Electronic supplementary materials

For https://doi.org/10.1631/jzus.A2500146

Three-degree-of-freedom motion posture stabilization control of platform based on DTW-LSTM-MATD3 under high and low frequency disturbances of ships

Qin ZHANG, Jingyi ZHOU, Bangping GU, Xiong HU✉

School of Logistics Engineering, Shanghai Maritime University, Shanghai 201306, China
✉ Xiong HU, huxiong@shmtu.edu.cn

Section S1

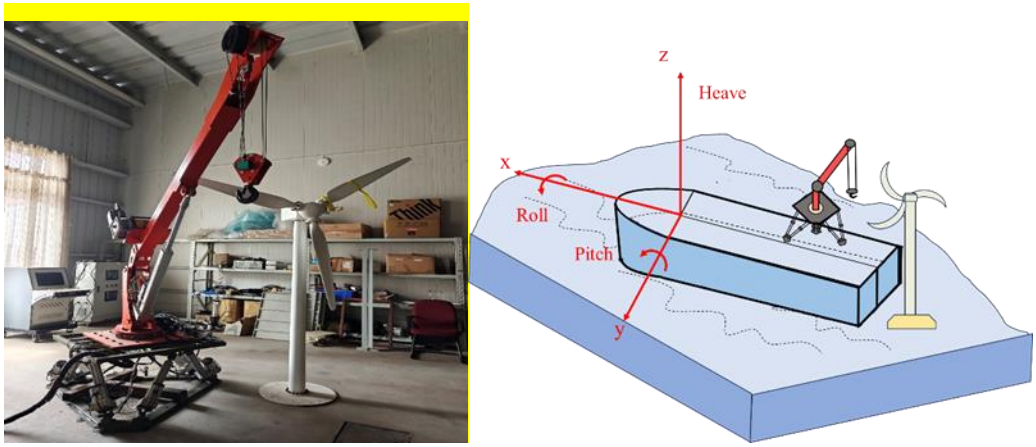

Fig. S1 Ship motion simulation and compensation system schematic diagram: (a) ship motion simulation diagram, (b) compensation system schematic diagram

Section S2.1 Analysis of the motion characteristics of the load platform

Calculate the connection points between the load platform and each electric cylinder in the moving coordinate system (i=1,2…6), and calculate the connection points between the base platform and each electric cylinder in the fixed coordinate system (i=1,2…6). As following:

$$P_1 = [R_1 \cos(\frac{\pi}{2} - b) \quad R_1 \sin(\frac{\pi}{2} - b) \quad 0]^T \tag{S1}$$

$$B_1 = [R_2 \cos(\frac{\pi}{6} + a) \quad R_2 \sin(\frac{\pi}{6} + a) \quad 0]^T \tag{S2}$$

In Eqs. (1) and (2), R_1, R_2 are the distances from point O_p, O_b to electric cylinder 1. The role of a and b in defining the path of the hinge point. The moving coordinate system can be made to coincide with the fixed coordinate system through several translations and rotations. The transformation matrix T is as follows:

$$T=\begin{bmatrix} cq_5\cdot cq_6 & -sq_5\cdot sq_6 & -sq_5 & q_1 \\ cq_4\cdot sq_6+sq_4\cdot sq_5\cdot cq_6 & cq_4\cdot cq_6-sq_4\cdot sq_5\cdot sq_6 & -sq_4\cdot cq_5 & q_2 \\ sq_4\cdot sq_6-cq_4\cdot sq_5\cdot cq_6 & sq_4\cdot cq_6+cq_4\cdot sq_5\cdot sq_6 & cq_4\cdot cq_5 & q_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(S3)}$$

Where c represents the *cos*, and s represents the *sin*. The upper-left 3×3 matrix is the rotation matrix r, and q_1, q_2, and q_3 are the translation distances of the moving coordinate system along the $x^{'}$, $y^{'}$, $z^{'}$ axes of the fixed coordinate system. q_4、q_5、q_6 are the rotation angles of the moving coordinate system around the $x^{'}$, $y^{'}$, $z^{'}$ axes of the fixed coordinate system, respectively. We use the coordinate transformation to convert P_i in the moving coordinate system into $P_i^{'}$ in the fixed coordinate system:

$$P_i^{'}=r(P_i+[0 \quad 0 \quad h]^T), i=1,2...6 \quad \text{(S4)}$$

Where r is the rotation matrix determined by the posture of the load platform, which is used to convert vectors from the moving coordinate system to the global coordinate system. h is the distance that the moving platform first translates along the Z-axis.

To precisely describe the positions and motions of the components of the Stewart platform, we establish a global coordinate system and select reference origins O_p and O_b in the coordinate systems of the load platform P and the base platform B. For each electric cylinder, the vector from the connection point on the base platform to the connection point on the load platform $\vec{S_i}$ $(i=1,2,\cdots,6)$ enable be expressed as:

$$\vec{S_i}=r\vec{P_i}+\overrightarrow{O_{bp}}-\vec{q_i} \quad \text{(S5)}$$

Where $\overrightarrow{O_{bp}}$ is the vector from point O_b to the connection point O_p between the load platform and the i-th upper leg. $\vec{q_i}$ is the vector from O_b to the connection point between the base platform and the i-th lower leg.

When performing kinematic analysis on a single electric cylinder, the extension length $L_i=|\vec{S_i}|$, unit vector $\vec{s_i}=\vec{S_i}/L_i$, and the relationship between the electric cylinder's velocity, acceleration, and the load platform's motion are obtained. Since the structure and connection methods of the electric cylinders in the Stewart platform are the same, the above calculation method applies to all six legs. Taking the calculation of the electric cylinder's motion speed as an example, for the i-th electric cylinder, its motion speed is given by $\dot{S}_i=s_i\cdot(v+\omega\times q_i)$, where v is the speed of the load platform's reference point O_P, and ω is the angular velocity of the load platform. The $\vec{s_i}$ and $\vec{q_i}$ will vary owing to different positions, and the motion speeds of the six electric cylinders can be calculated separately to analyze the overall motion characteristics of the load platform.

Section S2.2 Analysis of the Balance Equation of Electric Cylinders

When performing dynamic analysis on a single electric cylinder, it is necessary to consider the balance of forces and torques acting on it. Taking the *i*-th electric cylinder as an example, the main driving torque is in equilibrium with the gravitational torque, constraint torque, and frictional torque. The rotational balance equation is as follows:

$$-m_d a_d \vec{r_d} - m_u \vec{r_u} + (m_d \vec{r_d} + m_u \vec{r_u})g - (I_d + I_u)a - W(I_d + I_u)W + M_u \vec{s} + F_s \vec{S} - C_u W - f = 0 \quad \text{(S6)}$$

Where m_d and m_u are the masses of the lower and upper legs, respectively. $\vec{r_d}$ and $\vec{r_u}$ are the position vectors of the centers of gravity of the lower and upper legs. a_d is the accelerations of the lower legs. g is the acceleration owing to gravity. I_d and I_u are the inertia matrices of the lower and upper legs. a is the angular acceleration of the electric cylinder, and W is the angular velocity of the electric cylinder. $m_d a_d \vec{r_d}$ is the force acting on the lower leg along its center of gravity vector. M_u is the constraint torque at the Hooke hinge connection point between the base platform and the lower leg. $\vec{s}$ is the direction vector of the constraint torque at the Hooke hinge. F_s is the constraint torque at the connection point between the load platform and the upper leg owing to the spherical hinge. $\vec{S}$ is the direction vector of the constraint torque at the spherical hinge. C_u is the viscous friction coefficient of the Hooke hinge, and f is the frictional torque of the spherical hinge.

The force balance equation along the direction of the electric cylinder for the upper leg is:

$$F + s \cdot F_s - C_p \dot{L} - m_u s \cdot g + m_u s \cdot a_u = 0 \quad \text{(S7)}$$

Where F is the main driving force acting on the upper leg, C_p is the viscous friction coefficient of the spherical hinge, a_u is the acceleration of the upper leg, and $\dot{L}$ is the rate of change of the electric cylinder's length. Since each electric cylinder has the same physical structure and mechanical principles. After each cylinder moves, it needs to achieve the displacement required for compensation. Therefore, it is necessary to consider the differences in inertia and displacement vectors among the cylinders. Ultimately, the coordinated motion of the entire load platform must be achieved to satisfy the force and torque balance. The dynamic equation of the Stewart platform that meets the force balance and rotational balance is as follows:

$$M_p a_p + M_p g + r_p F_{ext} - \sum_{i=1}^{6} (F_s)_i = 0 \quad \text{(S8)}$$

Where M_p is the mass of the load platform, a_p is the acceleration of the load platform's center of gravity, g is the acceleration owing to gravity, r_p is the rotation matrix of the load platform, F_{ext} is the external force acting on the load platform, and $(F_s)_i$ is the constraint force at the connection

point between the *i*-th electric cylinder and the load platform.

Section S2.3 Motor System Modeling

The two-phase AC servo motor consists of two mutually perpendicular stator coils and a rotor. The motor characteristic curve is as follows:

$$M_m = -\frac{M_{se}}{E}\omega_m + C_m U_a \tag{S9}$$

Where M_m is the output torque, C_m is a constant coefficient, U_a is the input voltage, E is the rated voltage, and M_{se} is the stall torque at the rated voltage. When the motor input voltage U_a is given, the output torque M_m is used to drive the load and overcome the viscous friction force; thus, the electromagnetic torque is dissipated as both the load torque and the frictional resistance torque. The dynamic balance is given by:

$$\begin{cases} M_{\mathrm{m}}(t) = J_m \dfrac{d^2\theta_m}{dt^2} + f_m \dfrac{d\theta_m}{dt} \\ \omega_m(t) = \dfrac{d\theta_m}{dt} \end{cases} \tag{S10}$$

Where θ_m is the angular displacement of the motor rotor, J_m, f_m are the total moment of inertia and the total viscous friction coefficient referred to the motor, respectively.

We can obtain the differential equations describing the motion of the servo motor as follows:

$$J_m \frac{d^2\theta_m}{dt^2} + (f_m + C_\omega)\frac{d\theta_m}{dt} = C_m U_a(t) \tag{S11}$$

Under zero initial conditions, the Laplace transform of the equation is:

$$J_m s^2 \theta_m(s) + (f_m + C_\omega) s\theta_m(s) = C_m U_a(s) \tag{S12}$$

Thus, the transfer function model of the servo motor is obtained as:

$$G_1(s) = \frac{\theta_m(s)}{U_a(s)} = \frac{C_m}{s(J_m s + (f_m + C_\omega))} \tag{S13}$$

Based on the rated voltage, rated torque, rated output power, and other relevant parameters of the motor, the following values can be obtained: $C_m = 0.013 N \cdot m / V$, $C_\omega = 2.94 \times 10^{-2} N \cdot m/(r/s)$, $f_m = 2.54 \times 10^{-2} N \cdot m/(r/s)$, $J_m = 0.678 \times 10^{-4} kg \cdot m^2$, Substituting these into Eqs.(9), the transfer function of the servo motor used in this study is obtained as:

$$G_1(s) = \frac{13}{0.0678 s^2 + 54.8 s} \tag{S14}$$

The electric cylinder is a product designed by modularly combining the motor and ball screw through a high-strength servo synchronous belt, converting the motor's rotational motion into the linear motion of the ball screw nut, and then into the linear motion of the telescopic rod inside the electric cylinder. Ideally, the linear relationship between the motor rotation angle θ_m and the

telescopic rod displacement X_L is as follows:

$$\theta_M = \frac{2\pi}{P_n} i_r X_L \tag{S15}$$

where i_r is the reduction ratio, and P_n is the lead of the ball screw. However, in practical applications, when the angular displacement output by the motor is converted into the displacement of the telescopic rod through the transmission mechanism, there is a delay. To more accurately describe the response of the actual system, a first-order inertial element with an inertial time constant T_d is added to the transfer function. Therefore, the mathematical model of the electric cylinder is as follows:

$$G_2(s) = \frac{X_L(s)}{\theta_M(s)} = \frac{P_n}{2\pi i_r (T_d s + 1)} \tag{S16}$$

Where P_n=5mm, i_r=1, and T_d= 1. Substituting the relevant parameters into Eqs.(18), the transfer function of the electric cylinder is obtained as:

$$G_2(s) = \frac{5}{3.14s + 6.28} \tag{S17}$$

Since the motion control card used in this study differentially controls the servo electric cylinder, its model can be simplified to a differential element. Therefore, combining Eqs (14), (17), etc., the transfer function of the entire servo-electric cylinder system is:

Section S3.1 Lyapunov Analysis of Combined Reward Functions

To solve the problem of insufficient control intensity of linear reward in a small error interval and the instability of normal reward in a large error interval, we propose a composite reward function. This function employs normal rewards in small error intervals to enhance local exploration capability and linear rewards in large error intervals to ensure rapid convergence.

For the composite reward function, linear and normal terms are integrated, with the error $r(e)$ being segmented by interval:

$$\begin{cases} r_{line} = k_1 e(t), (\|e\| > e_0) \\ r_{norm} = k_2 \varepsilon(t), (\|e\| \le e_0) \end{cases} \tag{S18}$$

In Eqs.(18), r_{line}, r_{norm} are the linear and normal reward functions, $k_1, k_2 > 0$ are the linear and normal reward coefficients, e_0 is the error threshold, and $\|e\|$ is the Euclidean norm of the error. The system dynamics change synchronously with the interval:

$$\dot{e}(t) = \begin{cases} k_3 e(t) + k_4 k_1 e(t), (\|e\| > e_0) \\ k_3 e(t) + k_4 k_2 \varepsilon(t), (\|e\| \le e_0) \end{cases} \tag{S19}$$

In Eqs.(19), k_3, k_4 is the coefficient of the error function. Therefore, $V(e)$ is established as

follows:

$$V(e)=\begin{cases} e^T P_1 e, \|e\| > e_0 \\ e^T P_2 e + \beta\sigma^2, \|e\| \le e_0 \end{cases} \tag{S20}$$

In Eqs. (20), $P_1, P_2 > 0$ are positive definite matrices, and $\beta > 0$ is a correction coefficient. To analyze Lyapunov stability more clearly, we conduct the following analysis on the two intervals separately:

(a) For the interval with larger errors ($\|e\| > e_0$), Derivation of $V(e)$ leads to the following:

$$\dot{V}(e) = e^T[(k_3 + k_4 k_1)^T P_1 + P_1(k_3 + k_4 k_1)]e \tag{S21}$$

In Eqs. (21), $(k_3 + k_4 k_1)^T P_1 + P_1(k_3 + k_4 k_1) = -Q_1$, then:

$$\dot{V}(e) = -e^T Q_1 e < 0 \tag{S22}$$

According to the Lyapunov stability theory, the system is asymptotically stable in a large error interval, and the deterministic control with linear reward ensures rapid convergence of the error.

(b) For the interval with smaller errors ($\|e\| \le e_0$), take $V(e)$ the derivative and calculate to obtain $E[\dot{V}(e)]$:

$$E[\dot{V}(e)] = E[e^T({k_3}^T P_2 + P_2 k_3)e + 2k_2 e^T P_2 k_4 \varepsilon(t)] \tag{S23}$$

Owing to $E[\varepsilon(t)] = 0$ and state independence, the Eqs.(29) is as follows:

$$E[\dot{V}(e)] = E[e^T({k_3}^T P_2 + P_2 k_3)e] \tag{S24}$$

Let ${k_3}^T P_2 + P_2 k_3 = -Q_2$, where $Q_2 > 0$ is a positive definite matrix, simplify to get the following:

$$E[\dot{V}(e)] = -E[e^T Q_2 e] \le -\lambda_{\min}(Q_2) E[\|e\|^2] < 0 \tag{S25}$$

Among them, $\lambda_{\min}(Q_2) > 0$ is the smallest eigenvalue of the matrix Q_2. According to the stochastic Lyapunov stability theory, the system is asymptotically stable in the sense of expectation. The stochastic perturbation of the normal reward enhances local exploration within a small error interval while maintaining convergence in a probabilistic sense.

Section S3.2 TD3 Algorithm

The TD3 algorithm is a deep reinforcement learning algorithm based on the Actor-Critic framework. The Actor and Critic networks primarily consist of an input layer, hidden layers, and an output layer. The input layer uses a sequential input layer or a feature input layer, primarily to input the displacement of the electric cylinder within a certain period of time into the neural network. The hidden layer usually contains multiple fully connected layers; each neuron is composed of several neurons connected by weights to speed up the training of the network. The

output layer uses a tanh activation function and determines the output action based on the dimension of the action space. The Actor-Critic framework performs well in continuous action space problems. The algorithm addresses the central challenge of the MDP by iteratively refining the Critic and Actor networks, ultimately discovering the optimal policy for cumulative reward maximization. During the training of the agent, the state transition probability is as follows:

$$P_{ss'}^{a} = P(s_{t+1} = s' \mid s_t = s, a_t = a) \tag{S26}$$

where $P_{ss'}^{a}$ is the probability of the next state being s' when the current state of the agent is s and the current action is a.

In the TD3 algorithm, the agent uses a dual Critic network structure to estimate the Q-values and updates the network parameters through temporal difference errors. The target Q-values are calculated twice, effectively addressing the overestimation problem.

$$\begin{aligned} y_1 &= r_i + \gamma Q_i(s', a_1', \ldots\ldots a_N') \\ y_2 &= r_i + \gamma Q_i(s', a_1', \ldots\ldots a_N') \end{aligned} \tag{S27}$$

where y_1 is the target Q-value of Critic1 network, and y_2 is the target Q-value of Critic2 network.

There is always a size relationship between the two Q-values. The larger target Q-value has a higher possibility of causing overestimation of the policy. Therefore, the smaller Q-value is chosen as the target Q-value to solve the problem of Q-value overestimation and enhance the convergence of the algorithm.

$$y = r_t + \gamma \min Q_{\theta_i'}(S', A') \tag{S28}$$

Where y is the target Q-value at time step t.

To reduce the overestimation bias, the parameters of the target Actor network and the target Critic are updated in the same way, using soft updates:

$$\begin{aligned} \theta_i' &\leftarrow \tau\theta_i + (1-\tau)\theta_i' \\ \varphi_i' &\leftarrow \tau\varphi_i + (1-\tau)\varphi_i' \end{aligned} \tag{S29}$$

where τ is the soft update factor, generally $\tau \ll 1$, θ_i are the parameters of the Critic network, and φ_i are the parameters of the Actor network.

To increase the exploration of the policy, the TD3 algorithm adds noise ε when selecting actions:

$$a = \mu(O_1 + \varepsilon), |\varepsilon| \sim clip(N(0, \sigma), -c, c) \tag{S30}$$

The TD3 algorithm primarily consists of three parts. Firstly, it uses two Critic networks and takes the smaller value when calculating the target value to reduce the overestimation of Q-values. Secondly, it employs target policy smoothing regularization by adding noise to the action in the next state when calculating the target value, as shown in Eqs. (28), making the Q-value estimation

more accurate. Finally, it uses delayed policy updates, updating the Actor network only after the Critic network has been updated multiple times, ensuring more stable training of the Actor network.

Section S3.3 Noise Impact Analysis of the MATD3 Algorithm

The TD3 algorithm has achieved reasonable control effects in specific scenarios; the single-agent TD3 algorithm still has limitations. In the single-agent framework, only a single target can be controlled, which brings difficulties to the control of multiple targets. The overestimation of *Q*-values still exists, causing the policy update to deviate from the correct direction and affecting the performance of the algorithm. Therefore, scholars have proposed the MATD3 algorithm, which combines the idea of multi-agent with TD3. Each agent interacts with the environment to obtain initial information, such as the displacement and speed of the electric cylinder, and the agents share information with each other. A centralized training and decentralized execution strategy is adopted. When updating, each agent's Critic network obtains information from all agents to more comprehensively evaluate the actions taken by the agents. After the Critic network is updated multiple times, it outputs the *Q*-value of the agent, which is then used to update the parameters of its own Actor network towards the maximum *Q*-value. Through this process, the agents learn to maximize cumulative rewards and output the optimal actions for the electric cylinders.

To enable the MATD3 algorithm to perform well in continuous control tasks, deterministic policies are used to directly output action values. The loss function $L(\theta_i)$ for the Critic network of the *i*-th agent is given by:

$$L(\theta_i) = E_{s,a,r,s' \sim D}[(y_i - Q_i(s, a_1, ..., a_N; \theta_i))^2] \tag{S31}$$

Where θ_i represents the parameters of the Critic network for the *i*-th agent, *D* is the experience replay buffer, $Q_i(s, a_1, ..., a_N; \theta_i)$ is the value assessed by the Critic network for the current state *s*, and y_i is the target value for each agent.

By taking the gradient of the loss function with respect to θ_i, the gradient equation for the Critic network can be obtained:

$$\nabla_{\theta_i} L(\theta_i) = E_{s,a,r,s' \sim D}[2(y_i - Q_i(s, a_1, ..., a_N; \theta_i)) \nabla_{\theta_i} Q_i(s, a_1, ..., a_N; \theta_i)] \tag{S32}$$

In Eqs. (32), a mini-batch of data is usually sampled from the experience replay buffer *D* to approximate this expectation, and the Critic network parameters θ_i are updated using gradient descent.

After training with the MATD3 algorithm, agents can collaborate with each other, improving

the stability and robustness of the algorithm. It also has certain generalization capabilities, making it more valuable for practical multi-agent compensation control applications.

This study focuses on controlling the six electric cylinders of the Stewart platform. Therefore, the TD3 algorithm is extended to a multi-agent environment. For the MATD3 algorithm, the state space *S* and the action space *A* are concatenated and input into the Critic network to evaluate the actions taken by the compensation platform. Then, the action space *A* is input into the Actor network, which maps the state to the action and provides feedback to the agent. The reward function *R* is used to evaluate the state at the next time step. However, for the ship's 3-DoF posture motion, the increase in training data for the MATD3 algorithm weakens the state transition advantage of the MDP. Moreover, when disturbances exist in the motion state, the training difficulty also increases.

In the Actor network of the MATD3 algorithm, the input state s_t^* is affected by noise. Let the noise-free state s_t and the noise d_t. The policy function π_θ is used to select the optimal action. Consequently, the Actor network outputs a highly fluctuating action a_t^*, as follows:

$$s_t^* = s_t + d_t \tag{S33}$$

$$a_t^* = \pi_\theta(s_t^*) \tag{S34}$$

At this point, the Actor network generates an action error e_a^*:

$$e_a^* = a_t^* - a_t \tag{S35}$$

Where a_t is the action output by the Actor network in the noise-free state. According to Eqs.(33) and (34), noise can cause deviations in the policy gradient and network parameter updates of the Actor network. It also affects the *Q*-value estimation of the subsequent Critic network. In the Critic network, both the input state s_t^* and action a_t^* are affected by noise, leading to biased *Q*-value estimation, as follows:

$$e_c^* = Q(s_t^*, a_t^*) - Q(s_t, a_t) \tag{S36}$$

In Eqs.(36), the higher the noise frequency in the input state, the more the target *Q*-value deviates from the normal value. This forces the Critic network to learn from the noisy state rather than the true action value. It results in drastic gradient changes, severely affecting the stability of the agent's training. At this time, the system error is $e^* = e + d_t$ (where *e* is the error of the system in the noise-free state). In control theory, it is usually desired that the system error gradually tends to zero. The error gain *k* determines the system's response speed to the error. The dynamic equation of the system is $\dot{e} = -ke$ (where *k* is the control gain). Constructing a Lyapunov

function $V(e^*)=(e^*)^T Pe^*$ (where $P > 0$ is a positive definite matrix), the derivative of $V(e^*)$ is as follows:

$$\dot{V}(e^*)=2(\dot{e}^*)^T Pe^*=2(e+d_t)^T P(\dot{e}+\dot{d}_t) \tag{S37}$$

Split into four terms:

$$\dot{V}(e^*)=2\left(e^T P\dot{e}+e^T P\dot{d}_t+d_t^T P\dot{e}+d_t^T P\dot{d}_t\right) \tag{S38}$$

Substitute the dynamic equation $\dot{e}=-ke$ into it:

$$\begin{aligned}\dot{V}(e^*)&=2\left(e^T P(-ke)+e^T P\dot{d}_t+d_t^T P(-ke)+d_t^T P\dot{d}_t\right)\\&=-2ke^T Pe+2e^T P\dot{d}_t-2kd_t^T Pe+2d_t^T P\dot{d}_t\end{aligned} \tag{S39}$$

For the cross terms $2e^T P\dot{d}_t$ and $-2kd_t^T Pe$, apply Young's inequality, obtaining:

$$\begin{cases}2e^T P\dot{d}_t\le\alpha\|e\|^2+\dfrac{1}{\alpha}\|P\dot{d}_t\|^2\\-2kd_t^T Pe\le k^2\|d_t\|^2+\|Pe\|^2\end{cases} \tag{S40}$$

Where $\alpha>0$. Substitute the above inequality into the equation $\dot{V}(e^*)$:

$$\begin{aligned}\dot{V}(e^*)&\le-2ke^T Pe+\left(\alpha\|e\|^2+\frac{1}{\alpha}\|P\dot{d}_t\|^2\right)+\left(k^2\|d_t\|^2+\|Pe\|^2\right)+2d_t^T P\dot{d}_t\\&=\left(-2k\lambda_{\min}(P)+\alpha+\lambda_{\max}(P)\right)\|e\|^2+\left(\frac{\lambda_{\max}(P^2)}{\alpha}+2\lambda_{\max}(P)\right)\|d_t\|^2\end{aligned} \tag{S41}$$

Where $\lambda_{\min}(P)$ and $\lambda_{\max}(P)$ are the minimum and maximum eigenvalues of matrix P Choose appropriate parameters $\lambda_{\min}(P)$ and $\lambda_{\max}(P)$, then we get the following:

$$\begin{cases}-\alpha=-2k\lambda_{\min}(P)+\alpha+\lambda_{\max}(P)\\\beta=\dfrac{\lambda_{\max}(P^2)}{\alpha}+2\lambda_{\max}(P)\end{cases} \tag{S42}$$

Solving this yields the following:

$$\begin{cases}\alpha=k\lambda_{\min}(P)-\dfrac{\lambda_{\max}(P)}{2}\\\beta=\dfrac{\lambda_{\max}(P^2)}{k\lambda_{\min}(P)-\dfrac{\lambda_{\max}(P)}{2}}+2\lambda_{\max}(P)\ge\dfrac{\lambda_{\max}(P^2)}{k\lambda_{\min}(P)}+2\lambda_{\max}(P)\end{cases} \tag{S43}$$

Finally, we obtain the following:

$$\dot{V}(e^*)=-\alpha\|e\|^2+\beta\|d_t\|^2 \tag{S44}$$

Where $\alpha>0$ is the stable gain, and β is the noise-related coefficient. When $\dot{V}(e^*)<0$, the system remains in a stable state. Therefore, determining the critical condition of the Lyapunov function is crucial. To better analyze the stability performance of the system under different frequency noises, according to Parseval's theorem of equivalence between time-domain energy and frequency-domain energy, we have:

$$\begin{cases}\|e\|^2=\int_{-\infty}^{+\infty}|E(\omega)|^2 d\omega\\\|d_t\|^2=\int_{-\infty}^{+\infty}|D(\omega)|^2 H(\omega)d\omega\end{cases} \tag{S45}$$

Where $E(\omega)$、$D(\omega)$ are the frequency-domain system error and noise after Fast Fourier Transform (FFT), respectively, and $H(\omega)$ is the frequency-domain weighting function representing the system's sensitivity to ω. Since different frequency noises have different impacts on system performance, high-frequency noise significantly affects control accuracy, while low-frequency noise may be naturally attenuated. At this time, according to Eqs. (43), the critical condition of the Lyapunov function becomes:

$$\alpha\int_{-\infty}^{+\infty}|E(\omega)|^2\,d\omega=\beta\int_{-\infty}^{+\infty}|D(\omega)|^2\,H(\omega)d\omega \tag{S46}$$

Since the energy of the noise is concentrated at the boundary frequency $\omega_c=2\pi f_c$, the frequency-domain representation of the noise at this time can be expressed as $D(\omega)=\delta(\omega-\omega_c)$ (where $\delta(\cdot)$ is the Dirac function). Meanwhile, there is a positive correlation k_{ED} between the system's error energy and the noise energy. Therefore, the critical condition equation of the Lyapunov function can be simplified to:

$$\begin{cases}\alpha k_{ED}=\beta H(\omega_c)=\beta H(2\pi f_c)\\ \int_{-\infty}^{+\infty}|E(\omega)|^2\,d\omega=|E(\omega_c)|^2\\ |E(\omega)|^2=k_{ED}\,|D(\omega)|^2\\ \int_{-\infty}^{+\infty}|D(\omega)|^2\,H(\omega)d\omega=H(\omega_c)\end{cases} \tag{S47}$$

Further, the noise frequency f_c at the critical point of the system's Lyapunov function can be obtained as:

$$f_c=\frac{H^{-1}(\frac{\alpha k_{ED}}{\beta})}{2\pi} \tag{S48}$$

In summary, when the input noise frequency of the system $f>f_c$, $\dot{V}(e^*)>0$, the system does not satisfy Lyapunov stability. There are issues such as inaccurate *Q*-value estimation of the Critic and Actor networks, gradient direction deviation, and policy oscillation in the MATD3 algorithm. Noise interference within this range should be suppressed during the ship's compensation control process to maintain stable operation of the ship. When the input noise frequency of the system $f<f_c$, $\dot{V}(e^*)<0$, the system satisfies Lyapunov stability, and the power device of the system will produce low-frequency vibration, which should follow this type of noise motion. However, calculating f_c requires obtaining parameters such as the positive correlation k_{ED} between the error energy and noise energy in the system, and the frequency-domain weighting function *H* representing the system's sensitivity to ω. The acquisition of these parameters and functions is accompanied by problems such as frequency band conflicts. Therefore, the Dynamic Time Warping algorithm is adopted to distinguish noises in different

frequency ranges and find the boundary point f_c between high-frequency and low-frequency signals. Then, compensation control is carried out for different types of noise.

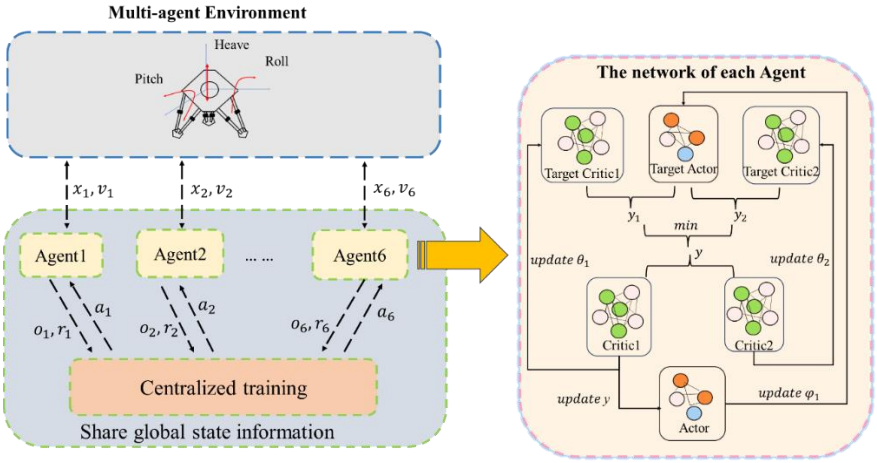

Fig. S2 Structural diagram of the MATD3 algorithm

Section S3.4 Expansion of the Softmax formula in DTW

Let the DTW value of the two ships' motion attitude time series be *D*, the DTW threshold be $D_{threshold}$, and their difference be $\Delta = D - D_{threshold}$. When there is high-frequency noise in the ship's motion ($f > 30Hz$), the two sequences show significant differences. Let, that is $D \gg D_{threshold}$, which $\Delta \gg 0$ increases. Substituting into the *Softmax* function formula:

$$D_{soft\max} = \frac{\exp(\Delta)}{1+\exp(\Delta)} = \frac{1}{1+\exp(-\Delta)} \tag{S49}$$

In Eqs.(49), Δ increases, $e^{-\Delta}$ decreases, therefore $D_{soft\max} \to 1$. Output discriminative results with low similarity between sequences.

When low-frequency noise exists in ship motion ($f < 30Hz$), the similarity between the two sequences is high, $D \ll D_{threshold}$, $\Delta \ll 0$, then substituting into the *Softmax* function formula gives:

$$D_{soft\max} = \frac{\exp(\Delta)}{1+\exp(\Delta)} = \frac{1}{1+\exp(-\Delta)} \to 0 \tag{S50}$$

Based on the above analysis, the noise generated by various physical characteristics of ships can be addressed by defining a difference threshold using DTW. This allows for a collaborative transformation from noise interference to discriminative output, enabling the algorithm to distinguish data differences caused by ship movement owing to noise.

Section S3.5 Noise Analysis of DTW-LSTM-MATD3

To handle the issues, we incorporate LSTM into the Actor and Critic networks. The input

state of the Actor network contains noise s_t^*. After passing through the fully connected layer, noise- layer has strong noise resistance, reducing the noise in the input. Through its gating mechanism, it generates a smooth hidden state h_t^{LSTM}. The final output action of the Actor network is determined by this hidden state h_t^{LSTM}, resulting in a smooth output action $\hat{a}_t$. This makes the output action error e_a^* of the Actor network approach zero, the equation is as follows:

$$\hat{a}_t \approx \hat{\pi}_\theta(h_t^{LSTM}) \quad \text{(S51)}$$

$$e_a^* = \hat{a}_t - a_t \approx 0 \quad \text{(S52)}$$

The Critic network receives the noisy state s_t^* and the smoothed action $\hat{a}_t$. After passing through the LSTM layer, the noisy deep features are further smoothed to obtain h_t^{state} and h_t^{action}. This process helps to train the Critic network to obtain *Q*-values that are closer to the true values，the *Q*-values are calculated as follows:

$$Q(s_t^*, \hat{a}_t) \approx Q(s_t, a_t) \quad \text{(S53)}$$

$$e_c^* = Q(s_t^*, \hat{a}_t) - Q(s_t, \hat{a}_t) \approx 0 \quad \text{(S54)}$$

$$Var(\nabla_\theta \hat{L}(\theta)) < Var(\nabla_\theta L^*(\theta)) \quad \text{(S55)}$$

The variance of MATD3 decreases, making the gradient updates of the Critic network more stable and the learning of *Q*-values smoother, which in turn increases the convergence speed. The policy becomes less sensitive to noise and does not fluctuate drastically owing to short-term noise, leading to more stable policy execution and reducing ineffective exploration caused by noise.

The rapid fluctuations of high-frequency noise ($f > f_c$) cause the component $\| d_t \|^2$ in the Lyapunov function to increase, $\dot{V}(e^*) > 0$ destabilizing the system. However, the LSTM network in the MATD3 algorithm smooths the noisy input states and actions according to Eqs. (53) and (55), suppressing the energy d_t of high-frequency noise and significantly weakening the perturbation term in the Lyapunov function. As a result, the derivative of the Lyapunov function in Eqs.(43), we get as follows:

$$\dot{V}(e^*) = 2(\dot{e}^*)^T P e^* = -\alpha \| e \|^2 + \beta \| d_t \|^2 \approx -\alpha \| e \|^2 < 0 \quad \text{(S56)}$$

$\dot{V}(e^*) < 0$, indicating that the system satisfies Lyapunov stability conditions.

Low-frequency signal ($f \le f_c$) changes slowly. The LSTM in the MATD3 algorithm can dynamically estimate and compensate d_t, weakening the energy of d_t to be less than the energy of the error e. At this time, the derivative of the Lyapunov function satisfies:

$$\dot{V}(e^*) = -\alpha \| e \|^2 + \beta \| d_t \|^2 < 0 \quad \text{(S57)}$$

Under these conditions, the system can stably track low-frequency fluctuation signals. In

summary, the DTW-LSTM-MATD3 algorithm reduces the system's sensitivity to noisy states, avoids oscillations or divergence during training, and maintains a stable state even after disturbance.