Journal of Marine Science and Application (2019) 18:510–521
https://doi.org/10.1007/s11804-019-00111-8

RESEARCH ARTICLE

# Rudder Roll Damping Autopilot Using Dual Extended Kalman Filter–Trained Neural Networks for Ships in Waves

Yuanyuan Wang [1] · Hung Duc Nguyen [1]

Received: 5 June 2018 / Accepted: 22 March 2019 / Published online: 16 October 2019


**Abstract**
The roll motions of ships advancing in heavy seas have severe impacts on the safety of crews, vessels, and cargoes; thus, it must be damped. This study presents the design of a rudder roll damping autopilot by utilizing the dual extended Kalman filter (DEKF)–trained radial basis function neural networks (RBFNN) for the surface vessels. The autopilot system constitutes the roll reduction controller and the yaw motion controller implemented in parallel. After analyzing the advantages of the DEKF-trained RBFNN control method theoretically, the ship's nonlinear model with environmental disturbances was employed to verify the performance of the proposed stabilization system. Different sailing scenarios were conducted to investigate the motion responses of the ship in waves. The results demonstrate that the DEKF RBFNN–based control system is efficient and practical in reducing roll motions and following the path for the ship sailing in waves only through rudder actions.

**Keywords** Rudder roll damping · Autopilot · Radial basis function · Neural networks · Dual extended Kalman filter training · Intelligent control · Path following · Advancing in waves

## 1 Introduction

At present, marine transportation is indispensable for the development of the world. The ocean-going ships always have to endure large environmental impacts including waves, winds, and currents. The dramatic roll motions will affect the comfort of seafarers and passengers because of seasickness. Moreover, it may lead to the instability and unsafety of the ship and cargoes. From the perspective of safety, the roll damping facilities or strategies are needed to damp the roll motions as much as possible. Some effective methods and devices, e.g. moving weights (Treakle et al. 2000), stabilizing fins, anti-roll tanks, and bilge keels, have been successfully designed to reduce roll motions. However, the additional equipment affects the ship's carrying capacity, seaworthiness, and structure strength and increases the shipbuilding and maintenance costs. Therefore, other applicable approaches are needed for vessels to keep stability while maintaining the orientation.

When altering the rudder angle, there exist additional force and moment on the roll motions. Consequently, the rudder is an alternative to control roll motions besides being used as the steering facility. Using the rudder for path following and roll reduction simultaneously is not a simple task due to the coupling between the motions of yaw and roll. Hence, qualified control strategies are required to handle the trade-offs. The conventional rudder roll stabilizer based on the proportional-integrative-derivative (PID) control method (Van and Van 1978) has been applied in the commercial autopilot system for its simplification and reliability. However, this kind of controllers, which was designed with fixed parameters and scheduling gains, does not work well in heavy seas (Sun et al. 2014). With the development of modern control theory, fuzzy logic control algorithm was adopted to design the rudder roll stabilization system (Nejim 2000). However, it is difficult

**Article Highlights**
- The desired feedback RBFNN–based control algorithm with DEKF training method was formulated.
- The DEKF RBFNN–based rudder roll stabilization was proposed.
- Roll damping and path following were achieved simultaneously only through rudder actions.
- In comparison with the BP RBFNN control system, the proposed system has low cost and better roll reduction performance.

✉ Yuanyuan Wang
Yuanyuan.Wang@utas.edu.au

[1] National Center for Maritime Engineering and Hydrodynamics, Australian Maritime College, University of Tasmania, Launceston 7250, Australia

to formulate the fuzzy control rules, which are generally obtained by trial-and-error-based human knowledge. The H-Infinite-based rudder roll stabilization system was developed by Zhang et al. (2006), but the control method has the risk of being unstable when the changing speed of the dynamics is beyond the adapting capability. The sliding mode control also has been adopted in this domain (Fang and Luo 2007). However, the high frequency of control actions may lead to unexpected dynamic distortions (Sun et al. 2014).

Impelled by the development of computing technology, the control algorithms based on neural networks became applicable. The advantages of the neural network control algorithm lie in the capability of approximating nonlinearity and robustness against system noises. Another feature of the neural network is the capability in 'comprehending' the multivariable characteristics of the system by adjusting the weights, which avoids the analytical analysis of the complicated nonlinear differential equations. In Alarçïn (2007), the rudder roll stabilization system for fishing vessels was designed by using a neural network approach, in which the sigmoid transfer function is used as the activation functions. By utilizing the hyperbolic tangent function, Fang et al. (2010) proposed a PID controller tuned by the neural network to control the ship's roll motions in random waves. The wavelet neural network also has been investigated to design the roll damping controller (Li et al. 2010). Amongst the multilayer feed-forward neural networks, the radial basis function neural networks (RBFNN) has simple architecture and owns the adequate generalization capability in avoiding unnecessary and lengthy calculations (Liu 2013). Thus, the RBFNN is adopted in this study to design the control system.

The training algorithm is essential to propose a neural network controller. Up to the present, most of the neural network–based roll stabilizers are trained by the well-known backpropagation (BP) algorithm or its variants (Alarçïn 2007; Fang et al. 2012). Although the BP methods are applicable to train neural networks, the relevant flaws are still of concern. Since BP is a first-order steepest descent method, the training of networks with gradient descent tends to be slowly and poorly approaching satisfactory results (Ko and Lee 2013). Additionally, the propagation of dynamic derivatives regarding the networks' output is computationally expensive (Choi et al. 2005). Actually, the training of proposed neural networks can be considered as the process of estimating parameters. As the optimal state estimator, the Kalman filter can be employed as the alternative. Amongst the Kalman filter variants, the extended Kalman filter (EKF) algorithms, which are capable of achieving second-order nonlinearity accuracy by using Jacobian matrix for approximation, can provide the online training mechanism (Medagam and Pourboghrat 2009). Although the unscented Kalman filter (UKF) algorithm has been investigated as a satisfied neural network training method to the control of the ship's motions (Wang et al. 2017a), the additional computational burdens are excessive since it needs to handle the propagations of 'sigma points' to capture the true means and covariances of the Gaussian random variables through the system dynamics. Contrary to the higher order training methods, the EKF-based training algorithm for RBFNN does not require batch processing and can easily access the Jacobian matrix, making it more suitable for online usage (Zhao et al. 2013). From the running time standpoint, the EKF-based training algorithm is more competent with reasonable computational expenses.

Besides the standard EKF, the augmented EKF (Goh and Mandic 2007) and decoupled EKF (Nouri et al. 2008) have been developed to train neural networks. It is shown that, with the application of the training algorithm based on EKF variants, the converging speed and the capability in restraining noises were improved (Sanchez et al. 2008). However, the algorithms were proposed to train the weights but not proposed to train the dynamic parameters in the activation functions. For the ship's motion control, which always requires the alternations of the desired attitude due to the passage planning, the capabilities of the RBFNN controller also depend on the online optimization of the centers in radial basis functions. Therefore, the dual EKF (DEKF) is utilized to train both weights and centers of the RBFNN controller in this study.

The main motivation of this study is to develop a qualified rudder roll damping autopilot based on DEKF RBFNN, which has low computational expenses and satisfactory tracking capabilities. The key objectives of this research are (1) to formulate the DEKF-based training method for the feedback RBFNN control scheme; (2) to analyze the merits of the proposed training method in comparison with the BP training method; (3) to develop a rudder roll stabilization system based on DEKF RBFNN for surface vessels advancing in waves; and (4) to validate the efficiency of the system by achieving the tasks of path following and roll damping simultaneously with environmental disturbances.

The organization of this study is as follows: section 2 introduced the preliminaries including the state space function of the ship's motions in four degrees of freedom (DOF) as well as the control scheme of a direct method for robust adaptive control by RBFNN. In section 3, the DEKF training method for RBFNN-based controller and its theoretical advantages are presented. After that, in section 4, the rudder roll stabilization system was proposed by using DEKF RBFNN–based control algorithm to fulfil the tasks of roll damping and path following through the rudder actions. The simulation scenarios and results are presented in section 5 to investigate the efficiency of the system. Conclusions are given in section 6.

## 2 Problem Formulations and Preliminaries of the Direct RBFNN Control

### 2.1 Ship's Motion Equations

The ship's mathematical model, which is deduced from Newton-Euler's law, is the representation of the ship's motions. The kinematic and kinetic states of the surface vessel are clarified in the two correlative coordinates, namely earth-fixed coordinate and body-fixed coordinate (see Fig. 1).

Considering the objectives of roll damping and path following, the four-DOF (i.e. surge, sway, yaw, and roll) model is used to describe the dynamic motions of surface vessels. The ship's four-DOF nonlinear model including control forces and manoeuvring characteristics can be expressed as (Fossen 1994):

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}\boldsymbol{v} \tag{1}$$

$$\boldsymbol{M}\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{v})\boldsymbol{v} + \boldsymbol{D}(\boldsymbol{v},\boldsymbol{\eta})\boldsymbol{v} + \boldsymbol{G}\boldsymbol{\eta} = \boldsymbol{\tau}_{\mathrm{ex}} + \boldsymbol{\tau} \tag{2}$$

where the vectors $\boldsymbol{\eta} = [x, y, \phi, \psi]^{\mathrm{T}}$ and $\boldsymbol{v} = [u, v, p, r]^{\mathrm{T}}$ are used to represent the states of the ship's motions, where $(x, y)$ are the position, $\phi$ and $\psi$ are the roll angle and yaw angle in the earth-fixed coordinate, $(u, v)$ are the linear velocity of surge and sway in the body-fixed coordinate, $p$ and $r$ are the roll rate and yaw rate of the ship, $\boldsymbol{J}$ is the transformation matrix of the kinetic and kinematic states, $\boldsymbol{M}$ is the system inertial matrix, $\boldsymbol{C}(\boldsymbol{v})$ is the Coriolis and centripetal matrix, $\boldsymbol{D}(\boldsymbol{v})$ is the damping matrix, $\boldsymbol{G}\boldsymbol{\eta}$ denotes the vector of gravitational force and moments, and $\boldsymbol{\tau}$ denotes the vector of control inputs, $\boldsymbol{\tau}_{\mathrm{ex}}$ is the matrix of external disturbance.

Specifically, the above-mentioned matrices are given as:

$$\boldsymbol{M} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 & 0 \\ 0 & m - Y_{\dot{v}} & -mz_g - Y_{\dot{p}} & mx_g - Y_{\dot{r}} \\ 0 & -mz_g - Y_{\dot{v}} & I_x - Y_{\dot{p}} & 0 \\ 0 & mx_g - N_{\dot{v}} & 0 & I_z - N_{\dot{r}} \end{bmatrix} \tag{3}$$

$$\boldsymbol{C}(\boldsymbol{v}) = \begin{bmatrix} 0 & 0 & mz_g r & Y_{\dot{v}} v - m(rx_g + v) \\ 0 & 0 & 0 & mu - X_{\dot{u}} u \\ -mz_g r & 0 & 0 & Y_{\dot{v}} v \\ -Y_{\dot{v}} v + m(rx_g + v) & -mu + X_{\dot{u}} u & -Y_{\dot{v}} v & 0 \end{bmatrix} \tag{4}$$

$$\boldsymbol{D}(\boldsymbol{v},\boldsymbol{\eta}) = \begin{bmatrix} -X_{uu}u - X_{uuu}u^2 & -X_{vv}v - X_{rv}r & 0 & -X_{rr}r \\ -Y_{uv\phi}v\phi - Y_{ur\phi}r\phi & -Y_{uuv}u^2 - Y_{vvv}v^2 - Y_{rrv}r^2 & 0 & -Y_{uur}u^2 - Y_{rrr}r^2 - Y_{vvr}v^2 \\ -K_{uv\phi}v\phi - K_{ur\phi}r\phi & -K_{uuv}u^2 - K_{vvv}v^2 - K_{rrv}r^2 & K_p - K_{ppp}p^2 & -K_{uur}u^2 - K_{rrr}r^2 - K_{vvr}v^2 \\ -N_{uv\phi}v\phi - N_{ur\phi}r\phi & -N_{uuv}u^2 - N_{vvv}v^2 - N_{rrv}r^2 & 0 & -N_{uur}u^2 - N_{rrr}r^2 - N_{vvr}v^2 \end{bmatrix} \tag{5}$$

$$\boldsymbol{G} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -K_{\phi} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{6}$$

$$\boldsymbol{\tau}_{\mathrm{ex}} = \left[\tau_{\mathrm{ex}|X}, \tau_{\mathrm{ex}|Y}, \tau_{\mathrm{ex}|K}, \tau_{\mathrm{ex}|N}\right]^{\mathrm{T}} \tag{7}$$

$$\boldsymbol{\tau} = [\tau_X, \tau_Y, \tau_K, \tau_N]^{\mathrm{T}} \tag{8}$$

where $m$ is the mass of the ship; $x_g$ and $z_g$ are the position of the ship's center of mass in the body-fixed coordinate; $I_x$ and $I_z$ are the moment of inertia; $X_{\dot{u}}$, $Y_{\dot{v}}$, $Y_{\dot{r}}$, $N_{\dot{v}}$, $Y_{\dot{p}}$, and $N_{\dot{r}}$ are the hydrodynamic added mass and added moment inertia of the ship; $X_{\cdot}$, $Y_{\cdot}$, $K_{\cdot}$, and $N_{\cdot}$ with subscripts are the hydrodynamic coefficients concerning the corresponding dimension; $\tau_X$, $\tau_Y$, $\tau_K$, and $\tau_N$ are the forces and moments generated from actuators including propellers and rudder on corresponding DOF; and $\tau_{\mathrm{ex}\,|\,X}$, $\tau_{\mathrm{ex}\,|\,Y}$, $\tau_{\mathrm{ex}\,|\,K}$, and $\tau_{\mathrm{ex}\,|\,N}$ are the environmental disturbances on corresponding DOF, including the disturbances of waves, currents, and wind.

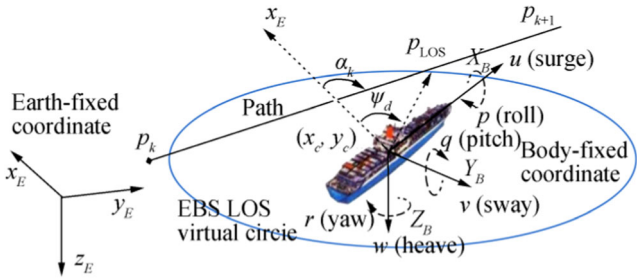



**Fig. 1** Two correlative coordinates for the motions of the surface vessel

### 2.2 RBFNN-Based Function Approximation for Feedback Control

In order to better represent the process of the ship's motions, the four-DOF mathematical model of the surface vessel can be rewritten as a second-order nonlinear system. Considering the fact that the transformation matrix of the states (i.e. $\boldsymbol{J}$) is a determined item in every iteration of calculating ordinary differential equation (ODE), when the relevant items are defined as $A(\boldsymbol{S}) = \frac{-\boldsymbol{C}(\boldsymbol{v})-\boldsymbol{D}(\boldsymbol{v},\boldsymbol{\eta})}{\boldsymbol{M}} + \frac{-\boldsymbol{G}\boldsymbol{\eta}}{\boldsymbol{M}}\boldsymbol{v}$, $B(\boldsymbol{S}) = \frac{1}{\boldsymbol{M}}$, $C = \boldsymbol{J} \in \boldsymbol{R}$, $d = \frac{\boldsymbol{\tau}_{\mathrm{ex}}}{\boldsymbol{M}}$, $u = \tau$, the Eq. (1) can be simplified as the following state space equations:

$$\begin{aligned} \dot{s}_1 &= Cs_2 \\ \dot{s}_2 &= A(\boldsymbol{S}) + B(\boldsymbol{S})u + d \\ Y &= s_1 \end{aligned} \tag{9}$$

where $\boldsymbol{S}=[s_1\ s_2]^{\mathrm{T}}$ are the system states of the plant, which contains the relevant controllable variables in the control process, $u$ is the input of the plant being controlled, and $Y$ is the output of the plant.

In the closed-loop control of the plant, the target value $\boldsymbol{Y}_d$, the error $\boldsymbol{e}$, and an augmented error term $e_A$ can be defined as:

$$\begin{aligned}\boldsymbol{Y}_d &= \left[Y_d\ \dot{Y}_d\right]^{\mathrm{T}}, \boldsymbol{e}=\boldsymbol{Y}_d-\boldsymbol{Y}=\left[e\ \dot{e}\right]^{\mathrm{T}}\\ &= \left[Y_d-Y, \dot{Y}_d-\dot{Y}\right]^{\mathrm{T}}, e_A=[\lambda\ 1]\boldsymbol{e}=\lambda e+\dot{e}\end{aligned} \tag{10}$$

where $\lambda$ is selected to ensure that the polynomial $e_A+\lambda$ is Hurwitz (Ge et al. 1999). The derivative of the augmented $s$ can be expressed as:

$$\begin{aligned}\dot{e_A} &= \lambda\dot{e}+\dot{e}\\ &= \lambda\dot{e}+\dot{Y}_d-\dot{Y}\\ &= \lambda\dot{e}+\dot{Y}_d-CA(\boldsymbol{S})-CB(\boldsymbol{S})u-Cd\\ &= D-CA(\boldsymbol{S})-CB(\boldsymbol{S})u-Cd\end{aligned} \tag{11}$$

where $D=\lambda\dot{e}+\dot{Y}_d$.

**Lemma:** For the system expressed by Eq. (9), considering that $B(\boldsymbol{S})>0$, the ideal control law to make the plant converge to desired states can be chosen as:

$$u^*=-\frac{1}{CB(\boldsymbol{S})}(CA(\boldsymbol{S})-D+Cd)-\left(\frac{\dot{B}(\boldsymbol{S})}{2CB(\boldsymbol{S})^2}-\frac{1}{\varepsilon CB(\boldsymbol{S})}-\frac{1}{\varepsilon CB(\boldsymbol{S})^2}\right)e_A \tag{12}$$

where $\varepsilon>0$ is the design parameter which is used to determine the converge rate of the tracking error, then $\lim_{n\to\infty}\|e\|=0$.

**Proof:** When the control law in Eq. (12) is substituted in the Eq. (11), the augmented term $s$ can be rewritten as:

$$\begin{aligned}\dot{e_A} &= D-CA(\boldsymbol{S})-CB(\boldsymbol{S})u-Cd\\ &= -\left(\frac{\dot{B}(\boldsymbol{S})}{2B(\boldsymbol{S})}+\frac{1}{\varepsilon}+\frac{1}{\varepsilon B(\boldsymbol{S})}\right)e_A\end{aligned} \tag{13}$$

Developing the Lyapunov function as $V=\frac{1}{2B(\boldsymbol{S})}{e_A}^2$, its derivative can be presented as:

$$\begin{aligned}\dot{V} &= \frac{1}{B(\boldsymbol{S})}e_A\dot{e_A}-\frac{\dot{B}(\boldsymbol{S})}{2B(\boldsymbol{S})^2}{e_A}^2\\ &= -\left(\frac{1}{\varepsilon B(\boldsymbol{S})}+\frac{1}{\varepsilon B(\boldsymbol{S})^2}\right){e_A}^2\end{aligned} \tag{14}$$

The above-mentioned function indicates that the augmented error $e_A$ would converge to zero. Correspondingly, $\lim_{n\to\infty}\|e\|=0$.

It is shown that the robust control law in Eq. (12) can be formulated as a function of the desired signals, the system states, and their relevant derivatives

$$\boldsymbol{z}=\left[\boldsymbol{S}\ e_A\ \frac{e_A}{\varepsilon}D\right]^{\mathrm{T}} \tag{15}$$

whereas, as can be seen in the above-mentioned equations, the corresponding matrices in the ship's motion equations (i.e. $A(\boldsymbol{S})$, $B(\boldsymbol{S})$, and $C$) are complicated. In addition, the various unknown disturbances acting on the ship add difficulty to achieve the model-based closed-loop control. Hence, to achieve the control law in Eq. (12), it is reasonable to utilize the online optimization algorithm to approximate the control laws.

The RBFNN was developed to approximate the unknown functions by adopting the radial basis functions as the activation functions (Broomhead and Lowe 1988). The scheme of the neural networks contains three layers: input layer, output layer, and hidden layer. The essential feature of the RBFNN-based control approach is that no prior knowledge of the dynamics of the plant is required to design the controller, and no offline training is required for the neural network (Ge et al. 2010). Therefore, the RBFNN can be applied in this study to approximate the function of control law as:

$$\begin{aligned}\hat{u} &= \hat{\omega}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{z})=[\hat{\omega}_1\Phi_1(\boldsymbol{z})+\hat{\omega}_2\Phi_2(\boldsymbol{z})+\cdots+\hat{\omega}_i\Phi_i(\boldsymbol{z})+\cdots+\hat{\omega}_m\Phi_m(\boldsymbol{z})]\\ \Phi(\boldsymbol{z})_i &= \exp\left(-\frac{\|\boldsymbol{z},-,\hat{\boldsymbol{c}}_i\|^2}{2{\sigma_{\mathrm{RBF}}}^2}\right)\end{aligned} \tag{16}$$

where $\hat{u}$ is the approximated value of ideal control law, $\hat{\omega}$ is the estimation of the ideal weights, $\Phi(\boldsymbol{z})_i$ is the radial basis function working as the activation function in the hidden layer, in which $\hat{\boldsymbol{c}}_i$ is the center and $\sigma_{\mathrm{RBF}}$ is the width representing the covering scope for the network input, and $\boldsymbol{\Phi}$ with $i$ subscripts denote the activation functions with the input $\boldsymbol{z}$.

Based on the introduction mentioned above, it is known that the unknown nonlinear plant can be controlled by the RBFNN-based controller with the input $\boldsymbol{z}$. The input matrix can be constructed by using the desired value and actual value. When the algorithm to update the neural network weights and centers is applied, the direct control method based on the RBFNN approximation can be used to provide control and make the plant converge to the desired states; the details are illustrated in Fig. 2.

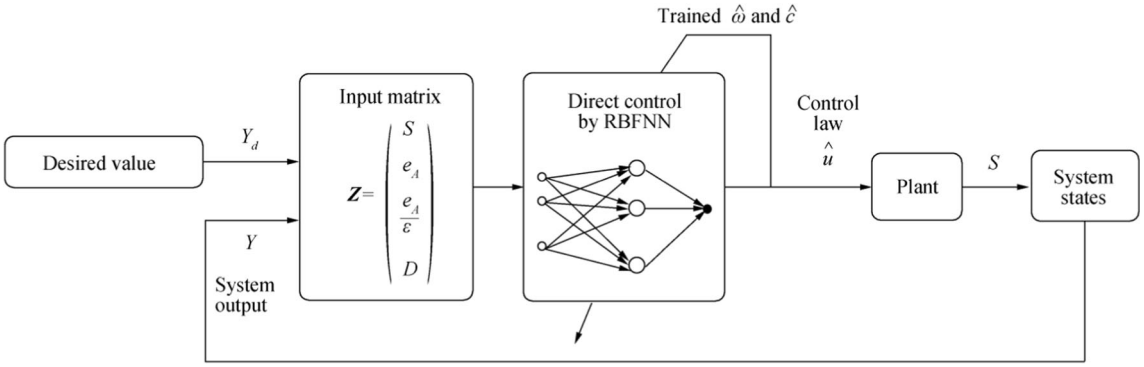



**Fig. 2** Architecture of the direct RBFNN control system

## 3 Control Algorithm Based on DEKF-Trained RBFNN

In this section, the DEKF training algorithm is introduced to train the RBFNN-based controller to improve the control performance with satisfactory design complications. The strengths of the DEKF training algorithm to the BP training method are analyzed mathematically as well.

### 3.1 RBFNN-Based Controller Trained by DEKF

The training algorithm for the RBFNN-based controller is presented as:

#### 3.1.1 Estimation of Weights

- Initialization

$$\hat{\boldsymbol{w}}_0 = E[\boldsymbol{w}_0]\boldsymbol{P}_0 = E\left[\left(\boldsymbol{w}_0-\hat{\boldsymbol{w}}_0\right)\left(\boldsymbol{w}_0-\hat{\boldsymbol{w}}_0\right)^{\mathrm{T}}\right] \tag{17}$$

where $\hat{\boldsymbol{w}}$ is the estimated weights initialized as small random value (e.g. magnitude of 0.1). $\boldsymbol{P}$ is the estimated error covariance of weights, and its initial value can be a diagonal matrix with diagonal components.

- Recursively executing with time interval $\Delta t$

Step W-1. Time updating

$$\widetilde{\boldsymbol{w}}_t = \hat{\boldsymbol{w}}_{t-\Delta t}\widetilde{\boldsymbol{P}}_t = \boldsymbol{P}_{t-\Delta t} \tag{18}$$

where $\boldsymbol{w}^{\sim}$ is the predicted weights and $\boldsymbol{P}^{\sim}$ is the predicted error covariance of weights.

Step W-2. Jacobian matrix calculation

$$\begin{aligned}\boldsymbol{H}_t &= \frac{\partial y_p}{\partial \boldsymbol{w}_t} = \frac{\partial y_p}{\partial y_m}\frac{\partial y_m}{\partial \boldsymbol{w}_t} \\ &= \left[\exp\left(-\left\|\frac{\boldsymbol{X}_t-\widetilde{\boldsymbol{c}}(1)\|^2}{2\sigma_{\mathrm{RBF}}{}^2}\right), \exp\left(-\left\|\frac{\boldsymbol{X}_t-\widetilde{\boldsymbol{c}}(2)\|^2}{2\sigma_{\mathrm{RBF}}{}^2}\right), \ldots, \exp\left(-\left\|\frac{\boldsymbol{X}_t-\widetilde{\boldsymbol{c}}(m)\|^2}{2\sigma_{\mathrm{RBF}}{}^2}\right)\right]\frac{\partial y_p}{\partial y_m}\end{aligned} \tag{19}$$

where $\boldsymbol{H}$ is the Jacobian matrix regarding neural network weights for DEKF estimation and $\frac{\partial y_p}{\partial y_m}$ is the Jacobian item representing the sensitivity between the plant input and output; more details about the implement are shown in Liu (2013).

Step W-3. Kalman gains matrix calculation

$$\boldsymbol{K}_t = \widetilde{\boldsymbol{P}}_t\boldsymbol{H}_t{}^{\mathrm{T}}\left[\boldsymbol{H}_t\widetilde{\boldsymbol{P}}_t\boldsymbol{H}_t{}^{\mathrm{T}} + \boldsymbol{R}^w\right]^{-1} \tag{20}$$

where $\boldsymbol{K}$ is the Kalman gain matrix for weights group, $\boldsymbol{R}^w$ is a diagonal matrix with components equal to or slightly less than 1.

Step W-4. Estimation of states

$$\hat{\boldsymbol{w}}_t = \widetilde{\boldsymbol{w}}_t + \boldsymbol{K}_t e_{Aw} \tag{21}$$

where $\hat{\boldsymbol{w}}$ is the estimated weights of the RBFNN-based controller and $e_{Aw}$ is the augmented deviation, which obeys the format in Eq. (11).

Step W-5. Error covariance updating

$$\boldsymbol{P}_t = \widetilde{\boldsymbol{P}}_t - \boldsymbol{K}_t\boldsymbol{H}_t{}^{\mathrm{T}}\widetilde{\boldsymbol{P}}_t + \boldsymbol{Q}^w \tag{22}$$

where $\boldsymbol{Q}^w$ is the matrix representing artificial process noise in the process of the weights' training. It is used to avoid numerical divergence.

#### 3.1.2 Estimation of Centers

- Initialization

$$\hat{\boldsymbol{c}}_0 = E[\boldsymbol{c}_0]\boldsymbol{G}_0 = E\left[\left(\boldsymbol{c}_0 - \hat{\boldsymbol{c}}_0\right)\left(\boldsymbol{c}_0 - \hat{\boldsymbol{c}}_0\right)^{\mathrm{T}}\right] \tag{23}$$

where $\hat{\boldsymbol{c}}$ is the estimated centers initialized as small random value. $\boldsymbol{G}$ is the estimated error covariance of centers initialized as a diagonal matrix with diagonal components.

- Recursively executing with time interval $\Delta t$
- Step C-1. Time updating

$$\widetilde{\boldsymbol{c}}_t = \hat{\boldsymbol{c}}_{t-\Delta t}\widetilde{\boldsymbol{G}}_t = \boldsymbol{G}_{t-\Delta t} \tag{24}$$

where $\widetilde{\boldsymbol{c}}$ is the predicted centers and $\widetilde{\boldsymbol{G}}$ is the predicted error covariance of centers.

Step C-2. Jacobian matrix calculation

$$\begin{aligned}\boldsymbol{J}_t &= \frac{\partial y_p}{\partial \boldsymbol{e}_t} = \frac{\partial y_p}{\partial y_m}\frac{\partial y_m}{\partial \boldsymbol{e}_t} \\ &= \left[\widetilde{\boldsymbol{w}}_t(1)\boldsymbol{\Phi}(1)\frac{\widetilde{\boldsymbol{c}}(1)-\boldsymbol{X}_t}{{\sigma_{\mathrm{RBF}}}^2}, \widetilde{\boldsymbol{w}}_t(2)\boldsymbol{\Phi}(2)\frac{\widetilde{\boldsymbol{c}}(2)-\boldsymbol{X}_t}{{\sigma_{\mathrm{RBF}}}^2}, \ldots, \widetilde{\boldsymbol{w}}_t(m)\boldsymbol{\Phi}(m)\frac{\widetilde{\boldsymbol{c}}(m)-\boldsymbol{X}_t}{{\sigma_{\mathrm{RBF}}}^2}\right]\frac{\partial y_p}{\partial y_m}\end{aligned} \tag{25}$$

where $\boldsymbol{J}$ is the Jacobian matrix regarding neural network centers for DEKF estimation.

Step C-3. Kalman gains matrix calculation

$$\boldsymbol{T}_t = \widetilde{\boldsymbol{G}}_t{\boldsymbol{J}_t}^{\mathrm{T}}\left[\boldsymbol{J}_t\widetilde{\boldsymbol{G}}_t{\boldsymbol{J}_t}^{\mathrm{T}} + \boldsymbol{R}^c\right]^{-1} \tag{26}$$

where $\boldsymbol{T}$ is the Kalman gain matrix for centers group and $\boldsymbol{R}^c$ is a diagonal matrix with components equal to or slightly less than 1.

Step C-4. Estimation of states

$$\hat{\boldsymbol{c}}_t = \widetilde{\boldsymbol{c}}_t + \boldsymbol{T}_t e_{Ac} \tag{27}$$

where $\hat{\boldsymbol{c}}$ is the estimated centers of RBFNN-based controller and $e_{Ac}$ is the augmented deviation, which obeys the format in Eq. (11).

Step C-5. Error covariance updating

$$\boldsymbol{G}_t = \widetilde{\boldsymbol{G}}_t - \boldsymbol{T}_t{\boldsymbol{J}_t}^{\mathrm{T}}\widetilde{\boldsymbol{G}}_t + \boldsymbol{Q}^c \tag{28}$$

where $\boldsymbol{Q}^c$ is the matrix representing artificial process noise in the process of the centers' training. It is used to avoid numerical divergence.

More details about the studies of the stability analysis and converging in the use of the EKF-based training algorithm can be referred in De and Yu (2007) and Wang and Huang (2011). Based on the above-mentioned formulas, the DEKF-based algorithm was used as the training algorithm by using the augmented system error to update the weights and centers of the RBFNN. As shown in Eqs. (21) and (27), the augmented error between the desired value and the actual value was adopted in the current iteration to approximate the weights and centers. After calculating the control law by using the proposed DEKF-trained RBFNN controller shown in Eq. (16), the actuator will make the plant converge to the desired value recursively.

### 3.2 Performance Analysis Between DEKF- and BP-Based Training Algorithm

It is well-known that the BP training algorithm is a classical method to set the weights and centers of RBFNN. Although the BP training method is effective in training neural networks, it only processes the present gradient descent procedure. That is to say, the gradient is calculated for the error surface merely based on the current state but not the whole set of the states, which are constantly changing under the control law. The BP training algorithm can be expressed as (Liu 2013):

$$\begin{aligned}\Delta \boldsymbol{w}_{\mathrm{BP}} &= -\eta^w\frac{\partial E}{\partial \boldsymbol{w}_{\mathrm{BP}}} \\ &= -\eta^w\frac{\partial E}{\partial y_p}\frac{\partial y_p}{\partial y_m}\frac{\partial y_m}{\partial \boldsymbol{w}_{\mathrm{BP}}} \\ &= \eta^w e_{Aw}\left[\exp\left(\frac{-\left\|\boldsymbol{X}_t - \boldsymbol{e}(1)\right\|^2}{2{\sigma_{\mathrm{RBF}}}^2}\right), \exp\left(\frac{-\left\|\boldsymbol{X}_t - \boldsymbol{e}(2)\right\|^2}{2{\sigma_{\mathrm{RBF}}}^2}\right), \ldots, \exp\left(\frac{-\left\|\boldsymbol{X}_t - \boldsymbol{e}(m)\right\|^2}{2{\sigma_{\mathrm{RBF}}}^2}\right)\right]\frac{\partial y_p}{\partial y_m} \\ \Delta \boldsymbol{c}_{\mathrm{BP}} &= -\eta^c\frac{\partial E}{\partial \boldsymbol{c}_{\mathrm{BP}}} \\ &= -\eta^c\frac{\partial E}{\partial y_p}\frac{\partial y_p}{\partial y_m}\frac{\partial y_m}{\partial \boldsymbol{c}_{\mathrm{BP}}} \\ &= \eta^c e_{Ac}\left[\boldsymbol{w}_t(1)\boldsymbol{\Phi}(1)\frac{\widetilde{\boldsymbol{c}}(1)-\boldsymbol{X}_t}{{\sigma_{\mathrm{RBF}}}^2}, \widetilde{\boldsymbol{w}}_t(2)\boldsymbol{\Phi}(2)\frac{\widetilde{\boldsymbol{c}}(2)-\boldsymbol{X}_t}{{\sigma_{\mathrm{RBF}}}^2}, \ldots, \widetilde{\boldsymbol{w}}_t(m)\boldsymbol{\Phi}(m)\frac{\widetilde{\boldsymbol{c}}(m)-\boldsymbol{X}_t}{{\sigma_{\mathrm{RBF}}}^2}\right]\frac{\partial y_p}{\partial y_m}\end{aligned} \tag{29}$$

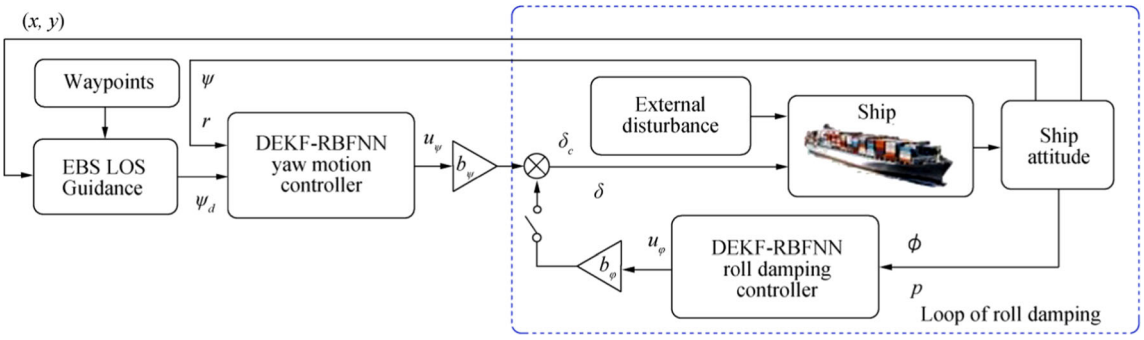



**Fig. 3** The scheme of the proposed rudder roll stabilization

where $\eta^w$ and $\eta^c$ are the learning rates of the neural networks and $s$ is the augmented error of the system. When the following two assumptions are taken, (1) $\tilde{\boldsymbol{P}} = p^w \boldsymbol{I}$, $\tilde{G} = p^c \boldsymbol{I}$ and (2) $\left[\boldsymbol{H}\tilde{\boldsymbol{P}}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R}^w\right]^{-1} = a^w \boldsymbol{I}$, $\left[\boldsymbol{J}\tilde{\boldsymbol{G}}\boldsymbol{J}^{\mathrm{T}} + \boldsymbol{R}^c\right]^{-1} = a^c \boldsymbol{I}$, the DEKF algorithm for neural network training can be simplified as the following form:

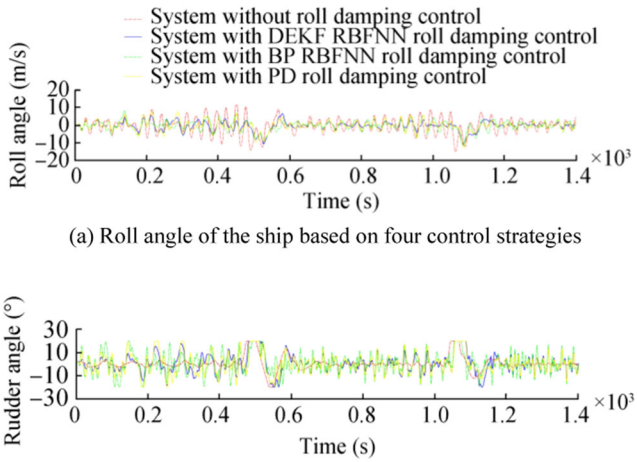
$$
\begin{aligned}
\Delta \boldsymbol{w}_{\mathrm{EKF}} &= \boldsymbol{K} e_{Aw} \\
&= \boldsymbol{P}\boldsymbol{H}^{\mathrm{T}}\left[\boldsymbol{H}\boldsymbol{P}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R}^w\right]^{-1} e_{Aw} \\
&= a^w p^w e_{Aw} \boldsymbol{H} \\
&= a^w p^w e_A \left[ \exp\left( \frac{-\left\| \boldsymbol{X}_t - \boldsymbol{c}(1) \right\|^2}{2\sigma_{\mathrm{RBF}}{}^2} \right), \exp\left( \frac{-\left\| \boldsymbol{X}_t - \boldsymbol{c}(2) \right\|^2}{2\sigma_{\mathrm{RBF}}{}^2} \right), \ldots, \exp\left( \frac{-\left\| \boldsymbol{X}_t - \boldsymbol{c}(m) \right\|^2}{2\sigma_{\mathrm{RBF}}{}^2} \right) \right] \frac{\partial y_p}{\partial y_m} \\
\Delta \boldsymbol{c}_{\mathrm{EKF}} &= \boldsymbol{T} e_{Ac} \\
&= \boldsymbol{G}\boldsymbol{J}^{\mathrm{T}}\left[\boldsymbol{J}\boldsymbol{G}\boldsymbol{J}^{\mathrm{T}} + \boldsymbol{R}^c\right]^{-1} e_{Ac} \\
&= a^c p^c e_{Ac} \boldsymbol{J} \\
&= a^c p^c e_{Ac} \left[ \tilde{\boldsymbol{w}}_t(1)\boldsymbol{\Phi}(1)\frac{\tilde{\boldsymbol{c}}(1) - \boldsymbol{X}_t}{\sigma_{\mathrm{RBF}}{}^2}, \tilde{\boldsymbol{w}}_t(2)\boldsymbol{\Phi}(2)\frac{\tilde{\boldsymbol{c}}(2) - \boldsymbol{X}_t}{\sigma_{\mathrm{RBF}}{}^2}, \ldots, \tilde{\boldsymbol{w}}_t(m)\boldsymbol{\Phi}(m)\frac{\tilde{\boldsymbol{c}}(m) - \boldsymbol{X}_t}{\sigma_{\mathrm{RBF}}{}^2} \right] \frac{\partial y_p}{\partial y_m}
\end{aligned}
\tag{30}
$$

Note that from the mathematical standpoint, the DEKF training algorithm can be interpreted as the degeneration of the BP training algorithm with $a^w p^w = \eta^w$ and $a^c p^c = \eta^c$.

The essentialities of the two above-mentioned assumptions can be implicated as follows. The first assumption means that the covariance of the weights and centers remains the diagonal form as its value is not changed during the propagation. Initially, this assumption is reasonable because the initial errors of the weights and centers do not have cross-correlations. However, during the training process, maintaining the diagonal forms of $\boldsymbol{P}$ and $\boldsymbol{G}$ means that the errors in the weights and centers are uncorrelated, which is unsound because the weights and centers in the previous steps affect the derivative value of the outputs which will contribute to the

**Table 1** The main characteristics of the full-scale container ship

| | |
|---|---|
| Length (m) | 175.00 |
| Breath (m) | 25.40 |
| Mean draft (m) | 8.00 |
| Displacement ($m^3$) | 21 222 |
| Block coefficient | 0.559 |
| Metacentric heights (m) | 0.3 |
| Rudder area ($m^2$) | 33.037 |
| Propeller diameter (m) | 6.533 |




(a) Roll angle of the ship based on four control strategies

(b) Rudder angle of the ship based on four control strategies

**Fig. 4** The roll motions and rudder actions of the ship sailing on trajectory

value of the following estimations. Hence, the off-diagonal forms are essential as they contain interrelated information for the training of weights and centers. Moreover, this assumption leads to the results that the weights and centers are updated with the same rate even though they need to be trained based on the changing rate of differences. Therefore, discarding the off-diagonal covariance has the significant impact on the training of weights and centers.

Based on the first assumption, the second assumption can be rewritten as:

$$\boldsymbol{H}\tilde{\boldsymbol{P}}\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R}^{w}=p^{w}\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}+\tau^{w}\boldsymbol{I}\boldsymbol{J}\tilde{\boldsymbol{G}}\boldsymbol{J}^{\mathrm{T}}+\boldsymbol{R}^{c}=p^{c}\boldsymbol{J}\boldsymbol{J}^{\mathrm{T}}+\tau^{c}\boldsymbol{I} \tag{31}$$

In order to achieve the assumptions which equal to $a^{w}\boldsymbol{I}$ and $a^{c}\boldsymbol{I}$, it is required that

$$\begin{aligned}\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}&=\gamma^{w}\boldsymbol{I}\\ \boldsymbol{J}\boldsymbol{J}^{\mathrm{T}}&=\gamma^{c}\boldsymbol{I}\end{aligned} \tag{32}$$

When the rows of matrices are orthogonal and the entries are $\sqrt{\gamma^{w}}$ and $\sqrt{\gamma^{c}}$, respectively, the assumptions will be satisfied with

$$\left[\boldsymbol{H}\tilde{\boldsymbol{P}}\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{R}^{w}\right]^{-1}=\frac{1}{p^{w}\gamma^{w}+\tau^{w}}\boldsymbol{I}\left[\boldsymbol{J}\tilde{\boldsymbol{G}}\boldsymbol{J}^{\mathrm{T}}+\boldsymbol{R}^{c}\right]^{-1}=\frac{1}{p^{c}\gamma^{c}+\tau^{c}}\boldsymbol{I} \tag{33}$$

However, these conditions cannot always be satisfied because the row entries of $\boldsymbol{H}$ and $\boldsymbol{J}$ are the partial derivatives of the outputs regarding the weights and centers in the proposed RBFNN.

Overall, when the DEKF algorithm is being simplified into the BP algorithm, the assumptions cannot always be guaranteed and would result in the discarding of some useful information in updating weights and centers. Therefore, the DEKF-based training algorithm is superior to the BP-based algorithm theoretically. The section of case studies will examine the comparative performance of BP- and DEKF-trained RBFNN control system.

## 4 DEKF RBFNN–Based Rudder Roll Damping Autopilot

The rudder roll stabilizer has been demonstrated to be a competent approach to reduce roll motions only using the rudder as the damping actuator (Perez and Blanke 2012). In comparison with the compact controller design, the separate controller design, which owns two controllers implemented in parallel, has fewer rudder actions and design complexity caused by the coupling items (Fang and Luo 2006). In this study, the rudder roll stabilization system is designed through developing and conducting the yaw motion controller and the roll damping controller separately. Based on the DEKF RBFNN control scheme, the autopilot control system concerning the objective of path following and roll damping is developed as shown in Fig. 3.

The enclosure-based steering line of sight (EBS LOS)–based guidance method (see Fig. 1) is employed in the control system. The dynamic desired course angle $\psi_d$ to track the trajectory and the tracking deviation $E_0$ can be calculated as:

$$\psi_d=\arctan\left(\frac{y_{\mathrm{los}}-y_c}{x_{\mathrm{los}}-x_c}\right) \tag{34}$$

$$E_0=(y_c-y_k)\cos(\alpha_k)-(x_c-x_k)\sin(\alpha_k) \tag{35}$$

where $(x_c, y_c)$ is the current position, $(x_{\mathrm{los}},\ y_{\mathrm{los}})$ is the position of EBS LOS point, $(x_k, y_k)$ is the previous pre-set waypoint, and $\alpha_k$ is the orientation of the pre-determined trajectory; more details can be seen in Fossen (2011).

In the proposed system, the roll damping controller will calculate the relevant control output based on the actual roll angle $\phi$ and roll rate $p$. Simultaneously, the actual position is utilized in the EBS LOS guidance block to calculate the instantaneous course angle. The actual yaw angle $\psi$ and yaw rate $r$ are compared with the desired yaw angle $\psi_d$ to build up the input matrix for the yaw motion controller. The ship's command rudder angle is equivalent to the summation of the control law from yaw motion controller and roll reduction controller with corresponding parameters:

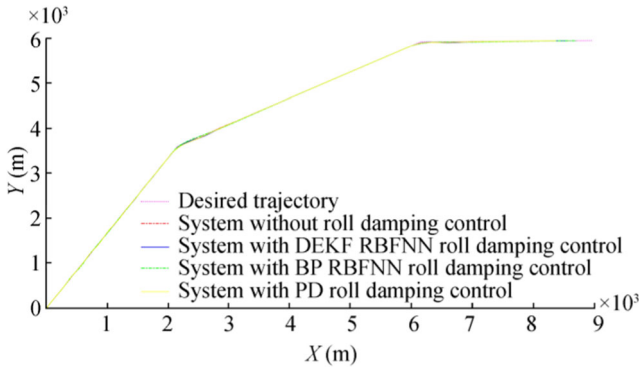



**Fig. 5** The trajectories of the vessel when following trajectory 1

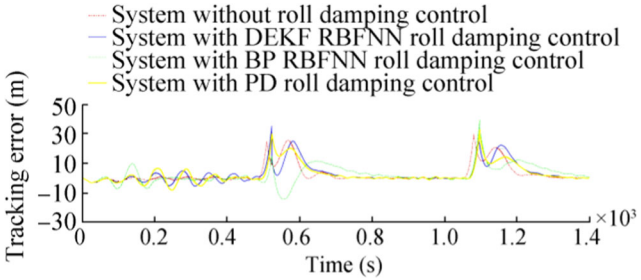



**Fig. 6** The path following error of the vessel advancing according to trajectory 1

**Table 2** The values of cost function and relevant roll damping percentage (sailing on trajectory 1)

| Controller types | Standard deviation of roll rate | Roll reduction percentage | Cost of roll motions | Cost of rudder actions |
|---|---|---|---|---|
| Without roll stabilizer | 0.01689 | N/A | 140 650 | 234 407 |
| DEKF RBFNN stabilizer | 0.00679 | 59.83 | 48 411 | 474 708 |
| BP RBFNN stabilizer | 0.00799 | 52.70 | 55 591 | 515 668 |
| PD stabilizer | 0.00995 | 41.08 | 61 926 | 521 293 |

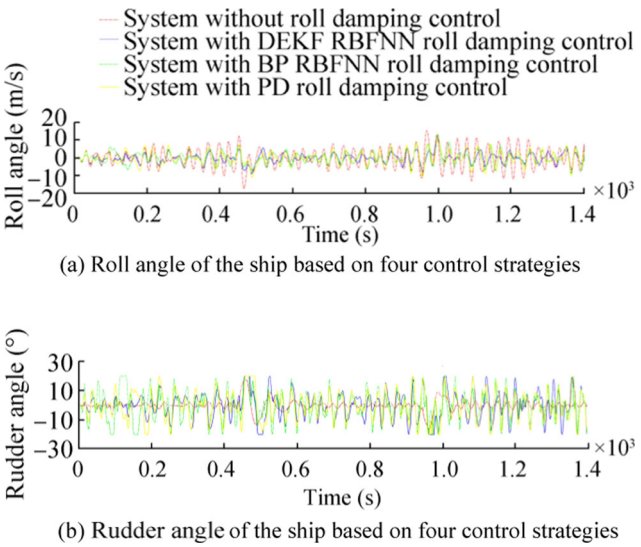
$$u_\psi = \hat{\boldsymbol{w}}_\psi{}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{z}_\psi) = \sum_{j=1}^{m}\hat{\boldsymbol{w}}_\psi(j)\exp\left(\frac{-\|\boldsymbol{z}_\psi-\hat{\boldsymbol{c}}_\psi(j)\|^2}{2\sigma_{\mathrm{RBF}}{}^2}\right)$$
$$u_\phi = \hat{\boldsymbol{w}}_\phi{}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{z}_\phi) = \sum_{j=1}^{m}\hat{\boldsymbol{w}}_\phi(j)\exp\left(-\frac{\|\boldsymbol{z}_\phi-\hat{\boldsymbol{c}}_\phi(j)\|^2}{2\sigma_{\mathrm{RBF}}{}^2}\right)\delta_c = b_\psi u_\psi + b_\phi u_\phi \tag{36}$$

where $u_\psi$ and $u_\phi$ are the approximated control laws for path following and roll reduction, $\boldsymbol{\Phi}$ are the neurons of the proposed RBFNN, $\boldsymbol{z}_\psi = [\psi,\ \lambda(\psi_d-\psi)+(\dot{\psi}_d-\dot{\psi})\ ,\ \frac{\lambda(\psi_d-\psi)+(\dot{\psi}_d-\dot{\psi})}{\varepsilon},\ \lambda(\dot{\psi}_d-\dot{\psi})+\psi_d]^{\mathrm{T}}$ and $\boldsymbol{z}_\phi = \left[\phi,\ \lambda(\phi_d-\phi)+(\dot{\phi}_d-\dot{\phi}),\ \frac{\lambda(\phi_d-\phi)+(\dot{\phi}_d-\dot{\phi})}{\varepsilon},\ \lambda(\dot{\phi}_d-\dot{\phi})+\phi_d\right]^{\mathrm{T}}$ (where $\phi_d = 0$ is selected in this study to show the roll stabilization) are the input of relevant controllers, $\hat{\boldsymbol{w}}_\psi$ and $\hat{\boldsymbol{w}}_\phi$ are the corresponding weights trained by the DEFK, $\hat{\boldsymbol{c}}_\psi$ and $\hat{\boldsymbol{c}}_\phi$ are the corresponding centers trained by the DEFK, $j$ is the total number of the neurons in the neural network. $\delta_c$ is the rudder command angle and $b_\psi$ and $b_\phi$ are the parameters reflecting the emphasis of control performance. Since the underactuation characteristic of the ship's motion control, the rudder is commonly utilized as the only actuator. The selection of the parameters is determined by the environmental conditions and stabilization requirements. In this study, the importance of the yaw control is assumed to be equivalent to that of the roll damping control; thus, the two parameters are assumed to be equivalent to each other (Fossen 2011).

To analyze the control capability of the proposed DEKF RBFNN–based rudder roll stabilization system, the evaluation items including roll reduction percentage, the cost functions of roll motion $C_{\mathrm{Roll}}$, and rudder angle $C_{\mathrm{Rudder}}$ (McGookin et al. 2000) are adopted and expressed as follows:

$$d_B = \sqrt{\frac{\sum_{k=0}^{M}\left(p_k^B\right)^2}{M-1}};\ d_A = \sqrt{\frac{\sum_{k=0}^{M}\left(p_k^A\right)^2}{M-1}}$$
$$P_{Reduction}\ (\%) = \frac{d_B-d_A}{d_B}\times 100\%$$
$$C_{Roll} = \sum_{k=0}^{M}\phi_k^2;\ C_{Rudder} = \sum_{k=0}^{M}\delta_k^2 \tag{37}$$

where $d_B$ is the standard deviation of roll rate $p^B$ without roll damping control and $d_A$ is the standard deviation of roll rate $p^A$ with roll damping control (Fossen 1994), $M$ is the amount of the total iterations, and $\phi_k$ and $\delta_k$ are the roll angle and rudder angle in the $k$th iteration respectively.




(a) Roll angle of the ship based on four control strategies

(b) Rudder angle of the ship based on four control strategies

**Fig. 7** The path following error of the vessel advancing according to trajectory 2

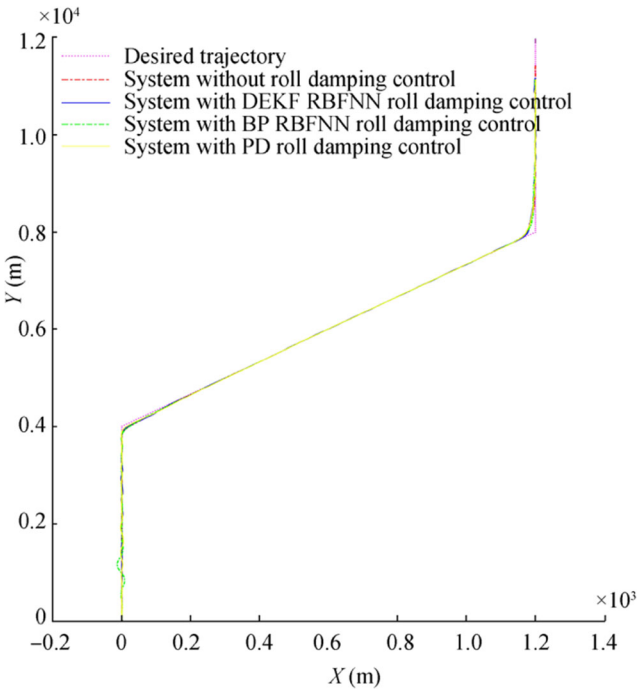



**Fig. 8** The trajectories of the vessel when following trajectory 2

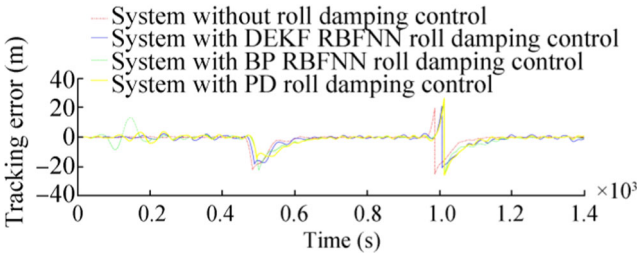



**Fig. 9** The path following error of the vessel sailing based on trajectory 2

## 5 Simulation Studies

In order to validate the efficiency of the rudder roll stabilization system based on DEKF-trained RBFNN, the four-DOF nonlinear mathematical model of a full-scale container ship is adopted in this study. The main characteristics of the relevant ship are outlined in Table 1. More details can be seen in Fossen (1994). The input and output of the plant (i.e. the surface vessel being controlled) are the rudder angle and angular motions, while the system states are the attitudes of the vessel.

### 5.1 Simulation Setting and Prior Training of the DEKF RBFNN–Based Controller

The Bogacki-Shampine algorithm was employed to solve the ODE of the ship's mathematical model with wave disturbances. The ship's propeller rotation rate was set at 80 r/min. From the perspective of engineering practice, the slew rate of rudder angle (i.e. motions of control actuator) was constrained within ±5 (°) /s regarding the characteristics of normal servo motors as well as the requirement of IMO (Oda et al. 2008), while the angle was correspondingly constrained to $\delta_{\max} = \pm 20°$ to avoid sharp fluctuations of rudder actions.

Two scenarios were carried out to investigate the performance of the proposed stabilization system: in the first scenario, the ship is requested to sail from the beginning waypoint (0, 0) to the next waypoint (3600, 2160), and then heading to the following point at (5920, 6120) before advancing to the waypoint (5920, 9600) (defined as trajectory 1) with initial state at $[\boldsymbol{\eta}_0, \boldsymbol{\nu}_0] = [0\ \text{m}, 0\ \text{m}, 0°, 30°, 8\ \text{m/s}, 0\ \text{m/s}, 0\ (°)/\text{s}, 0\ (°)/\text{s}]^{\text{T}}$; the second scenario (trajectory 2) is designed to make the ship sailing from the initial waypoint (0, 0) to (4000, 0), and then to (8000, 1200) before arriving at the waypoint (12 000, 1200) with initial state at $[\boldsymbol{\eta}_0, \boldsymbol{\nu}_0] = [0\ \text{m}, 0\ \text{m}, 0°, 0°, 8\ \text{m/s}, 0\ \text{m/s}, 0\ (°)/\text{s}, 0\ (°)/\text{s}]^{\text{T}}$. The characteristic of the environmental disturbance was selected as sea state 5.

The parameters of the proposed DEKF training method for approximating the weights and centers of the RBFNN are tuned by the experience and trial-and-error method as $\varepsilon = 0.5$, $\lambda = 4$, $\boldsymbol{R}_{\psi}^{w} = \boldsymbol{R}_{\psi}^{c} = \boldsymbol{I}_3$, $\boldsymbol{R}_{\phi}^{w} = \boldsymbol{R}_{\phi}^{c} = \boldsymbol{I}_3$, $\boldsymbol{Q}_{\psi}^{w} = \boldsymbol{Q}_{\psi}^{c} = 0.01 \times \boldsymbol{I}_9$, and $\boldsymbol{Q}_{\phi}^{w} = \boldsymbol{Q}_{\phi}^{c} = 0.00001 \times \boldsymbol{I}_9$.

In order to save the learning time for the DEKF RBFNN controller, the prior training was conducted. Nine neuron nodes were involved in the neural network scheme. In the present study, the prior case for the ship advancing based on the desired path is carried out to train the ship to learn how to sail on the desired trajectory while reducing roll motions. The 'convergent weight matrices' obtained after 100 s can then be selected as the initial matrices for the RBFNN–based roll stabilizer to handle the motions of the ship $\hat{\boldsymbol{w}}_{\psi 0} = [0.09, -0.001, -0.002, -0.003, -0.004, -0.003, -0.0022, -0.001, -0.0003]^{\text{T}}$; $\hat{\boldsymbol{w}}_{\phi 0} = [-0.17, -0.44, -0.77, -0.94, -0.75, -0.34, -0.03, 0.07, 0.05]^{\text{T}}$.

**Table 4** The comparison of computational burdens amongst DEKF-, UKF-, and PB-trained RBFNN controllers

| CPU frequency | Running time for each iteration (s) | |
|---|---|---|
| | EKF RBFNN controller | BP RBFNN controller |
| 2.60 G | $1.532 \times 10^{-4}$ | $1.490 \times 10^{-4}$ |
| 3.4 G | $0.715 \times 10^{-4}$ | $0.743 \times 10^{-4}$ |

### 5.2 Results and Discussion

In order to highlight the capability of the designed DEKF RBFNN–based rudder roll stabilization system, the BP RBFNN–based and proportional derivative (PD)–based rudder roll stabilization system (Wang et al. 2015) were employed in this study for comparison. Two different pre-set trajectories are proposed to investigate the roll damping and path following performance of the proposed control system.

For the first scenario, as shown in Fig. 4, without the roll damping controller, the DEKF RBFNN–based yaw motion controller is capable of maintaining the ship sailing on the desired path steadily. However, the roll angle of the ship is

**Table 3** The values of cost function and relevant roll damping percentage (sailing on trajectory 2)

| Controller types | Standard deviation of roll rate | Roll reduction percentage | Cost of roll motions | Cost of rudder actions |
|---|---|---|---|---|
| Without roll stabilizer | 0.02106 | N/A | 197 138 | 293 559 |
| DEKF RBFNN stabilizer | 0.00881 | 58.17 | 53 481 | 578 768 |
| BP RBFNN stabilizer | 0.01130 | 46.35 | 78 299 | 580 114 |
| PD stabilizer | 0.01299 | 38.30 | 93 876 | 636 885 |

huge due to environmental disturbances. When the roll damping control loop is activated, the tasks of path following and roll damping are achieved by using the rudder roll stabilization system synchronously. The figures also show that the DEKF RBFNN–based control system uses mild rudder actions, but it performs better in roll damping than that of the BP RBFNN- and PD-based control systems. It can be explained that the increase of yaw angle deviations is the price paid for roll reduction and added to the complicated coupling system. Figures 5 and 6 illustrate the trajectories and the corresponding deviations between desired trajectories and actual trajectories for four types of control systems (i.e. without roll damping control, with DEKF RBFNN roll damping control, with BP RBFNN roll damping control, and with PD roll damping control). It is shown that the ship is capable of sailing on the desired trajectory without significant tracking errors when using the control of the proposed rudder roll stabilization system.

The values of relevant evaluation items are summarized in Table 2. The roll damping percentages of the proposed rudder roll stabilization systems are calculated as 59.83%, 52.70%, and 41.08%, respectively. It is found that comparing with the BP RBFNN- and PD-based stabilizer, the DEKF RBFNN stabilizer is capable of providing effective rudder actions with better roll damping performance and path following accuracy.

A similar conclusion can be drawn from the second scenario with different wave encounter angles. In this process, the dynamic performances of the ship turning to both the port side and starboard side are considered. Figures 7, 8, and 9 show that the DEKF RBFNN–based rudder roll stabilization system is promising in making the vessel to track the desired path and to reduce the roll motion even subject to the waves on the beam sea. The values of relevant evaluation items are shown in Table 3. It can be observed that the roll damping rate of the DEKF RBFNN–based system (i.e. 58.17%) is larger than that of the BP RBFNN system at 46.35% and PD system at 38.30%. Thus, the roll damping and path following, as well as the efficiency, of rudder actions are validated.

In practice, the functionality of the autopilot is achieved by the embedded computer. Regarding the computational expenses, the active control running time per period can be used to evaluate the computational complexity. That is the elapsed time when running the control program at every interval. Table 4 outlines the computational expenses of the DEKF RBFNN roll stabilizer and the BP RBFNN roll stabilizer (Wang et al. 2015). It is indicated that the computing load of the proposed DEKF RBFNN–based control system approximately equals to that of the BP-based control system, since DEKF only needs to perform the calculation of Jacobian matrix in one integration. Running time for each iteration (s) is shown in Table 4.

Therefore, the competent control performance, the low computational overhead, and the efficiency of using actuators make the EKF RBFNN a good choice to design autopilot and rudder roll stabilizer. From the investigations concerning different trajectories and encounter angles of waves, the EKF RBFNN–based rudder roll stabilization system was demonstrated to be effective in maintaining the ship advancing on the desired path and compensating the huge roll motions at the same time.

## 6 Conclusion

In this paper, the DEKF RBFNN algorithm has been proposed to develop the rudder roll stabilization system, which contains the yaw motion controller and the roll reduction controller implemented in parallel. The rudder roll stabilization system incorporated with the nonlinear mathematical model had been used to verify the control performance of the ship by only using the rudder as the steering actuator. It is found that the designed control system is feasible to maintain the surface vessel advancing on the pre-set path while reducing the roll motion at the same time. Comparing with the BP RBFNN–based control system, the DEKF RBFNN–based system has faster converge speed and better robustness against the external disturbances. It is worth noting that the designed DEKF RBFNN–based control algorithm has promising control performance but equivalent computational cost as it avoids some complicated integration calculations. Therefore, the DEKF RBFNN control algorithm is competent to design the rudder roll stabilization system with qualified capability in reducing roll motions and following path.

Further investigations will focus on the validation of the DEKF RBFNN–based rudder roll stabilization autopilot through experimental approaches by using the free running model scaled vessel ‘Hoorn’ (Wang et al. 2017b). To further improve the accuracy of the estimations and the robustness in coping with the environmental variables, other competent training algorithms are going to be investigated to design the neural network–based control system for vessels.

**Funding** This research is a part of the project titled ‘Intelligent Control for Surface Vessels Based on Kalman Filter Variants Trained Radial Basis Function Neural Networks’ partially funded by the Institutional Grants Scheme (TGRS 060515) of Tasmania, Australia.

## References

Alarçïn F (2007) Internal model control using neural network for ship roll stabilization. J Mar Sci Technol 15(2):141–147. https://doi.org/10.6119/JMST

Broomhead DS, Lowe D (1988) Radial basis functions, multi-variable functional interpolation and adaptive networks. Royal Signals and Radar Establishment Malvern, London, 39

Choi J, Lima ACC, Haykin S (2005) Kalman filter-trained recurrent neural equalizers for time-varying channels. IEEE Trans Commun 53(3):472–480. https://doi.org/10.1109/TCOMM.2005.843416

De JRJ, Yu W (2007) Nonlinear system identification with recurrent neural networks and dead-zone Kalman filter algorithm. Neurocomputing 70(13–15):2460–2466. https://doi.org/10.1016/j.neucom.2006.09.004

Fang MC, Luo JH (2006) A combined control system with roll reduction and track keeping for the ship moving in waves. J Ship Res 50(4):344–354

Fang MC, Luo JH (2007) On the track keeping and roll reduction of the ship in random waves using different sliding mode controllers. Ocean Eng 34(3):479–488. https://doi.org/10.1016/j.oceaneng.2006.03.004

Fang MC, Zhuo YZ, Lee ZY (2010) The application of the self-tuning neural network PID controller on the ship roll reduction in random waves. Ocean Eng 37(7):529–538. https://doi.org/10.1016/j.oceaneng.2010.02.013

Fang MC, Lin YH, Wang BJ (2012) Applying the PD controller on the roll reduction and track keeping for the ship advancing in waves. Ocean Eng 54:13–25. https://doi.org/10.1016/j.oceaneng.2012.07.006

Fossen TI (1994) Guidance and control of ocean vehicles. Wiley, New York, 48, 302, 440

Fossen TI (2011) Handbook of marine craft hydrodynamics and motion control, vol 243. Wiley, New York, 433

Ge SS, Hang CC, Tao Z (1999) A direct method for robust adaptive nonlinear control with guaranteed transient performance. Syst Control Lett 37(5):275–284. https://doi.org/10.1016/S0167-6911(99)00032-8

Ge SS, Hang CC, Lee TH, Tao Z (2010) Stable adaptive neural network control. Springer, New York, 44

Goh SL, Mandic DP (2007) An augmented extended Kalman filter algorithm for complex-valued recurrent neural networks. Neural Comput 19(4):1039–1055. https://doi.org/10.1162/neco.2007.19.4.1039

Ko CN, Lee CM (2013) Short-term load forecasting using SVR (support vector regression)-based radial basis function neural network with dual extended Kalman filter. Energy 49:413–422. https://doi.org/10.1016/j.energy.2012.11.015

Li H, Guo C, Li X (2010) Ship roll stabilization using supervision control based on inverse model wavelet neural network. Proceedings of the 8th World Congress on Intelligent Control and Automation, Jinan, China, 4829–4833. https://doi.org/10.1109/WCICA.2010.5554721

Liu J (2013) Radial basis function (RBF) neural network control for mechanical systems: design, analysis and Matlab simulation. Springer, New York, 58–60

McGookin EW, Murray-Smith DJ, Li Y, Fossen TI (2000) Ship steering control system optimisation using genetic algorithms. Control Eng Pract 8(4):429–443. https://doi.org/10.1016/S0967-0661(99)00159-8

Medagam PV, Pourboghrat, F (2009) Optimal control of nonlinear systems using RBF neural network and adaptive extended Kalman filter. Proceedings of the American Control Conference 2009, Hyatt Regency Riverfront, USA, 355–360. https://doi.org/10.1109/ACC.2009.5160105

Nejim S (2000) Rudder roll damping system for ships using fuzzy logic control. Proceedings of the OCEANS 2000 MTS/IEEE Conference and Exhibition, Providence, USA, 1137–1143. https://doi.org/10.1109/OCEANS.2000.881755

Nouri K, Dhaouadi R, Braiek NB (2008) Adaptive control of a nonlinear dc motor drive using recurrent neural networks. Appl Soft Comput 8(1):371–382. https://doi.org/10.1016/j.asoc.2007.03.002

Oda H, Ohtsu K, Sato H, Kanehiro K (2008) Designing advanced rudder roll stabilization system. Proceedings of the 7th JFPS International Symposium on Fluid Power, Toyama, Japan, 169–174

Perez T, Blanke M (2012) Ship roll damping control. Annu Rev Control 36(1):129–147. https://doi.org/10.1016/j.arcontrol.2012.03.010

Sanchez EN, Alanís AY, Loukianov AG (2008) Discrete-time high order neural control: trained with Kalman filtering, vol 112. Springer, New York, 332

Sun B, Zhu D, Yang SX (2014) A bioinspired filtered backstepping tracking control of 7000-m manned submarine vehicle. IEEE Trans Ind Electron 61(7):3682–3693. https://doi.org/10.1109/TIE.2013.2267698

Treakle TW, Mook DT, Liapis SI, Nayfeh AH (2000) A time-domain method to evaluate the use of moving weights to reduce the roll motion of a ship. Ocean Eng 27(12):1321–1343. https://doi.org/10.1016/S0029-8018(99)00051-7

Van AJ, Van NLH (1978) Optimum steering of ships with an adaptive autopilot. Proceedings of the Fifth Ship Control Systems Symposium, Annapolis, USA, 8–17

Wang X, Huang Y (2011) Convergence study in extended Kalman filter-based training of recurrent neural networks. IEEE Trans Neural Netw 22(4):588–600. https://doi.org/10.1109/TNN.2011.2109737

Wang Y, Chai S, Khan F, Nguyen HD (2015) Radial basis function neural network based rudder roll stabilization for ship sailing in waves. Proceedings of the 5th Australian Control Conference, Gold coast, Australia, 256–261

Wang Y, Chai S, Khan F, Nguyen HD (2017a) Unscented Kalman filter trained neural networks based rudder roll stabilization system for ship in waves. Appl Ocean Res 68:26–38. https://doi.org/10.1016/j.apor.2017.08.007

Wang Y, Chai S, Nguyen HD (2017b) Modelling of a surface vessel from free running test using low cost sensors. Proceedings of the 3rd International Conference on Control, Automation and Robotics, Nagoya, Japan, 299–303. https://doi.org/10.1109/ICCAR.2017.7942707

Zhang XK, Jin YC, Yang C, Zhang L (2006) A kind of robust rudder roll-damping system. Paper presented at the Systems and Control in Aerospace and Astronautics. Proceedings of the 1st International Symposium on Systems and Control in Aerospace and Astronautics, Harbin, China, 1151–1154. https://doi.org/10.1109/ISSCAA.2006.1627570

Zhao J, Zhu X, Wang W, Liu Y (2013) Extended Kalman filter-based Elman networks for industrial time series prediction with GPU acceleration. Neurocomputing 118:215–224. https://doi.org/10.1016/j.neucom.2013.02.031