)$ is the soft update coefficient.

In summary, the DDPG algorithm leverages deterministic policy gradients together with the powerful approximation capabilities of deep neural networks to enable efficient learning of optimal strategies in continuous-control settings. Its relatively simple architecture, fast convergence, and high control precision make it particularly well suited for systems characterized by continuity, strong nonlinearities, and significant uncertainties-such as ship heading-control applications.

3. Controller

3.1. Controller Design

In deriving the state-space representation of the ship heading-control system, let $x_1=\phi$, $x_2=\dot{\phi}$, $u=\delta$ and denote the ship's heading angle, heading rate, and rudder-angle input, respectively. In practical marine environments, external disturbances are unavoidable and exhibit stochastic, time-varying characteristics due to wind, waves, and currents, as well as internal factors such as propeller pulsation and hull vibration.

On the basis of (1) and (2), by introducing a lumped disturbance term into the state-space formulation, the nonlinear ship-control model that accounts for realistic operating conditions can be expressed as follows:

$$\begin{cases}\dot{x}_1=f_1(x_1)+g_1(x_1)x_2+d_1\\ \dot{x}_2=f_2(\bar{x}_2)+g_2(\bar{x}_2)u+d_2\\ y=x_1\end{cases} \tag{17}$$

Here, $x=[x_1,x_2]^{\mathrm{T}}$ and $\bar{x}_2=[x_1,x_2]^{\mathrm{T}}$, with $f_2(\bar{x}_2)=-(K/T)H(\dot{\phi})$ and $g_2(\bar{x}_2)=(K/T)$ (it is noted that $f_1(x_1)=0$ and $g_1(x_1)=1$). However, in the subsequent design, we assume that f_i and g_i (for i=1, 2) are unknown nonlinear functions; $d_1=0$, while $d_2=0.05^*\sin(0.8^*t)+0.1$ denotes the bounded uncertain terms of the system (e.g.,

nonlinear uncertainties, bounded disturbances, time-varying parameters, etc.); u stands for the control input of the system subject to saturation constraints; and $y = x_1$ is the output of the system.

Assumption 1 in the control system, the reference signal $y_d(t)$ is smooth and bounded, and its second-order derivative is continuous and bounded, i.e., there exists a positive constant B_0 such that the set $\prod_0 := \{(y_d, \dot{y}_d, \ddot{y}_d) : y_d^2 + \dot{y}_d^2 + \ddot{y}_d^2 \le B_0\}$ holds.

Assumption 2 The uncertain term d has an upper bound, i.e., there exists an unknown positive constant $|d^*|$ that satisfies $d \le |d^*|$.

In modern control theory and engineering practice, input saturation is a ubiquitous constraint that can severely degrade system performance and, in extreme cases, drive the closed loop unstable. Designing adaptive control algorithms via backstepping while explicitly accounting for input saturation has therefore become a central research focus.

The algorithm is developed on the basis of rigorous mathematical derivation and Lyapunov stability theory, and proceeds in two key stages. First, an appropriate Lyapunov function (Massera, 1949) is constructed to design an intermediate (virtual) control law; by exploiting the properties of its time derivative, a virtual control input is synthesized to ensure the stability of each local subsystem. Second, using this intermediate law as a foundation and incorporating the actual input constraints, the real control law is derived so as to guarantee closed-loop stability while improving overall performance. The detailed design procedure is as follows:

Step 1: Define the first error surface $z_1 = x_1 - y_d$ from system (17), and its derivative is:

$$\dot{z}_1 = f_1(x_1) + g_1(x_1)x_2 + d_1 - \dot{y}_d \tag{18}$$

Given a compact $\Omega_{x_1} \in R^1$, let θ_1^* and δ_1^* be for any $x_1 \in \Omega_{x_1}$, and define:

$$\begin{aligned} h_1(Z_1) &= \frac{1}{g_1(x_1)}(f_1(x_1) - \dot{y}_d) \\ &= \theta_1^{*T}\xi_1(Z_1) + \delta_1^* \end{aligned} \tag{19}$$

Select the virtual control law:

$$\alpha_2 = -\left(c_1 + \frac{1}{\gamma_1^2}\right)z_1 - \hat{\theta}_1^T\xi_1(Z_1) \tag{20}$$

Where $\hat{\theta}_1$ is the estimated value of θ_1^*, and its online update form is as follows:

$$\dot{\hat{\theta}}_1 = \Gamma_1\left[\xi_1(Z_1)z_1 - \sigma_1\hat{\theta}_1\right] \tag{21}$$

Where $c_1 > 0, \gamma_1 > 0, \Gamma_1 > 0, \sigma_1 > 0$.

To avoid repeated integration of α_2, the DSC technique is introduced, and a first-order filter β_2 (Lee, 2024) is defined as follows:

$$\tau_2\dot{\beta}_2 + \beta_2 = \alpha_2, \beta_2(0) = \alpha_2(0) \tag{22}$$

Step 2: Define the second error surface $z_2 = x_2 - \beta_2$, then:

$$\dot{z}_1 = f_1(x_1) + g_1(x_1)(z_2 + \beta_2) + d_1 - \dot{y}_d \tag{23}$$

Define

$$\begin{aligned} \eta_2 &= \beta_2 - \alpha_2 \\ &= \hat{\theta}_1^T\xi_1(Z_1) + \left(c_1 + \frac{1}{\gamma_1^2}\right)z_1 + \beta_2 \end{aligned} \tag{24}$$

Substituting (19) and (24) into (23) yields,

$$\begin{aligned} \dot{z}_1 = {} & d_1 + g_1(x_1) \\ & \left(z_2 - \tilde{\theta}_1^T\xi_1(Z_1) - \left(c_1 + \frac{1}{\gamma_1^2}\right)z_1 \right. \\ & \left. + \delta_1^* + \eta_2 \right) \end{aligned} \tag{25}$$

Then

$$\begin{aligned} \dot{z}_2 = {} & g_2(\overline{x_2})(-\theta_2^T\xi_2(Z_2) + d_2 \\ & -(c_2 + \frac{1}{\gamma_2^2})z_2 + e + \Delta u + \delta_2^*) \end{aligned} \tag{26}$$

Where $c_2 > 0$.

Consider the saturation auxiliary system,

$$\dot{e} = \begin{cases} -ke - \dfrac{|z_2\Delta u| + \frac{1}{2}\Delta u^2}{e^2}e + \Delta u, |e| \ge \varepsilon \\ 0, |e| < \varepsilon \end{cases} \tag{27}$$

Where $k > 0, \varepsilon > 0, \Delta u = u - v$.

The baseline control law is chosen as follows:

$$v_{BL} = -\left(c_2 + \frac{1}{\gamma_2^2}\right)z_2 + e - \hat{\theta}_2^T\xi_2(Z_2) \tag{28}$$

$$\dot{\hat{\theta}}_2 = \Gamma_2\left[\xi_2(Z_2)z_2 - \sigma_2\hat{\theta}_2\right] \tag{29}$$

Step 3: Under the DDPG framework, v_{RL} is constructed and trained by v_{BL}. In this paper, the residual compensation learned by the DDPG actor is superimposed on v_{BL} to obtain v_{RL}. Denote the system state as s_t.

$$v_{BL}(s_t) = \pi_B(s_t) \tag{30}$$

Where $V \subset R^m$ is the feasible control domain (such as input saturation constraint) $\pi_B : S \to V$.

1. State Normalization:

$$\overline{s_t} = N(s_t) \tag{31}$$

Where $N(\cdot)$ is a fixed, invertible normalization mapping (e.g., a dimensionality or magnitude scaling transformation) that does not update during training.

2. Deterministic Policy and Exploration:

$$a_t = \mu_\theta(\overline{s_t}), a_t = a_t + \varepsilon_t \tag{32}$$

Where μ_θ is the actor of DDPG; ε_t is the zero-mean exploration noise (available during training, and set to $\varepsilon_t = 0$ during deployment).

3. Dimension/Amplitude Mapping (mapping dimensionless actions to physical units):

$$\Delta v_t = S a_t \tag{33}$$

Where S is a constant linear mapping responsible solely for converting units and scaling magnitudes.

4. First-order Time-domain Smoothing (exponential moving average to suppress chattering):

$$\Delta v_t = \lambda_t \Delta v_t + (1-\lambda_t)\Delta v_{t-1} \tag{34}$$

Where $\lambda_t \in (0,1]$ can be fixed or scheduled in stages (for example, a smaller value of λ_t is adopted in the steady-state phase to enhance smoothness).

5. Residual Superposition and Feasible Region Projection:

$$v_{RL}(s_t) = \Pi_V(v_{BL}(s_t) + \beta_t \Delta v_t) \tag{35}$$

Where $\Pi_V(\cdot)$ is the projection onto V; $\beta_t \in [0,1]$ is the residual weight to balance the baseline and RL compensation.

Can be obtained

$$\begin{aligned} v = v_{BL} + v_{RL} = -\left(c_2 + \frac{1}{\gamma_2^2}\right) z_2 \\ + e - \hat{\theta}_2^T \xi_2(Z_2) + v_{RL} \end{aligned} \tag{36}$$

3.2. controller configuration

the controller configuration and the DDPG-related parameters are introduced as follows:

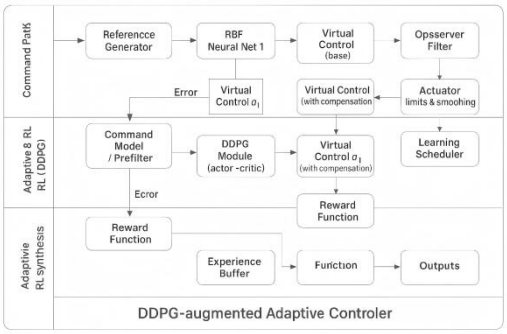

Fig. 1 Controller Flow Chart

Figure 1 presents the flow diagram of the proposed controller. The process begins with the Reference Generator, which produces the desired heading command. This reference is first processed by RBF Neural Net 1 and then passed to the Virtual Control (base) block to generate a baseline virtual control signal. This signal is subsequently filtered through the Observer Filter and adjusted by the Actuator Limits & Smoothing module, producing a compensated virtual control (a1). The tracking error between this compensated signal and the reference is then forwarded to the Adaptive & RL (DDPG) module.

Within this module, the Command Model Prefilter preprocesses the error signal before feeding it into the DDPG Module (Actor–Critic), which outputs an adaptive correction term that further refines the virtual control (a1). Simultaneously, the tracking error is transformed into a reward signal via the Reward Function, while the Learning Scheduler manages the DDPG training process.

Finally, reward signals together with state-transition samples are stored in the Experience Buffer of the Adaptive RL Synthesis module. Through experience replay and optimization in the Function module, updated control commands are generated, thereby completing the full closed-loop process of the DDPG-augmented adaptive controller, enabling both effective regulation and online learning.

4. System Stability Analysis

Proof: Select the Lyapunov function (Lee, 2024) as follows:

$$\begin{aligned} V = \frac{1}{2} z_1^2 + \frac{1}{2} \tilde{\theta}_1^T \Gamma_1^{-1} \tilde{\theta}_1 \\ + \frac{1}{2} z_2^2 + \frac{1}{2} \tilde{\theta}_2^T \Gamma_2^{-1} \tilde{\theta}_2 \\ + \frac{1}{2} \eta_2^2 + \frac{1}{2} \beta_t \Delta \hat{v}_t^2 + \frac{1}{2} e^2 \end{aligned} \tag{37}$$

Taking the derivative, we get

$$\begin{aligned} \dot{V} = z_1 \dot{z}_1 - \tilde{\theta}_1^T \Gamma_1^{-1} \dot{\hat{\theta}}_1 + z_2 \dot{z}_2 - \tilde{\theta}_2^T \Gamma_2^{-1} \dot{\hat{\theta}}_2 \\ + \eta_2 \dot{\eta}_2 + \beta_t \Delta \hat{v}_t \Delta \dot{\hat{v}}_t + e\dot{e} \end{aligned} \tag{38}$$

Where $\dot{\eta}_2 = \dot{\beta}_2 - \dot{\alpha}_2 = -\eta_2 / \tau_2 - \dot{\alpha}_2$ is bounded by combining Young's inequality (Zhang, 2004) to obtain $z_1 g_1 \eta_2 - \eta_2^2 / \tau_2 \le (g_1^2 z_1^2)/2 + \eta_2^2 (1/2 - 1/\tau_2)$, and $\tau_2 < 2$ is selected to ensure that the negative terms dominate:

$$\eta_2 \dot{\eta}_2 \le \frac{g_1^2}{2} z_1^2 - \frac{1}{2\tau_2} \eta_2^2 \tag{39}$$

Combined with the residual boundedness $|\Delta v_t| \le M$ from (32), it can be obtained by scaling that:

$$\beta_t \Delta \hat{v}_t \Delta \dot{\hat{v}}_t \le \frac{\beta_t M^2}{2} + \frac{\beta_t}{2} \Delta \hat{v}_t^2 \tag{40}$$

Substituting (24), (25), (27), (39) and (40) into (38), we can obtain

$$\begin{aligned} \dot{V} \le -c_1 z_1^2 + z_1 g_1(x_1) z_2 + z_1 g_1(x_1) \eta_2 + z_1 d_1 \\ - \sigma_1 \tilde{\theta}_1^T \hat{\theta}_1 + z_1 \delta_1^* - c_2 z_2^2 - z_1 g_1(x_1) z_2 \\ + z_2 g_2 e + z_2 d_2 - \sigma_2 \tilde{\theta}_2^T \hat{\theta}_2 + z_2 \delta_2^* \\ + z_2 g_2 \Delta \hat{v}_t + \frac{g_1^2}{2} z_1^2 - \frac{1}{2\tau_2} \eta_2^2 - k e^2 \\ + e z_2 g_2 + \frac{\beta_t M^2}{2} + \frac{\beta_t}{2} \Delta \hat{v}_t^2 \\ \le -\left(c_1 - \frac{g_1^2 + 1}{2}\right) z_1^2 - \left(c_2 - \frac{1}{2}\right) z_2^2 \\ - \frac{1}{2\tau_2} \eta_2^2 - k e^2 - \frac{\sigma_1}{2} \| \tilde{\theta}_1 \|^2 - \frac{\sigma_2}{2} \| \tilde{\theta}_2 \|^2 + D \end{aligned} \tag{41}$$

Where $D = \frac{D_1^2 + D_2^2 + \Delta_1^2 + \Delta_2^2 + \beta_t M^2}{2}$ is a constant.

Selecting the design parameters to satisfy $c_1 > \frac{g_1^2 + 1}{2}$, $c_2 > \frac{1}{2}$, $\tau_2 > 0$, $k > 0$, $\sigma_1 > 0, \sigma_2 > 0$, then the coefficients of all error terms in $(\dot{V})$ are negative.

At this time, $\dot{V} \le -CV + D$ (Sanjeevin, 2025), where $C = \min\left\{2(c_1 - \frac{g_1^2+1}{2}), 2(c_2 - \frac{1}{2}), \frac{1}{\tau_2}, 2k, \sigma_1 / \lambda_{max}(\Gamma_1^{-1}), \sigma_2 / \lambda_{max}(\Gamma_2^{-1})\right\} > 0$.

According to Lyapunov stability theory, all signals $z_1, z_2, \tilde{\theta}_1, \tilde{\theta}_2, \eta_2, e, \Delta \hat{v}_t$ of the system are uniformly ultimately bounded, and the tracking error $z_1 = x_1 - y_d$ can make the ultimate bound arbitrarily small by adjusting parameters.

5. Simulation Verification

In this section, MATLAB is used to carry out simulation studies to evaluate the effectiveness of the proposed controller. The simulations are conducted using the "Yulong" training vessel from Dalian Maritime University (Li, 2025). The principal particulars of the vessel are as follows: length 126.0 m, beam 20.8 m, full-load draft 8.0 m, block coefficient 0.681, and service speed 7.7 m/s. Based on these specifications, the Nomoto model parameters are computed, and the initial values of $c_1 = 0.55$, $c_2 = 150$, $c_{21} = 0.5$, $\varepsilon = 0.01$, $\Gamma_1 = diag\{0.001\}$, $\Gamma_2 = diag\{0.01\}$, $T = 1200$, e are all set to 20 for the simulations.

DDPG-related parameters:

Definition of state space and action space:

The state space comprises 8 continuous variables: heading tracking error (rad), angular velocity (rad/s), heading angle (rad), reference angle (rad), first-stage output of the RBF neural network (dimensionless), reference angular velocity (rad/s), internal state of the virtual control (consistent with the unit of the virtual control variable), and error rate of change (rad/s). For network input, these variables undergo element-wise normalization using scaling factors of π/3, π/3, 2π, 2π, 10, π/3, π/3, and π/3 in the specified order, respectively. The action space consists of one continuous action: a compensation signal (with units matching the virtual control variable), which is superimposed onto the baseline virtual control variable to generate the final virtual control variable. The amplitude of this compensation signal is constrained by a predefined maximum action magnitude; this constraint can be enforced by first clamping the output to the range [-1, 1] via a tanh activation function, followed by scaling to the preset upper limit. The Actor network takes the aforementioned environmental states as input and outputs the continuous compensation signal. The Critic network accepts state-action pairs as input and produces a single value estimate. Network layers are constructed using layer graphs or a series of network functions, including fully connected layers, batch normalization layers, and activation layers. Specifically, the Actor network is composed of multiple hidden layers and terminates with a linear layer (a tanh activation may be applied at the output to constrain the action range), while the Critic network delivers a single-value output via a value function layer. Prior to training, weight initialization is performed, and the loss functions and optimizers are defined. The Actor network is updated based on the Critic's estimate of the negative value. The Critic network is optimized by minimizing the mean squared error between the actual return and the predicted value. The network training function conducts joint training of the two networks in accordance with predefined training hyperparameters, including learning rate, number of iterations, and validation frequency.

The heading tracking error refers to the deviation between the actual heading of the ship and the desired heading. The reward is inversely proportional to this error, promoting a reduction in the error. The reward function is designed with penalty terms for factors such as sharp steering maneuvers, frequent rudder changes, out-of-range operations, dynamic mismatches, and response delays. These penalties are introduced to discourage unreasonable control actions and guide the agent toward learning efficient steering commands, thereby optimizing control performance.

Additionally, the reward function takes into account the smoothness of the control actions to avoid abrupt control changes. It also includes penalties for out-of-range behaviors to ensure that the control inputs stay within feasible and safe limits.

The final reward function is formulated as follows: $(e, \Delta\delta, c, s) = e + \Delta\delta + c + s$.

Where e is the heading error; $\Delta\delta$ is the rudder angle amplitude; c is the heading convergence term; s is the steady-state additional smoothing term.

Simulation Results are as follows:

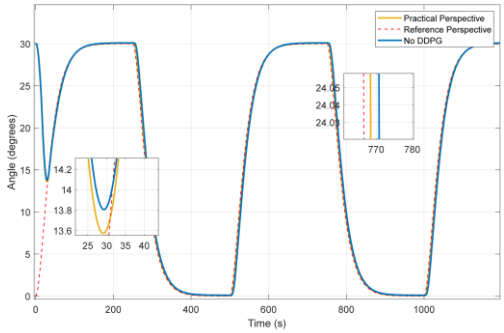

Fig. 2 Time response of curve of ship course

The heading trajectory curve in Figure 2 includes the actual heading (yellow curve), the desired heading (red dashed line), and the actual heading achieved by the RBF Neural Network-based DSC control method that does not utilize DDPG, but accounts for input saturation (blue curve). In the initial phase, the actual heading curve rapidly converges towards the desired heading, and after stabilization, the error approaches zero, demonstrating the effectiveness and accuracy of the heading tracking system presented in this study.

Compared to the heading control without DDPG, the proposed method exhibits faster response and better tracking performance. The deviation angle is smaller, and the system demonstrates superior ability to precisely control the ship's heading under complex operational conditions.

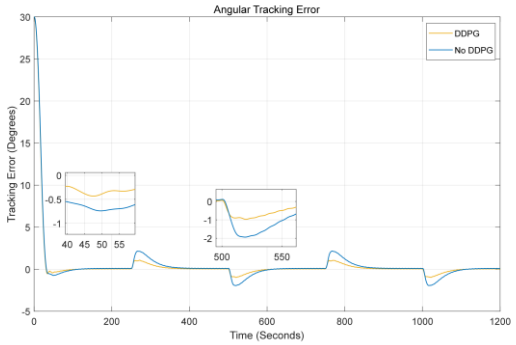

Fig. 3 Time-history curve of ship tracking error

Figure 3 presents the heading tracking error. The heading tracking error for the proposed method (yellow curve) remains at a low level for the majority of the time, achieving effective tracking to a significant degree. In comparison with the RBF Neural Network-based DSC control method that accounts for input saturation but does not use DDPG (blue curve), the proposed method demonstrates distinct advantages.

At the beginning of the simulation, the heading tracking error for the proposed method quickly converges to near zero, with faster convergence and greater stability. In the steady-state phase, the error fluctuations are smaller, resulting in a more stable ship heading. From a local detail perspective, in the zoomed-in region, the error peak of the proposed method is significantly lower than the blue curve, showing a reduction of approximately 50%. The method exhibits stronger ability to suppress error growth when facing local disturbances, resulting in higher tracking accuracy.

Overall, the proposed method outperforms the alternative in terms of convergence speed, steady-state stability, and disturbance rejection accuracy, demonstrating significant advantages in all aspects.

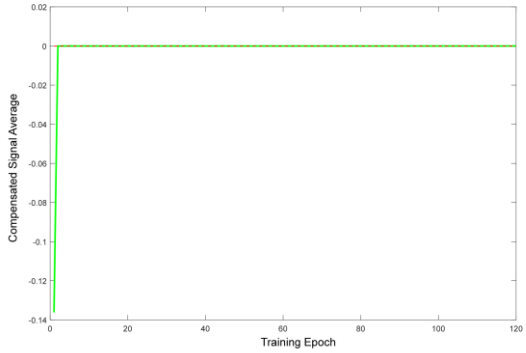

Fig. 4 Average compensation signal

Figure 4 illustrates the evolution of the mean compensation signal over training episodes, with the horizontal axis representing the number of training iterations and the vertical axis indicating the magnitude of the compensation signal. The results show that, in the early stages of training, the compensation signal rapidly converges and stabilizes near zero, and it remains consistently steady throughout subsequent iterations. This behavior indicates that the compensation strategy employed in this study exhibits strong convergence properties, reaching a stable regime early in the training process.

Once the system converges to this stable state, the controller requires virtually no additional corrective action. This implies that the dynamic response of the system has been effectively optimized: the control policy has been thoroughly learned and refined, and the system output is capable of accurately tracking the desired trajectory with minimal compensatory effort.

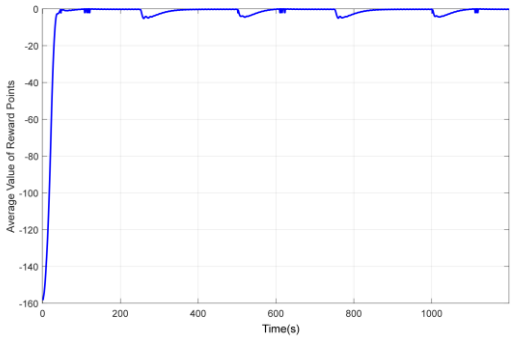

Fig. 5 Time response of average reward value

Figure 5 shows the average reward value of the DDPG agent. As illustrated, the mean reward increases rapidly during the early stages of training, followed by a gradual convergence toward zero. When the tracking error is substantial, the reward assumes a negative value, acting as a penalty imposed on the non-negligible error. During this process, several sharp and rapid transitions occur in the reference angle (desired heading)-for example, abrupt changes from near 0° to approximately 30°, and then quickly back to 0°. The control policy must adapt to these sudden heading transitions, and fluctuations in control performance naturally lead to corresponding variations in the reward signal, producing small oscillations.

This result demonstrates that, under the reinforcement-learning paradigm, the proposed control strategy is able to achieve effective convergence of the reward function through iterative training. After convergence, the reward signal exhibits a reasonable degree of stability. Although minor fluctuations persist due to environmental dynamics or fine adjustments in the learned policy, the overall trend indicates that the agent progressively approaches an optimal control strategy.

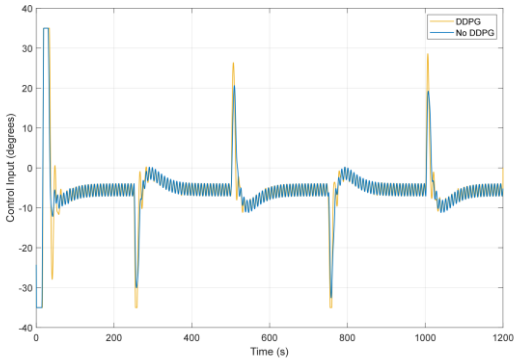

Fig. 6 Time response of ship rudder angle

Figure 6 presents the time history of the ship's rudder-angle input. Overall, the rudder-angle evolution for the proposed method (yellow curve) is broadly consistent with that of the RBF neural network–based DSC controller without DDPG but with input saturation considered (blue curve). The two methods exhibit similar rudder adjustment trends across most time intervals, indicating that both controllers generate comparable steering commands under nominal operating conditions.

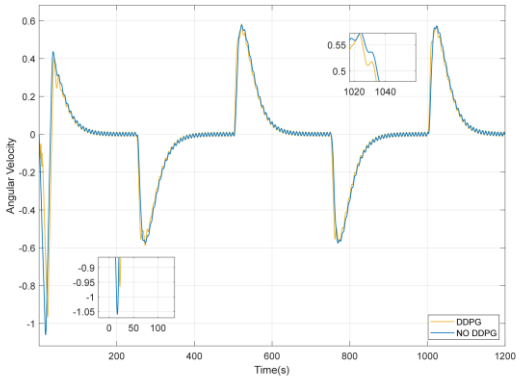

Fig. 7 Time response of ship yaw rate

Figure 7 shows the time history of the ship's yaw-rate response. Overall, both the proposed method (yellow curve) and the RBF neural network–based DSC controller without DDPG but with input saturation considered (blue curve) exhibit fluctuating yaw-rate profiles with broadly similar temporal trends. This indicates that both controllers capture the essential dynamic characteristics of the vessel's turning behavior, producing comparable yaw-rate responses under the tested conditions.

6. Conclusion

This study addresses key limitations of traditional ship heading-control methods-namely insufficient robustness under complex sea conditions, strong dependence on manual parameter tuning, and the difficulty of handling input-saturation constraints. To overcome these challenges, we propose an adaptive DSC framework augmented by the DDPG algorithm. By embedding deep reinforcement learning within a direct adaptive DSC architecture, the method enables online optimization of control parameters and continuous policy improvement, thereby enhancing the system's capability to cope with nonlinear uncertainties and external disturbances.

Simulation results demonstrate that the proposed controller achieves high tracking accuracy and strong robustness in dynamically varying environments. Compared with conventional adaptive control algorithms, the DDPG-enhanced approach more effectively suppresses the influence of disturbances, significantly reduces steady-state tracking errors, and maintains system stability as well as rapid response performance even under input-saturation constraints.

Overall, the DDPG-based adaptive heading-control strategy presented in this work provides an intelligent and efficient solution for autonomous ship-navigation systems. Moreover, it offers a promising direction for advancing the application of deep reinforcement learning in the control of complex nonlinear systems.

References

Cui, B., Chen, Y., Hong, X. et al. (2024), Research on path-following technology of a single-outboard-motor USV based on DRL and MPC, Journal of Marine Science and Engineering, Vol. 12, No. 12, pp. 2321.

Du, J.L., Hu, X. and Sun, Y.Q. (2020), Adaptive robust nonlinear control design for course tracking of ships subject to external disturbances and input saturation, IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 50, No. 1, pp. 193–202.

Gao, X., Hu, X. and Yang, A. (2025), Fuzzy reinforcement learning disturbance cancellation optimized course tracking control for USV autopilot under actuator constraint, Journal of Marine Science and Engineering, Vol. 13, No. 8, pp. 1429.

Greep, J., Bayezit, A.B., Mak, B. et al. (2025), Ship course-keeping in waves using sample-efficient reinforcement learning, Engineering Applications of Artificial Intelligence, Vol. 141, pp. 109848.

Jia, X.L. and Yang, Y.S. (1997), Mathematical model of ship motion, Dalian Maritime University Press: Dalian.

Le, T.T. (2021), Ship heading control system using neural network, Journal of Marine Science and Technology, Vol. 26, No. 3, pp. 963–972.

Lee, T.C., Zhang, Y. and Mareels, I. (2024), Multi-agreements in social networks based on LaSalle invariance principle and positive definiteness, Automatica, Vol. 165, pp. 111666.

Li, J.F. (2012), Design of marine course autopilot considering input saturation, Navigation College, Dalian Maritime University: Dalian.

Lillicrap, T.P., Hunt, J.J., Pritzel, A. et al. (2016), Continuous control with deep reinforcement learning, Proceedings of the 4th International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, USA.

Liu, H., Zhou, X., Tian, X. et al. (2024), Adaptive self-structuring neural network tracking control for underactuated USVs with actuator faults and input saturation, Ocean Engineering, Vol. 309, pp. 118535.

Liu, S., Zhang, G., Zhang, W. et al. (2022), Robust fuzzy

dynamic surface formation control for underactuated ships using MLP and LFG, Systems Science & Control Engineering, Vol. 10, No. 1, pp. 272–281.

Ma, L., Qi, M.Z., Chen, S.C. et al. (2025), Optimization of underactuated ship sliding mode controller based on the DDPG algorithm, Journal of Marine Science and Technology, Vol. 33, No. 2, pp. 160–168.

Massera, J.L. (1949), On Lyapunov's conditions of stability, Annals of Mathematics, Vol. 50, No. 3, pp. 705–721.

Mu, D.D., Wang, G.F., Fan, Y.S. et al. (2019), Adaptive course control based on trajectory linearization control for unmanned surface vehicle with unmodeled dynamics and input saturation, Neurocomputing, Vol. 330, pp. 1–10.

Ou, C.K., Xie, L., Zha, T.Q. et al. (2024), Marine path planning based on deep reinforcement learning and historical trajectory, Navigation of China, Vol. 47, No. 1, pp. 36–44, 51.

Sanjeevini, S., Lai, B. and Kouba, O. et al. (2025), Input-to-state stability of discrete-time linear time-varying systems, Automatica, Vol. 177, pp. 112331.

Sivaraj, S., Dubey, A. and Rajendran, S. (2023), On the performance of different deep reinforcement learning based controllers for the path-following of a ship, Ocean Engineering, Vol. 286, pp. 115607.

Sun, W.C., Bu, R.X. and Liu, Y. (2020), Ship PID derivative compensation course control with roll suppression function, Journal of Shanghai Maritime University, Vol. 41, No. 3, pp. 19–24.

Tian, C., Cheng, T., Peng, Z. et al. (2025), A survey on deep learning fundamentals, Artificial Intelligence Review, Vol. 58, No. 2, pp. 363–381.

Wang, X., Liu, Y., Zhang, H. et al. (2024), PID controller based on improved deep deterministic policy gradient for trajectory tracking control of unmanned surface vehicles, Journal of Marine Science and Engineering, Vol. 12, No. 10, pp. 1771.

Xu, Q., Li, N. and Jin, Z.H. (2012), Combined optimization of allocation for full and empty containers, Journal of Transportation Systems Engineering and Information Technology, Vol. 12, No. 1, pp. 145–152.

Xie, Y., Luo, F., Zeng, J. et al. (2020), Disturbance observer-based path following control of unmanned surface vessel with control input saturation and unknown disturbance, IOP Conference Series: Earth and Environmental Science, Vol. 440, No. 2, pp. 022062.

Zhang, Y.Z. (2004), Proof and application of Young's inequality, Henan Science, Vol. 22, No. 1, pp. 23–29.

Zhong, Q.M., Qin, L.H., Zhang, S.T. et al. (2025), Nonlinear roll control of ships based on deep reinforcement learning, Ship Engineering, Vol. 47, No. 7, pp. 112–120.

Received **26 December 2025**

1st Revised **06 January 2026**

Accepted **09 January 2026**