Title	A study on the implementation of nonlinear Kalman filter applying MMG model
Author(s)	Koike, Hiroaki; Dostal, Leo; Sawada, Ryohei et al.
Citation	Journal of Marine Science and Technology (Japan). 2023, 28(4), p. 733-745
Version Type	VoR
URL	https://hdl.handle.net/11094/93290
rights	This article is licensed under a Creative Commons Attribution 4.0 International License.
Note	

ORIGINAL ARTICLE

A study on the implementation of nonlinear Kalman filter applying MMG model

Hiroaki Koike[1] · Leo Dostal[2] · Ryohei Sawada[3] · Yoshiki Miyauchi[1] · Atsuo Maki[1]

Abstract

Many technologies need to be established to realize autonomous ships. In particular, accurate state estimation in real time is one of the most important technologies. In the ship and ocean engineering fields, there have been many studies on state estimation using nonlinear Kalman filters. Several methods have been proposed for nonlinear Kalman filters. However, there is insufficient verification on the selection of which filter should be applied among them. Therefore, this study aims to validate the filter selection to provide a guideline for filter selection. The effects of modeling error, observation noise, and type of maneuvers on the estimation accuracy of the unscented Kalman filter (UKF) and ensemble Kalman filter (EnKF) used in this study were investigated. In addition, it was verified whether filtering could be performed in real time. The results show that modeling error significantly impacts the estimation accuracy of the UKF and EnKF. However, the observation noise and types of maneuvers did not have an impact like the modeling error. Thus, we obtained the guideline that UKF and EnKF should be used differently depending on the required computation time. We also obtained that keeping the modeling error sufficiently small is essential to improving the estimation accuracy.

Keywords Kalman filter · UKF · EnKF · MMG model · State estimation

1 Introduction

The number of ship crews engaged in coastal shipping is aging, and there has been a shortage of manpower in recent years [1]. One of the solutions to this problem is the autonomous ship. Significant studies have been conducted across the globe on this topic to solve the various challenges associated with autonomous ships. For instance, automatic berthing/unberthing experiments using actual vessels were successfully conducted in the 1980s in Japan, and research on automatic berthing/unberthing has a long research history [2].

✉ Hiroaki Koike
hiroaki_koike@naoe.eng.osaka-u.ac.jp

Atsuo Maki
maki@naoe.eng.osaka-u.ac.jp

1 Osaka University, 2-1 Yamadaoka, Suita, Osaka, Japan

2 Institute of Mechanics and Ocean Engineering, Hamburg University of Technology, 21043 Hamburg, Germany

3 National Maritime Research Institute, 6-38-1, Shinkawa, Mitaka, Tokyo 181-0004, Japan

Several issues need to be resolved in establishing this automatic berthing/unberthing technology. For instance, establishing an accurate maneuvering motion model, establishing optimal berthing paths, and developing automatic berthing/unberthing algorithms [3–5].

To achieve automatic berthing and unberthing, accurate state estimation must be performed within the control period. Developing a real-time accurate state estimation method is one of the challenges. State estimation in this research refers to the estimation of states in a system from observed values obtained using sensors, filters, and other methods.

To avoid damage to the ship and pier when a vessel contacts the pier during berthing, the speed at which the vessel hits the pier, i.e., the berthing speed, should be sufficiently slow [6]. Roubos conducted a statistical study of berthing speed in Rotterdam. The results showed that the average berthing speed in 555 examples of common merchant vessels, including container ships, tankers, and bulk carriers, was approximately 4 [cm/s] [7]. In addition, some ports have established the lateral (sway) velocity as a rule. For instance, at the Kinjo Wharf in the Port of Nagoya in Japan, large passenger ships are required to berth at no more than 8 [cm/s] [8]. As described above, it is essential for safety to

estimate the ship states with the highest possible accuracy to keep the berthing speed smaller than the criteria, even when performing automatic berthing. A commonly used method for estimating states in real time is the Kalman filter [9, 10]. The Kalman filter is a kind of sequential Bayesian filter proposed by Kalman. It has proven to be optimal in the sense that it minimizes the sum of squares of the estimation error within the assumptions of noise normality and linearity of the state equation [11]. The Kalman filter has been applied for state estimation in various engineering fields, and many applications of nonlinear Kalman filters can be applied to even nonlinear systems for vehicles [12, 13], space crafts [14–16], and robots [17]. It is worth noting that the extended Kalman filter (EKF), one of the nonlinear Kalman filter algorithms, was implemented on the Apollo spacecraft for a manned flight to the moon and used to accurately calculate the spacecraft's position [18].

In naval architecture and ocean engineering fields, there have been many studies on state estimation using nonlinear Kalman filters. For instance, Balchen used the EKF for state estimation using the dynamic positioning system (DPS) [19]. Sorensen also used Kalman filter for DPS [20]. Abkowitz optimized the maneuvering model using Kalman filter [21]. Fossen et al. used EKF to estimate the maneuvering motion model [22]. Ochiai applied EKF to estimate the attitude of the ship, future position, and future velocity using the data of global positioning system (GPS) and automatic radar plotting aids [23]. Miyoshi used EKF to estimate ship position from GPS data for tracking control. They also used EKF to estimate sway velocity, which is not actually observed and verified its performance in a real ship test [24]. S.Fossen and T.I. Fossen estimated the ship's position and speed from AIS data using EKF [25]. T.I. Fossen also calculated course over ground and speed over ground from Global Navigation Satellite System data using EKF [26]. However, many studies using nonlinear filters other than EKF have been conducted. For instance, Subchan et al. estimated the hydrodynamic forces and moments from the experimental results of free-running tests using an unscented Kalman filter (UKF). The estimated hydrodynamic forces and moments were then used to estimate the hydrodynamic derivatives included in the maneuvering equation of motion [27]. Deng et al. designed a model predictive controller for the DPS system that simultaneously estimates states and determines input control using a UKF-based observer [28].

Several nonlinear Kalman filters, including EKF, have been proposed, and the user must select a suitable filter among them according to the user's purpose. In addition, there have been many studies comparing the performance among filters. For instance, Konatowski compared the estimation accuracy of EKF and UKF for constant velocity and complex motion systems. The results showed that

UKF exhibited better estimation accuracy [29]. In addition, Konatowski conducted a comparison of the estimation accuracy of the EKF, UKF, and particle filter (PF) using a simulation of an object moving in a two-dimensional Cartesian coordinate system. Consequently, he showed that PF exhibited the best estimation accuracy [30]. In addition, Chen et al. compared the estimation accuracy when using the EKF, UKF, PF, and linear minimum mean square error filter as filters applied to the tracking filter [31]. The tracking filter is an algorithm that estimates the accurate value of the target's position and velocity based on the observed values of the target object, including the observation errors obtained from the sensors. Thus, many studies have compared the performance of nonlinear filters. Several studies on the implementation of Kalman filters have also been conducted. For instance, Nishimura presented the algorithms for evaluating the effect of errors due to modeling errors in the Linear Kalman filter [32]. However, there has not been any attempt to discuss guidelines for which filter to use in each case. This research assumed the implementation of a nonlinear Kalman filter with a complex motion model applied to an actual ship system. It differs from other papers in that it examines possible problems in implementing such a filter and provides guidelines for filter selection.

The contributions of this study are summarized as follows.

– This study provides guidelines to implement UKF and EnKF to guide users in state estimation.
– The following three aspects are examined for their impact on filter estimation accuracy for the implementation.

  1. Effect of modeling error
  2. Effect of observation noise
  3. Effect of type of maneuvering motion

– Verifying the real-time performance of each filtering method

2 Notation

The notation used in this paper is explained as follows. First, $\mathbb{R}^n$ denotes a Euclidean space of dimension n. The hat (ˆ) denotes the estimated value, and the superscript minus (¯) denotes the prior estimate. The superscript minus denotes prior estimates. A diagonal matrix $A \in \mathbb{R}^{n \times n}$ is denoted as $A = \mathrm{diag}(a_{11}, a_{22}, \ldots, a_{nn})$. Here, a_{ij} represents the elements of the i rows and j columns of the matrix. In addition, the

superscript T denotes the transpose of the matrix. Vectors are column vectors unless otherwise stated.

3 Overview of the Kalman filter

This section outlines the nonlinear Kalman filter used in this study. First, we consider the following discrete nonlinear system:

$$\begin{cases} \boldsymbol{x}(k+1) = \boldsymbol{f}(\boldsymbol{x}(k), \boldsymbol{u}(k)) + \boldsymbol{v}(k) \\ \quad \boldsymbol{y}(k) = \boldsymbol{H}\boldsymbol{x}(k) + \boldsymbol{w}(k). \end{cases} \tag{1}$$

$\boldsymbol{f}(\boldsymbol{x}(k), \boldsymbol{u}(k)) : \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}^n$ is the nonlinear system model. $\boldsymbol{H} \in \mathbb{R}^{m \times n}$ is output matrix. Here, k denotes the number of steps at a given time t_k, denoted as $t_k = \Delta t k, (k = 0, 1, 2, \ldots)$; Δt denotes the time interval of one time step; $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector; $\boldsymbol{u} \in \mathbb{R}^l$ is the input vector; $\boldsymbol{y} \in \mathbb{R}^m$ is the observation vector; $\boldsymbol{v} \in \mathbb{R}^n$ is the system noise according to $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, \mathbf{Q}(k))$; $\boldsymbol{w} \in \mathbb{R}^m$ is the additive observation noise according to $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \mathbf{R}(k))$. The covariance matrix $\mathbf{Q}, \mathbf{R}$ are defined as follows:

$$\begin{aligned} E[\mathbf{v}(i)\mathbf{v}^T(j)] &= \delta_{ij}\mathbf{Q}(I) \\ E[\mathbf{w}(i)\mathbf{w}^T(j)] &= \delta_{ij}\mathbf{R}(I), \quad \forall i, j \\ E[\mathbf{v}(i)\mathbf{w}^T(j)] &= \boldsymbol{0}. \end{aligned} \tag{2}$$

Here, δ_{ij} is the Kronecker delta.

3.1 UKF [33–35]

UKF is a nonlinear Kalman filter that uses a nonlinear transformation called the "U transform." This transformation method performs a nonlinear transformation of sample points called sigma points and calculates the expected value and variance from the sample mean of the transformed sigma points. Using this transformation, UKF has the advantage that, unlike EKF, it does not require local linearization using the Jacobian matrix [33]. The following is the computational procedure of UKF.

First, $2n + 1$ sigma points $\chi_i \in \mathbb{R}^n$ are determined using the state estimate $\hat{\boldsymbol{x}}(k-1)$ at $k-1$ and the error covariance matrix $\boldsymbol{P}(k-1) \in \mathbb{R}^{n \times n}$:

$$\begin{cases} \chi_0(k-1) = \hat{\boldsymbol{x}}(k-1) \\ \chi_i(k-1) = \hat{\boldsymbol{x}}(k-1) + (\sqrt{(n+\kappa)\boldsymbol{P}(k-1)})_i, \\ \quad for\ i = 1, 2, \ldots, n \\ \chi_{i+n}(k-1) = \hat{\boldsymbol{x}}(k-1) - (\sqrt{(n+\kappa)\boldsymbol{P}(k-1)})_i, \\ \quad for\ i = 1, 2, \ldots, n \end{cases} \tag{3}$$

$(\sqrt{\boldsymbol{P}(k-1)})_i \in \mathbb{R}^n$ denotes the i-th column of the square root matrix of the error covariance matrix $\boldsymbol{P}(k-1)$. Here, the square root matrix refers to the matrix $\boldsymbol{B} \in \mathbb{R}^{n \times n}$ that satisfies the relation $\boldsymbol{A} = \boldsymbol{B}\boldsymbol{B}$ for some positive definite symmetric matrix $\boldsymbol{A}$. This $\boldsymbol{B}$ is written as $\boldsymbol{B} = \sqrt{\boldsymbol{A}}$. For each sigma point, the weight $W_i \in \mathbb{R}(i = 0, 1, 2, \ldots, 2n)$ are determined as follows:

$$\begin{cases} W_0 = \dfrac{\kappa}{n+\kappa} \\ W_i = \dfrac{1}{2(n+\kappa)}. \end{cases} \tag{4}$$

Here, $\kappa \in \mathbb{R}$ is the design parameter for approximating the statistical moment μ_r of second or higher order. The statistical moment μ_r is defined in Eq.(5) and represents the r-order moment around the mean μ:

$$\mu_r = E[(X_p - \mu)^r]. \tag{5}$$

where X_p is the random variable for the states. Next, the sigma points are updated using the system model $\boldsymbol{f}(\boldsymbol{x}(k), \boldsymbol{u}(k))$:

$$\chi_i^-(k) = \boldsymbol{f}(\chi_i(k-1), \boldsymbol{u}(k)). \tag{6}$$

By this operation, the prior state estimate $\hat{\boldsymbol{x}}^-(k)$ and prior error covariance matrix $\boldsymbol{P}^-(k)$ are obtained from the updated sigma points as follows:

$$\begin{cases} \hat{\boldsymbol{x}}^-(k) = \displaystyle\sum_{i=0}^{2n} W_i \chi_i^-(k) \\ \boldsymbol{P}^-(k) = \displaystyle\sum_{i=0}^{2n} W_i \{\chi_i^-(k) - \hat{\boldsymbol{x}}^-(k)\} \{\chi_i^-(k) - \hat{\boldsymbol{x}}^-(k)\}^T \\ \quad + \mathbf{Q}(k). \end{cases} \tag{7}$$

However, the sigma points are updated using the prior error covariance matrix $\boldsymbol{P}^-$ as follows:

$$\begin{cases} \chi_0^-(k) = \hat{\boldsymbol{x}}^-(k) \\ \chi_i^-(k) = \hat{\boldsymbol{x}}^-(k) + \sqrt{(n+\kappa)\boldsymbol{P}^-(k)}_i \\ \chi_{i+n}^-(k) = \hat{\boldsymbol{x}}^-(k) - \sqrt{(n+\kappa)\boldsymbol{P}^-(k)}_i. \end{cases} \tag{8}$$

Update the output sigma point $\Upsilon_i^-(k) \in \mathbb{R}^m$ from the updated sigma point:

$$\Upsilon_i^-(k) = \boldsymbol{H}\chi_i^-(k). \tag{9}$$

From these output sigma points, a prior output estimate $\hat{\boldsymbol{y}}^-(k)$ is obtained:

$$\hat{\boldsymbol{y}}^-(k) = \sum_{i=0}^{2n} W_i \boldsymbol{\Upsilon}_i^-(k). \tag{10}$$

The prior output error covariance matrix $\boldsymbol{P}_{yy}^-(k) \in \mathbb{R}^{m \times m}$ is obtained from the output sigma points and prior output estimates $\hat{\boldsymbol{y}}^-$:

$$\boldsymbol{P}_{yy}^-(k) = \sum_{i=0}^{2n} W_i \{\boldsymbol{\Upsilon}_i^-(k) - \hat{\boldsymbol{y}}^-(k)\} \{\boldsymbol{\Upsilon}_i^-(k) - \hat{\boldsymbol{y}}^-(k)\}^{\mathrm{T}} \\ + \boldsymbol{R}(k). \tag{11}$$

The prior state output error covariance matrix $\boldsymbol{P}_{xy}^-(k) \in \mathbb{R}^{m \times n}$ is obtained as follows:

$$\boldsymbol{P}_{xy}^-(k) = \sum_{i=0}^{2n} W_i \{\boldsymbol{\chi}_i(k) - \hat{\boldsymbol{x}}^-(k)\} \{\boldsymbol{\Upsilon}_i^-(k) - \hat{\boldsymbol{y}}^-(k)\}^{\mathrm{T}}. \tag{12}$$

The Kalman gain $\boldsymbol{G}(k) \in \mathbb{R}^{n \times m}$ is obtained from $\boldsymbol{P}_{yy}^-(k)$ and $\boldsymbol{P}_{xy}^-(k)$:

$$\boldsymbol{G}(k) = \frac{\boldsymbol{P}_{xy}^-(k)}{\boldsymbol{P}_{yy}^-(k)}. \tag{13}$$

The state estimate $\hat{\boldsymbol{x}}(k)$ is obtained using the above Kalman gain $\boldsymbol{G}(k)$, prior state estimate $\hat{\boldsymbol{x}}^-(k)$, prior output estimate $\hat{\boldsymbol{y}}^-(k)$, and observables $\boldsymbol{y}(k)$, as shown in Eq.(14). The posterior error covariance matrix $\boldsymbol{P}(k) \in \mathbb{R}^{n \times n}$ is also calculated using the prior estimation error covariance matrix $\boldsymbol{P}^-(k)$, Kalman gain $\boldsymbol{G}(k)$, prior state output error covariance matrix $\boldsymbol{P}_{xy}^-(k)$, and measurement values $\boldsymbol{y}(k)$, as shown in Eq.(14):

$$\begin{cases} \hat{\boldsymbol{x}}(k) = \hat{\boldsymbol{x}}^-(k) + \boldsymbol{G}(k)(\boldsymbol{y}(k) - \hat{\boldsymbol{y}}^-(k)) \\ \boldsymbol{P}(k) = \boldsymbol{P}^-(k) - \boldsymbol{G}(k)(\boldsymbol{P}_{xy}^-(k)). \end{cases} \tag{14}$$

3.2 EnKF [36–38]

One of the nonlinear Kalman filters, EnKF, is a filter that uses the probability of the system state distribution held by a set called the ensemble members $\{\boldsymbol{x}^{(i)}(k)\}_{i=1}^N$. The method then updates the ensemble members using the update rule of the Kalman filter each time an observation value is obtained. This method, like UKF, does not require the computation of a Jacobian matrix and has the advantage that it can be applied to non-differentiable state-space models. The following summarizes the computation steps of EnKF. First, N sets of ensemble members $\boldsymbol{x}^{(i)}(k-1) \in \mathbb{R}^n$ ($i = 1, 2, \ldots, N$) are updated based on the system model. Moreover, $\boldsymbol{v}^{(i)}(k-1) \in \mathbb{R}^n (i = 1, 2, \ldots, N)$ represents the ensemble member of the system noise:

$$\begin{cases} \boldsymbol{x}^{(i)}(k-1) = \boldsymbol{f}(\boldsymbol{x}^{(i)}(k-1), \boldsymbol{u}(k-1), \boldsymbol{v}^{(i)}(k-1)), \\ \boldsymbol{v}^{(i)}(k-1) \sim \mathcal{N}(0, \boldsymbol{Q}(k-1)). \end{cases} \tag{15}$$

Here, the system model $\boldsymbol{f}$ is treated as Eq.(1) for convenience because it gives disturbance when updating ensemble members in EnKF:

$$\boldsymbol{f}(\boldsymbol{x}(k), \boldsymbol{u}(k), \boldsymbol{v}(k)) = \boldsymbol{f}(\boldsymbol{x}(k), \boldsymbol{u}(k)) + \boldsymbol{v}(k). \tag{16}$$

Next, the sample variance-covariance matrix $\hat{\boldsymbol{V}}(k-1)$ of the state variable vector $\boldsymbol{x}^{(i)-}(k-1)$ and the sample variance-covariance matrix $\hat{\boldsymbol{R}}(k-1)$ of the observed noise are calculated from the updated ensemble members:

$$\begin{cases} \boldsymbol{x}^{(i)-}(k-1) = \boldsymbol{x}^{(i)}(k-1) - \frac{1}{N} \sum_{j=1}^N \boldsymbol{x}^{(j)}(k-1) \\ \hat{\boldsymbol{V}}(k-1) = \frac{1}{N-1} \sum_{j=1}^N \boldsymbol{x}^{(j)-}(k-1)\boldsymbol{x}^{(j)-}(k-1)^T \\ \hat{\boldsymbol{R}}(k-1) = \frac{1}{N-1} \sum_{j=1}^N \boldsymbol{w}^{(j)}(k-1)\boldsymbol{w}^{(j)}(k-1)^T. \end{cases} \tag{17}$$

Next, the Kalman gain $\boldsymbol{G}(k)$ is obtained from the sample variance-covariance matrix $\hat{\boldsymbol{V}}(k-1)$, the sample variance-covariance matrix of the observed noise $\hat{\boldsymbol{R}}(k-1)$, and the matrix representing the observed state $\boldsymbol{H}$:

$$\boldsymbol{G}(k) = \hat{\boldsymbol{V}}(k-1)\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\hat{\boldsymbol{V}}(k-1)\boldsymbol{H}^{\mathrm{T}} + \hat{\boldsymbol{R}}(k-1))^{-1}. \tag{18}$$

Finally, the ensemble members are updated through the Kalman filter update rule:

$$\boldsymbol{x}^{(i)}(k) = \boldsymbol{x}^{(i)}(k-1) + \boldsymbol{G}(k)(\boldsymbol{y}(k) \\ + \boldsymbol{w}^{(i)}(k) - \boldsymbol{H}\boldsymbol{x}^{(i)}(k-1)). \tag{19}$$

4 System model

In this study, we used the maneuvering modeling group (MMG) model as the system model, representing the surge-sway-yaw maneuvering motion [39]. The model used in this study is for a vessel with a single propeller and twin rudders. Space-fixed coordinate systems $O - XY$ and ship fixed coordinate systems are used $o - xy$, as shown in Fig. 1. The origin of ship fixed coordinate systems o is on midship. Here, $O - XY$ represents a right-handed coordinate system, and Z-axis is the positive downward; $o - xy$ is a right-handed coordinate system, and z-axis is the positive downward. U denotes the ship's speed, and u, v_m denotes the speed in the surge and sway directions, respectively; ψ and r denote the angle and angular velocity in the yaw direction, respectively; β denotes the drift angle. U_T, γ_T denotes the wind speed and direction in the true direction, and U_A, γ_A denotes the relative wind speed and direction.
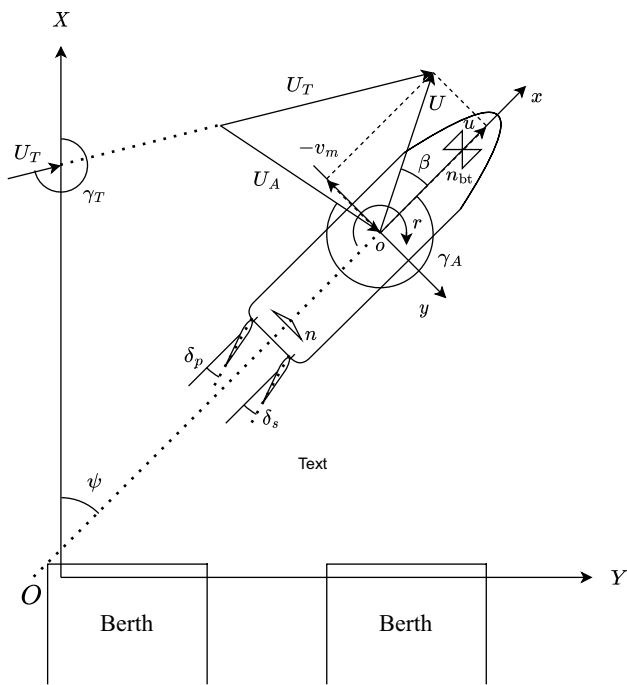
Fig. 1 Coordinate systems

The MMG model used is shown in Eq.(20). In the following, the dots represent differentiation with respect to time t:

$$(m + m_x)\dot{u} - (m + m_y)v_m r + mx_G r^2 = X$$
$$(m + m_y)\dot{v}_m + mx_G\dot{r} + (m + m_x)ur = Y \quad (20)$$
$$(I_{zz} + mx_G^2 + J_{zz})\dot{r} + mx_G(\dot{v}_m + ur) = N.$$

Here, x_G denotes the longitudinal coordinate of the center of the ship's gravity; m denotes the ship's mass and m_x, m_y denotes the added mass of the x-axis direction and y-axis direction; I_{zz}, J_{zz} denote the moment of inertia and added moment of inertia in the yaw direction, respectively; X, Y denote the hydrodynamic forces at the midship in x and y axis, respectively; N is the force moment around the z axis. In addition, the elements of the hydrodynamic forces to be considered are represented as follows:

$$X = X_H + X_P + X_R$$
$$Y = Y_H + Y_P + Y_R \quad (21)$$
$$N = N_H + N_P + N_R.$$

Here, the subscripts H, R, and P denote the hull, rudder, and propeller, respectively. For the hydrodynamic forces and moment acting on the hull, we employ the Yoshimura's low speed maneuvering model shown in Eq.(22) [40]:

$$X_H = \left(\frac{\rho}{2}\right)Ld$$
$$\left[\begin{array}{l} \left\{ X'_{0(F)} + \left(X'_{0(A)} - X'_{0(F)} \right)(\beta/\pi) \right\}uU \\ + X'_{vr}L \cdot v_m r \end{array} \right]$$

$$Y_H = \left(\frac{\rho}{2}\right)Ld$$
$$\left[\begin{array}{l} Y'_v v_m|u| + Y'_r L \cdot ru \\ -\left(\frac{C_D}{L}\right)\int_{-L/2}^{L/2} |v_m + C_{rY}rx|\left(v_m + C_{rY}rx\right)dx \end{array} \right] \quad (22)$$

$$N_H = \left(\frac{\rho}{2}\right)L^2 d$$
$$\left[\begin{array}{l} N'_v v_m u + N'_r L \cdot r|u| \\ -\left(\frac{C_D}{L^2}\right)\int_{-L/2}^{L/2} |v_m + C_{rY}rx|\left(v_m + C_{rY}rx\right)xdx \end{array} \right].$$

Here, ρ is the water density; L is the length between perpendicular of the ship; d is the draft; $X'_{0(A)}$, $X'_{0(F)}$ are the resistance coefficient in the forward and backward directions, respectively; C_D is the drag coefficient; C_{rY}, C_{rN} are the modification coefficients for the sway direction force and yaw moments, respectively; X'_{vr}, Y'_v, Y'_r, N'_v, N'_r, are non-dimensionalized hydrodynamic derivatives. The states x, control input vector u, and measurement vector y are as follows. The non-dimensionalized values are represented by superscript primes, as in $X'_{0(A)}$. The added mass term included in the original Yoshimura's low speed model is transferred to the left side of Eq.(20):

$$x \equiv (x, u, y, v, \psi, r)^T \in \mathbb{R}^6$$
$$u \equiv (n, \delta_s, \delta_p,)^T \in \mathbb{R}^3 \quad (23)$$
$$y \equiv (x, u, y, v, \psi, r)^T \in \mathbb{R}^6.$$

Here, n is the number of propeller revolutions; δ_s and δ_p are the starboard and port rudder angles, respectively.

The MMG model estimate the time derivative $\dot{x}$ by solving $g(x, u) : \mathbb{R}^6 \times \mathbb{R}^3 \to \mathbb{R}^6$ expressed in Eq.(20):

$$\dot{x} = g(x, u). \quad (24)$$

To use the Kalman filter in discrete time, we used the following discretized state transitions. Let $x(k)$ denote the state at k steps and $u(k)$ denote the control input at k steps:

$$x(k) + g(x(k), u(k))\Delta t = f(x(k), u(k)). \quad (25)$$

Here, $f(x(k), u(k))$ in Eq.(25) is applied to the system model in Eq.(1).

In this study, H is a unitary matrix (I) representing the observed model since the elements of the observed and estimated states coincide, as follows:

$$H = I \in \mathbb{R}^{6\times6}. \quad (26)$$

5 Investigation of factors affecting the accuracy of filtering

As mentioned in the introduction, the choice of the Kalman filter determines the accuracy of the estimation of state quantities. In this section, these factors are outlined and their importance is discussed. In this study, we examine the following three factors.

1. Modeling error on estimation accuracy
2. Observation noise
3. Type of maneuvering motion

For some parameters of the Kalman filter, first, the initial value of $x(0)$ was taken as the initial value of the observed value $y(0)$. Thus, it was defined as in Eq. (27):

$$x(0) = y(0). \tag{27}$$

The covariance matrix $\mathbf{Q}$ of the system noise, the covariance matrix $\mathbf{R}$ of the observed noise, and the initial value of the error covariance matrix $\mathbf{P}(0)$ are identical to align conditions other than items for verification. The parameter κ of UKF is also used with $\kappa = 3$. $\mathbf{Q} \in \mathbb{R}^{6\times6}, \mathbf{R} \in \mathbb{R}^{6\times6}, \mathbf{P}(0) \in \mathbb{R}^{6\times6}$ is a diagonal matrix. The covariance matrix of the system noise $\mathbf{Q}$ is as follows:

$$\mathbf{Q} = \mathbf{0}. \tag{28}$$

The covariance matrix of the observed noise $\mathbf{R}$ can be obtained from the system's performance specification of the onboard sensor of the system. Since this study was based on simulation, it was determined by referencing the measured data of onboard sensors. The value of $\mathbf{P}(0)$ is excessively small, the Kalman gain may become too small [41], and to avoid this, we adopt a unit matrix for $\mathbf{P}(0)$.

$$\mathbf{R} = \text{diag}(5.0 \times 10^{-2}, \ldots, 5.0 \times 10^{-2}). \tag{29}$$

$$\mathbf{P}(0) = \mathbf{I} \in \mathbb{R}^{6\times6}. \tag{30}$$

The number of ensemble members was set to $N = 100$ as the parameter for EnKF. Furthermore, to compare the filtering performance, we used the linear Kalman filter (KF) with the persistence prediction model to perform the filtering time. The detail of the algorithms can be found in the literature [9, 10]. In this study, KF was applied to the linear state-space model, as shown in Eq.(31):

$$x(k + 1) = Ax(k) + v(k)$$
$$y(k) = Cx(k) + w(k). \tag{31}$$

Here, A is $A = I \in \mathbb{R}^{6\times6}$ and C is $C = I \in \mathbb{R}^{6\times6}$. The parameter of KF was tuned separately by trial and error for better filtering performance.

5.1 Effect of modeling error on estimation accuracy

In this subsection, we investigate the effect of modeling errors on filtering using the Kalman filter. It is impossible to construct a motion model that estimates motion completely without error. Hence, modeling errors always exist. In this study, we did not consider wind in the system model but as a modeling error.

First, two models are prepared for comparison. The first is the MMG model using Eq. (21). The second is the MMG model mentioned above with the effect of wind, as shown below. The system model used for estimation does not incorporate a wind model. Instead, the impact of wind is treated as a modeling error, and the wind is only taken into account in the observations for filtering. The equation of state used in this study is as follows:

$$X = X_{\text{H}} + X_{\text{P}} + X_{\text{R}} + \underline{X_{\text{A}}}$$
$$Y = Y_{\text{H}} + Y_{\text{P}} + Y_{\text{R}} + \underline{Y_{\text{A}}} \tag{32}$$
$$N = N_{\text{H}} + N_{\text{P}} + N_{\text{R}} + \underline{N_{\text{A}}}.$$

Here, the subscript A represents the effect of wind. The force due to wind was obtained using the Fujiwara's equation [42] shown in Eq.(34):

$$\begin{cases} X_{\text{A}} = (1/2)\rho_{\text{A}} U_{\text{A}}^2 A_{\text{T}} \cdot C_X \\ Y_{\text{A}} = (1/2)\rho_{\text{A}} U_{\text{A}}^2 A_{\text{L}} \cdot C_Y \\ N_{\text{A}} = (1/2)\rho_{\text{A}} U_{\text{A}}^2 A_{\text{L}} L_{\text{OA}} \cdot C_N, \end{cases} \tag{33}$$

where

$$\begin{cases} C_X = X_0 + X_1 \cos\left(2\pi - \gamma_{\text{A}}\right) + X_3 \cos 3\left(2\pi - \gamma_{\text{A}}\right) \\ \quad + X_5 \cos 5\left(2\pi - \gamma_{\text{A}}\right) \\ C_Y = Y_1 \sin\left(2\pi - \gamma_{\text{A}}\right) + Y_3 \sin 3\left(2\pi - \gamma_{\text{A}}\right) \\ \quad + Y_5 \sin 5\left(2\pi - \gamma_{\text{A}}\right) \\ C_N = N_1 \sin\left(2\pi - \gamma_{\text{A}}\right) + N_2 \sin 2\left(2\pi - \gamma_{\text{A}}\right) \\ \quad + N_3 \sin 3\left(2\pi - \gamma_{\text{A}}\right). \end{cases} \tag{34}$$

Here, ρ_{A} is the density of the air; A_T, A_L are the transverse projected and lateral projected areas, respectively; L_{OA} is the overall length of the ship; X_i, Y_i, N_i are the wind pressure coefficients derived using the regression equation [42]. This equation is based on wind tunnel experimental data of many scaled models with the geometric parameters of the ships as explanatory variables. The model that takes wind into account is referred to as the w/ 'wind model. The model that does not take wind into account is referred to as the without-wind model. An investigation of the effect of modeling errors is conducted as follows:

Table 1 Average wind speed and direction

Item	Value
Average of wind speed: $\bar{U}_T$	1.64 [m/s]
Average of wind direction: $\bar{\gamma}_T$	2.63 [rad]
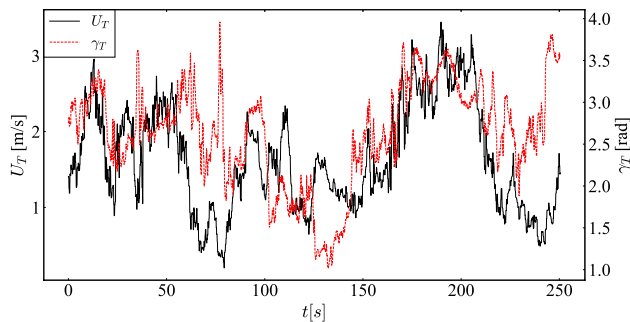
Fig. 2 Example of time series of wind velocity and direction

– First, time-series data are generated using the w/ wind model for three different wind speeds. White noise, according to $w \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$, is applied to the time-series data as the observation noise. These generated data were used for filtering. Δt of the data is 0.1[s].

– In order to take the effect of wind as the modeling error, we apply the without-wind model to the system models of UKF and EnKF. The data are filtered for the three different wind speeds generated above.

– We use UKF and EnKF to calculate the root mean square error (RMSE) between the result of each filtered state and the true value. The definition of RMSE is given in Eq. (35):

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(y_i - x_i\right)^2}. \tag{35}$$

Here, N is the number of data; y_i is the filtered value; x_i is the true value. The true values are the time-series data before white noise is added.

Wind direction and speed were observed from the data on June 10, 2021, at Inukai Pond on the Osaka University campus. Here, the wind is assumed to be a variable in time but constant in space. Figure 2 shows the time series of wind direction and speed. Table 1 presents the average wind speed $\bar{U}_T$ and the average wind direction $\bar{\gamma}_T$.

For the time-series data of wind direction and speed applied to the simulation, only the wind speed was multiplied by 1, 0.5, and 0 to produce three different data with average wind speeds of $\bar{U}_T = 1.64$ [m/s], $\bar{U}_T = 0.82$ [m/s], and $\bar{U}_T = 0.0$ [m/s], respectively. When the wind speed is

$\bar{U}_T = 0.0$ [m/s], the with-wind and without models completely coincide, and the modeling error increases as the average wind speed increases. First, the modeling errors of the with-wind and without models were compared by the trajectory of the turning test. The results shown in Fig. 3 are consistent with the with-wind and without models when the average wind speed is $\bar{U}_T = 1.64$ [m/s].

Figure 3 shows that the with-wind model trajectory drifts more downwind than the without-wind model results.

Next, we present the filtering results. Filtering is performed using UKF, EnKF, and KF under each condition shown in Table 2. Figure 4 shows the comparison of RMSE in each state.

As shown in Fig. 4, the estimation accuracy became worse as the modeling error increased for UKF and EnKF. In particular, the estimation accuracy rapidly worsened for EnKF due to modeling error. These results suggest that EnKF is relatively more easily affected by the modeling error of the system model than UKF. The covariance matrix of the system noise was set to $\mathbf{Q} = \mathbf{0}$ in this validation. Therefore, the effect of setting of $\mathbf{Q}$ significantly affected the estimation accuracy, especially in EnKF that applied the ensemble member of the system noise to the state as a disturbance. Next, we show the filtering results for UKF and EnKF when the system noise covariance matrix $\mathbf{Q}$ is set to $\mathbf{Q} \neq \mathbf{0}$. The covariance matrix $\mathbf{Q}_{\text{UKF}} \in \mathbb{R}^{6\times6}$ of the system noise in UKF was obtained by trial and error for better filtering performance as follows:

$$\mathbf{Q}_{\text{UKF}} = \text{diag}(1.0 \times 10^{-4}, \dots, 1.0 \times 10^{-4}). \tag{36}$$

The covariance matrix $\mathbf{Q}_{\text{EnKF}} \in \mathbb{R}^{6\times6}$ of the system noise in EnKF is as follows:

$$\mathbf{Q}_{\text{EnKF}} = \text{diag}(1.0 \times 10^{-2}, \dots, 1.0 \times 10^{-2}). \tag{37}$$

Here, the filtering was performed using UKF and EnKF under the conditions shown in Table 3. The filtering results for each obtained state and RMSE were compared for each filter. Figure 5 shows the obtained RMSE for each state. In addition, the estimated time-series data were those at $\bar{U}_T = 1.64$[m/s] with the largest modeling error.

Figure 5 shows that the estimation improved when the system noise was considered. In particular, some states in EnKF significantly improve estimation accuracy. In other words, tuning the ensemble members of the system noise can improve the estimation accuracy. For UKF, tuning the system noise covariance matrix also improved the estimation accuracy. However, the difference was not as remarkable as in the case of EnKF. Therefore, when using UKF for state estimation, it is more important than for EnKF to reduce the modeling error of the system model as much as possible.
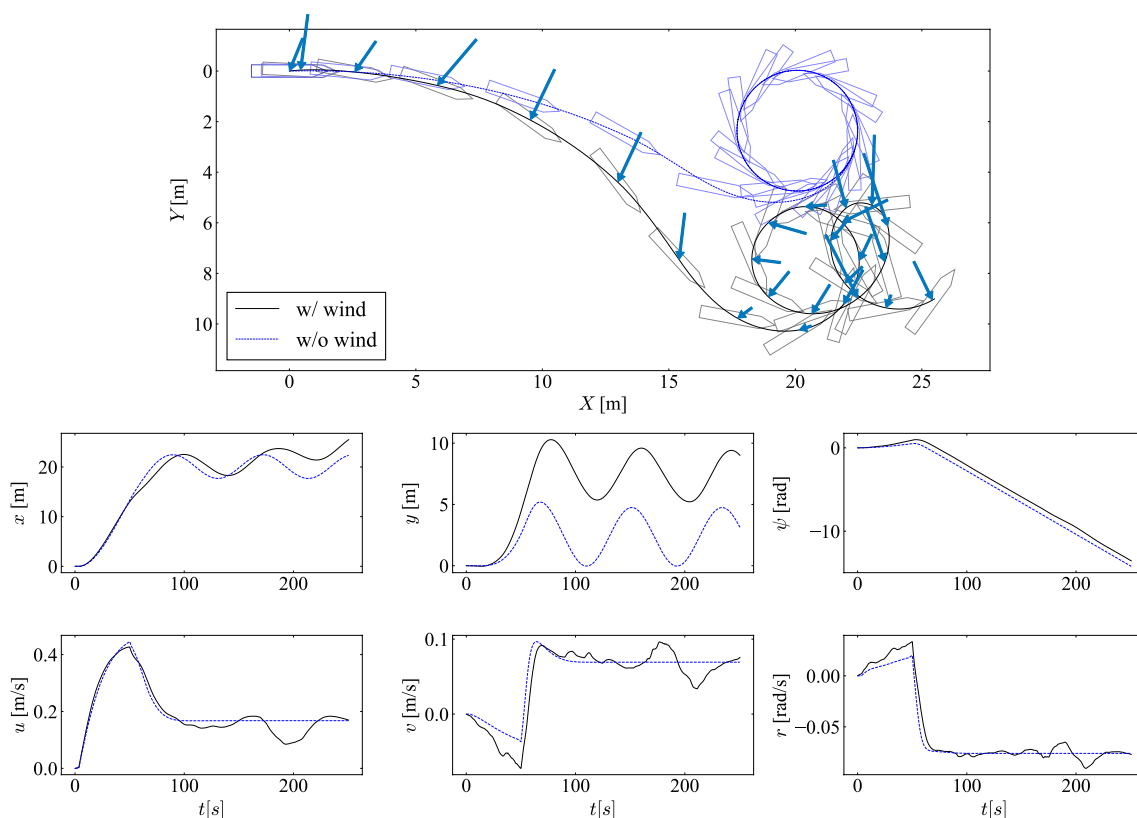
Fig. 3 Comparison of the w/ wind model and w/o wind model trajectory and time series

Table 2 System model and model to be estimated (different wind speeds are applied)

System model	Target of estimation
w/o wind model	w/ wind model ($\bar{U}_T = 0.0$[m/s], $\bar{\gamma}_T = 2.63$ [rad])
w/o wind model	w/ wind model ($\bar{U}_T = 0.82$[m/s], $\bar{\gamma}_T = 2.63$ [rad])
w/o wind model	w/ wind model ($\bar{U}_T = 1.64$[m/s], $\bar{\gamma}_T = 2.63$ [rad])

5.2 Effect of observation noise on estimation accuracy

Observed values by sensors always include observation noise. When using the Kalman filter, it is necessary to input parameters, such as the covariance matrix of the noise for the observed noise $\boldsymbol{w}$ and the system noise $\boldsymbol{v}$. In addition, several studies have been conducted on how to determine these parameters when implementing the Kalman filter [41]. However, they are often determined by trial and error. In this section, we investigate the effect of the magnitude of the observation noise on the filtering accuracy. To achieve this, we implemented the following method.

– The time-series data are generated using the same procedure as in Sect. 5.1. Here, the time-series

data were generated using the without-wind model. To verify the effect of the observed noise, we prepared data with three types of noise applied to the covariance matrix of the observed noise: $\mathbf{R} = \text{diag}(0.05^2, \ldots, 0.05^2)$, $\mathbf{R} = \text{diag}(0.10^2, \ldots, 0.10^2)$, and $\mathbf{R} = \text{diag}(0.20^2, \ldots, 0.20^2)$.

– Filtering is performed on the three types of data using UKF and EnKF. At this time, in order to align the conditions, the covariance matrix of the observed noise used in UKF and the ensemble members of the observed noise used in EnKF are the same regardless of the data type. The system model applied to each filter is the without-wind model.

– Using UKF and EnKF, we calculate the RMSE between the result of each filtered state and the true value.

Using the above method, we performed the filtering using UKF, EnKF, and KF under the conditions shown in Table 4. The filtering results for each obtained state and RMSE were compared for each filter. Figure 6 shows the obtained RMSE for each state.

As shown in Fig. 6, the RMSE obtained using UKF and EnKF showed that the observation accuracy tends to be worse as the variance of the observation noise increases.
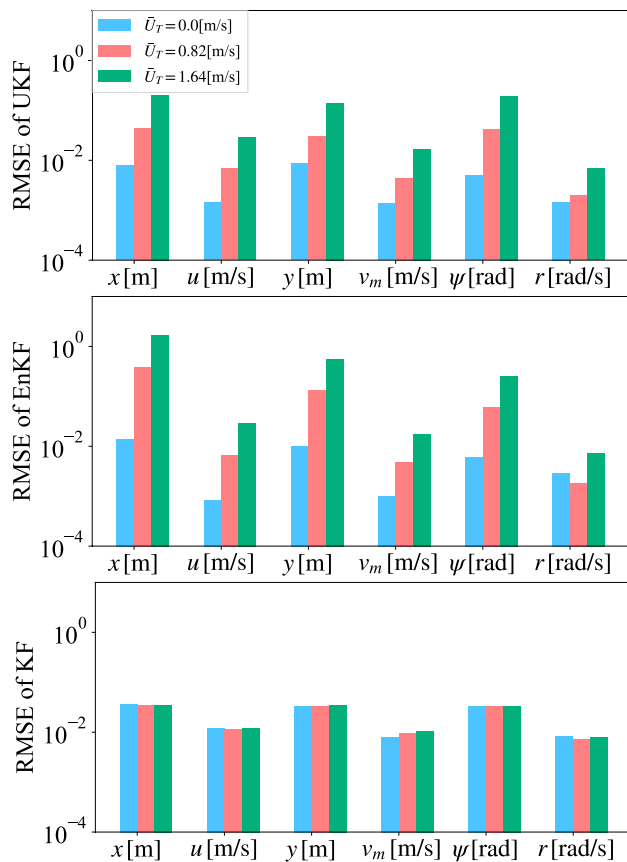
Fig. 4 Comparison of RMSE in wind condition (upper tile: `UKF` result, middle tile: `EnKF` result, and lower tile: `KF` result)

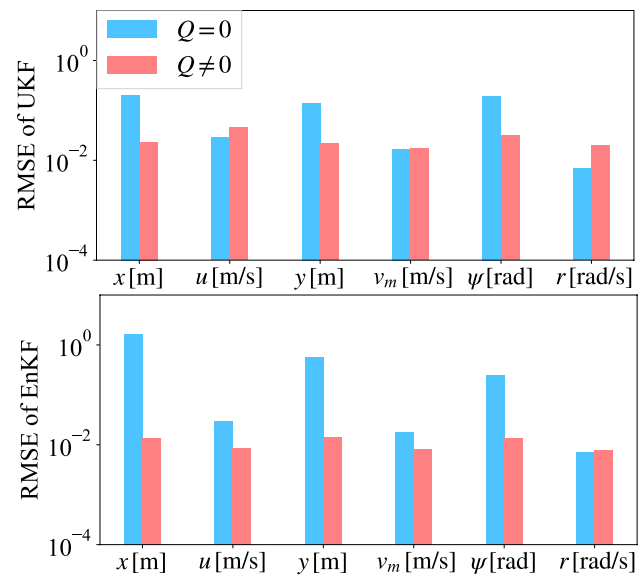**Fig. 5** Comparison of RMSE in wind condition ($\mathbf{Q} = 0$ and $\mathbf{Q} \neq 0$, upper tile: `UKF` result, lower tile: `EnKF` result)

Table 4 System model and model to be estimated (different $\mathbf{R}$ are applied)

System model	Target of estimation
w/o wind model	w/o wind model ($\mathbf{R} = \mathrm{diag}(0.050^2, \dots, 0.050^2)$)
w/o wind model	w/o wind model ($\mathbf{R} = \mathrm{diag}(0.10^2, \dots, 0.10^2)$)
w/o wind model	w/o wind model ($\mathbf{R} = \mathrm{diag}(0.20^2, \dots, 0.20^2)$)

Table 3 System model and model to be estimated (different $\mathbf{Q}$ are applied to system model)

System model	Target of estimation
w/o wind model ($\mathbf{Q} = \mathbf{0}$)	w/ wind model ($\bar{U}_T = 1.64[\mathrm{m/s}]$, $\bar{\gamma}_T = 2.63$ [rad])
w/o wind model ($\mathbf{Q} \neq \mathbf{0}$)	w/ wind model ($\bar{U}_T = 1.64[\mathrm{m/s}]$, $\bar{\gamma}_T = 2.63$ [rad])

However, the observation noise did not affect the estimation accuracy significantly as the modeling error. Therefore, it is essential to consider the effects of observation noise for accurate state estimation but not the extent of modeling errors in the system model.

5.3 Effect of type of maneuvers on estimation accuracy

In this section, we investigate the effect of the type of maneuvers on the estimation accuracy of `UKF` and `EnKF`.

Thus, we generate the following two types of maneuvering motions as time-series data.

1. Zig-zag motion
2. Random motion

As mentioned in the introduction, Konatowski compared the estimation accuracy of `EKF` and `UKF` for constant velocity and complex motion systems [29]. For the type of maneuvering motion, as long as the Zig-zag maneuvers are given input with periodic motion, the final result is a periodic solution as long as it does not diverge. The motion is broadly considered as stationary, periodic, and aperiodic solutions. For simplicity (Fig. 7), no verification for step response will be performed at this time. We examine the Zig–zag and random maneuvers to compare periodic and aperiodic motions. Here, the propeller revolution was set to be constant. The rudder angle was given as Zig-zag maneuvers and random maneuvers. Here, the random maneuvers were created by giving the rudder angle as a uniform random number (δ_s is $[-35, 105]$ δ_p is $[-105, 35]$). Figure 8 shows an example of rudder angle time series and the trajectory.
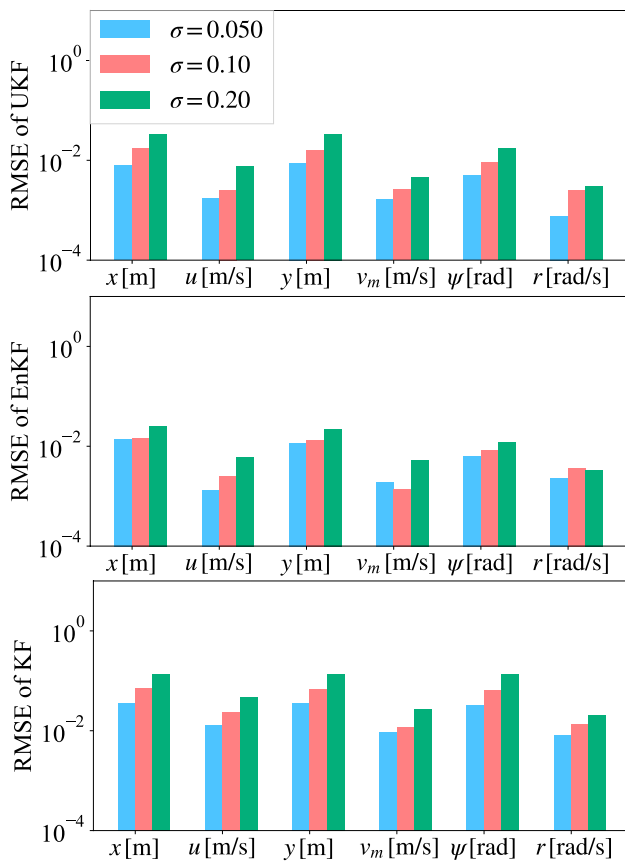
Fig. 6 Comparison of RMSE under different observation noise conditions (upper tile: `UKF` result, middle tile: `EnKF` result, and lower tile: `KF` result)
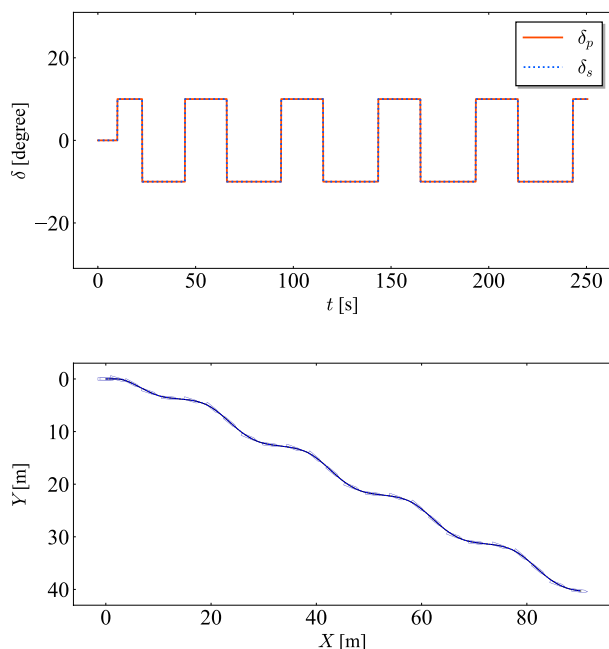

Fig. 7 Time series of rudder angle and trajectory(upper panel: rudder angle of Zig-zag motion, lower panel: trajectory of Zig-zag motion)
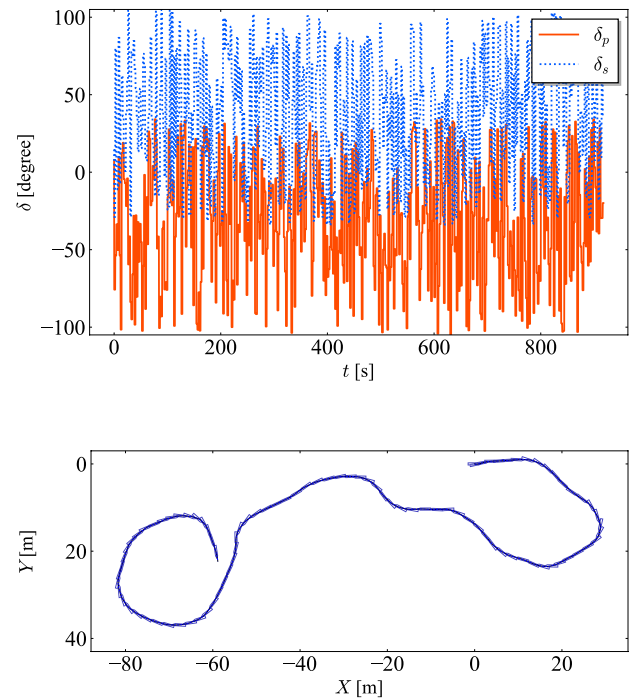

Fig. 8 Time series of rudder angle and trajectory(upper panel: rudder angle of random motion, lower panel: trajectory of random motion)

We used the following procedures to examine the effect of the type of maneuvers.

– The same procedure as in Sect. 5.2 is used to create the time-series data. At this time, the two types of inputs shown in Fig. 8 are used. The system model is the without-wind model described above. The filtering data are created by applying white noise of $w \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ to the time-series data.
– Filtering is performed on the two types of data using `UKF` and `EnKF`. In order to align the conditions, the covariance matrix $\mathbf{R}$ of the observed noise used in `UKF` and the variance of the ensemble members of the noise used in `EnKF` were the same, regardless of the data type.
– `UKF` and `EnKF` are used to calculate the RMSE between the result of each filtered state and the true value.

Figure 9 compares the obtained filtering results of each state and RMSE for each filter.

As shown in Fig. 9, there was no significant difference in the estimation accuracy of each state between different types of maneuvers in the estimation using `EnKF` and `UKF`. Thus, it can be considered that the difference in the type of maneuvers has little effect on the estimation accuracy. From the dynamical perspective, the periodic or aperiodic motion did not have a significant difference in estimation accuracy.
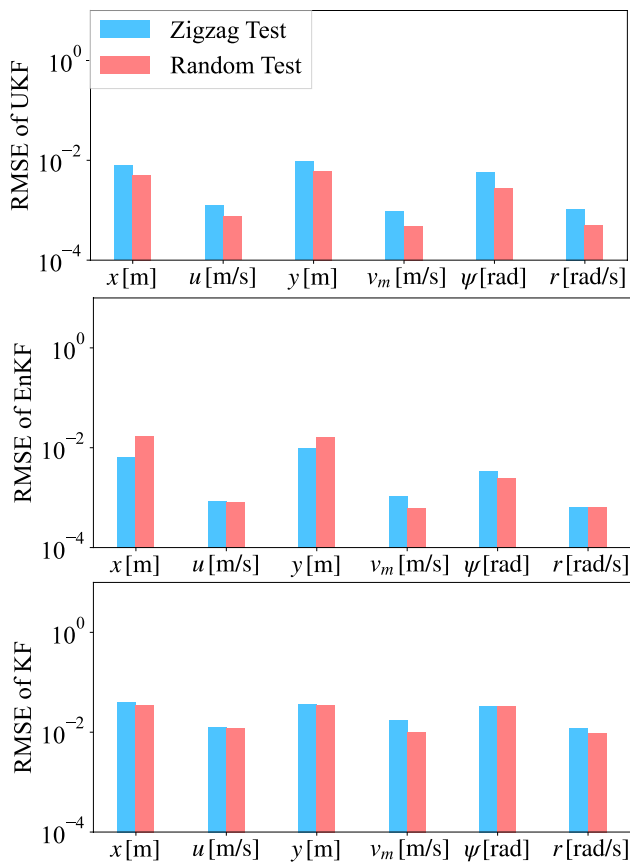
Fig. 9 Comparison of RMSE in different maneuvering motion conditions (upper tile: `UKF` result, middle tile: `EnKF` result, and lower tile: `KF` result)

5.4 Comparison of computation time

Real-time state estimation is important for real-time control. Therefore, we verify the real-time performance of the filtering algorithm by comparing the computational time of `UKF` and `EnKF`. The sample data to be filtered are numerically simulated Zig-zag maneuvers explained in Sect. 5.3 was performed. Ten datasets with different applied noises were prepared, and the average computational time was obtained for each filter. The computational time per step (0.1[s]) is obtained to verify the real-time performance of the filtering algorithm and compare the computational time for each filter. Table 5 presents the computer specification used in this study.

Table 5 Computer specifications

CPU	Apple M1
Clock	3.2 GHz
Size of memory	8 G byte
Programming language	Python

The comparison of computational time is shown in Fig. 10.

As shown in Fig. 10, all filters took less than 0.1[s] per step to filter. As a result, all filters had real-time performance. The computational time for `EnKF` depends on the number of ensemble members, which was about four times longer than that of `UKF`.

5.5 Guideline for filtering

From the results of Sects. 5.1 to 5.4, we obtained the following guideline for filter selection.

– Modeling error:
  Modeling error had a significant effect on estimation accuracy for `UKF` and `EnKF`. In `UKF` and `EnKF`, it was possible to improve the estimation accuracy by tuning the covariance matrix and the ensemble members of the system noise given to `EnKF`. In particular, `EnKF` showed significant improvement. It can also happen that `KF` is better than `UKF` and `EnKF` if the modeling error is high. However, the accuracies of `UKF` and `EnKF` were better than that of the `KF` in all other cases.

– Covariance matrix:
  It is essential to choose the appropriate covariance matrix for the observed noise because it affects the accuracy of the `UKF` and `EnKF` estimates. However, the importance was not as large as the modeling error. Therefore, even if the covariance matrix of the observed noise is not chosen carefully, it is possible to obtain an acceptable estimation.

– Periodic or aperiodic motion:
  We could not confirm that filtering performance was affected by periodic or aperiodic motion. Thus, there is no need to select between `UKF` and `EnKF` for different maneuverings.
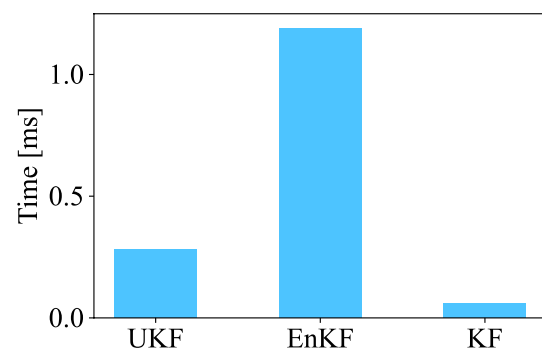
– Computation time:

Fig. 10 Comparison of computational time between EnKF, UKF, and KF

For both filters, the computational time per step was less than 0.1 [s], the observation period. Therefore, there was a real-time performance in this calculation. `EnKF` took about four times longer to compute than `UKF`. Thus, it is appropriate to use `UKF` when real-time performance is required under severe conditions.

Since `UKF` and `EnKF` do not have extreme differences in estimation accuracy, it is essential to select the appropriate filter for the computational time and purpose.

5.6 Discussion

To further improve the estimation accuracy in the future, it is important to keep the modeling error of the system model sufficiently small when using `UKF` and `EnKF`. Several attempts have been recently made to improve the performance of maneuvering motion models in ship maneuvering fields. For instance, Araki et al. estimated the maneuvering model by utilizing several experimental fluid dynamic systems and computational fluid dynamics free-running trials. They employed the `EKF` and the constrained least square method with the generalized reduced gradient algorithm [43]. Jian et al. identified the Abkowitz model from the simulation dataset using least square support vector machines [44]. Serge and Guedes developed an identification algorithm for ship maneuvering mathematical models using the classic genetic algorithm [45]. Wakita et al. proposed a system identification method for generating low speed maneuvering models using neural networks [46]. Miyauchi et al. estimated the parameters of the MMG model from the trajectories of free-running tests using an optimization algorithm, the covariance matrix adaptation evolution strategy (CMA-ES). The results show that the model is more consistent with the results of free-running tests than with the parameters obtained from captive model tests [47]. Based on these studies, it is possible that a motion model with improved performance could be applied to the nonlinear Kalman filter to provide a more accurate state estimation. Furthermore, from a robustness perspective, we consider it important to develop algorithms that account for the sudden deterioration of observation noise due to instrument failure. To deal with this problem, for example, the development of sensor fusion approaches would be useful. Currently, there is a lot of research on sensor fusion, and a sensor fusion technique using UKF has been proposed. For example, a study on sensor fusion technology using UKF for aircraft was conducted by Majeed and Kar [48]. They are also developing a sensor fusion algorithm using a nonlinear Kalman filter for ships.

6 Conclusion

In this study, we investigated the performance of `UKF` and `EnKF` filters to provide guidance. We examined the effect of the type of maneuvers, observation noise, and modeling error on the estimation accuracy of `UKF` and `EnKF`. The results showed that the modeling error of the system model has a significant effect on the estimation accuracy of `EnKF` and `UKF`. There was no significant difference in estimation accuracy between the different types of observation noise and the type of maneuvers. Thus, since berthing motion is considered to be similar to random motion, both filters are considered to be applicable during berthing. In addition, the time required for filtering was measured to verify the online capability of the system, and real-time filtering was possible for `EnKF` and `UKF`. However, in this study, `EnKF` took approximately four times longer to compute than `UKF`. Therefore, we obtained that `EnKF` and `UKF` should be used separately depending on the required computational time and purpose.

References

1. Ministry of Land (2021) Infrastructure, transport and tourism. White paper on land, infrastructure, transport and tourism in Japan
2. Takai T (1990) Automatic berthing experiments using" shioji-maru" (in japanese). J Jpn Inst Navig 83:267–276
3. Maki A, Sakamoto N, Akimoto Y, Nishikawa H, Umeda N (2020) Application of optimal control theory based on the evolution strategy (cma-es) to automatic berthing. J Mar Sci Technol 25(1):221–233
4. Rachman DM, Maki A, Miyauchi Y, Umeda N (2022) Warm-started semionline trajectory planner for ship's automatic docking (berthing). Ocean Eng 252:111127
5. Miyauchi Y, Sawada R, Akimoto Y, Umeda N, Maki A (2022) Optimization on planning of trajectory and control of

autonomous berthing and unberthing for the realistic port geometry. Ocean Eng 245:110390

6. Jahren C, Jones R (1993) Ferry landing design phase i. final report. Tech Rep

7. Roubos A, Groenewegen L, Peters DJ (2017) Berthing velocity of large seagoing vessels in the port of rotterdam. Mar Struct 51:202–219

8. Association IMWP https://www.isemikawapilot.jp/wp-content/uploads/2021/12/keiryu.pdf

9. Kalman RE (1960) On the general theory of control systems. in Proceedings First International Conference on Automatic Control, Moscow, USSR, pp. 481–492

10. Kalman RE, Bucy RS (1961) New results in linear filtering and prediction theory

11. Humpherys J, Redd P, West J (2012) A fresh look at the kalman filter. SIAM Rev 54(4):801–823

12. Yang C, Shi W, Chen W (2017) Comparison of unscented and extended kalman filters with application in vehicle navigation. J Navig 70(2):411–431

13. Gustafsson F, Gunnarsson F, Bergman N, Forssell U, Jansson J, Karlsson R, Nordlund P-J (2002) Particle filters for positioning, navigation, and tracking. IEEE Trans Signal Process 50(2):425–437

14. Giannitrapani A, Ceccarelli N, Scortecci F, Garulli A (2011) Comparison of ekf and ukf for spacecraft localization via angle measurements. IEEE Trans Aerosp Electron Syst 47(1):75–84

15. Farina A, Ristic B, Benvenuti D (2002) Tracking a ballistic target: comparison of several nonlinear filters. IEEE Trans Aerosp Electron Syst 38(3):854–867

16. Ceccarelli N, Garulli A, Giannitrapani A, Leomanni M, Scortecci F (2007) Spacecraft localization via angle measurements for autonomous navigation in deep space missions. IFAC Proc 40(7):551–556

17. Larsen TD, Hansen KL, Andersen NA, and Ravn O (1999) Design of kalman filters for mobile robots; evaluation of the kinematic and odometric approach. In Proceedings of the 1999 IEEE international conference on control applications (Cat. No. 99CH36328), vol. 2, pp. 1021–1026, IEEE

18. Battin RH (2002) Some funny things happened on the way to the moon. J Guid Control Dyn 25(1):1–7

19. Balchen JG, Jenssen NA, Mathisen E, Sælid S (1980) A dynamic positioning system based on kalman filtering and optimal control. Model Identif Control Norweg Res Bull 1(3):135–163

20. Sørensen AJ, Sagatun SI, Fossen TI (1996) Design of a dynamic positioning system using model-based control. Control Eng Pract 4(3):359–368

21. Abkowitz MA (1980) Measurement of hydrodynamic characteristics from ship maneuvering trials by system identification. Tech Rep

22. Fossen TI, Sagatun SI, Sørensen AJ (1996) Identification of dynamically positioned ships. Control Eng Pract 4(3):369–376

23. Ochiai Y, Okazaki T, Ohtsu K (2006) Study on precise prediction of ship's position using ekf (in japanese). J Japan Inst Navig 115:1–10

24. Miyoshi S, Hara Y, Ohtsu K (2008) A study on optimum tracking control with kalman filter for vessel (in japanese). J Japan Inst Navig 118:47–53

25. Fossen S, and Fossen TI (2018) Extended kalman filter design and motion prediction of ships using live automatic identification system (ais) data. In 2018 2nd European Conference on Electrical Engineering and Computer Science (EECS), pp. 464–470, IEEE

26. Fossen TI (2022) Line-of-sight path-following control utilizing an extended kalman filter for estimation of speed and course over ground from gnss positions. J Mar Sci Technol 27(1):806–813

27. Subchan S, Ismail RW, Asfihani T, and Adzkiya D (2019) Estimation of hydrodynamic coefficients using unscented kalman filter and recursive least square. 2019 IEEE 11th International Workshop on Computational Intelligence and Applications, IWCIA 2019 - Proceedings, pp. 9–13

28. Deng F, Yang HL, Wang LJ, Yang WM (2019) UKF based nonlinear offset-free model predictive control for ship dynamic positioning under stochastic disturbances. Int J Control Autom Syst 17:3079–3090

29. Konatowski S, Pieniżny AT (2007) A comparison of estimation accuracy by the use of KF. EKF & UKF filters, Computational Methods and Experimental Measurements XIII

30. Konatowski S, Kaniewski P, and Matuszewski J (2016) Comparison of estimation accuracy of ekf, ukf and pf filters. Ann Navig

31. Chen H, Chen G, Blasch E, and Pham K (2009) Comparison of several space target tracking filters. In Sensors and Systems for Space Applications III, vol. 7330, pp. 118–129, SPIE

32. Nishimura T (1970) Modeling errors in kalman filters. Tech Rep

33. Julier SJ, and Uhlmann JK (1997) A new extension of the kalman filter to nonlinear systems. In Signal processing, sensor fusion, and target recognition VI, vol. 3068, pp. 182–193, International Society for Optics and Photonics

34. Julier S, Uhlmann J, Durrant-Whyte HF (2000) A new method for the nonlinear transformation of means and covariances in filters and estimators. IEEE Trans Autom Control 45(3):477–482

35. Julier SJ, Uhlmann JK (2004) Unscented filtering and nonlinear estimation. Proc IEEE 92(3):401–422

36. Evensen G (1994) Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. J Geophys Res 99(C5):10143

37. Evensen G (2003) The ensemble kalman filter: Theoretical formulation and practical implementation. Ocean Dyn 53(4):343–367

38. Katzfuss M, Stroud JR, Wikle CK (2016) Understanding the ensemble kalman filter. Am Stat 70(4):350–357

39. Yasukawa H, Yoshimura Y (2015) Introduction of mmg standard method for ship maneuvering predictions. J Mar Sci Technol 20(1):37–52

40. Yoshimura Y, Nakao I, and Ishibashi A (2009) Unified mathematical model for ocean and harbour manoeuvring. International Conference on Marine Simulation and Ship Maneuverability

41. Schneider R, Georgakis C (2013) How to not make the extended kalman filter fail. Ind Eng Chem Res 52(9):3354–3362

42. Fujiwara T, Ueno M, Nimura T (1998) Estimation of wind forces and moments acting on ships. J Soc Naval Arch Japan 1998(183):77–90

43. Araki M, Sadat-Hosseini H, Sanada Y, Tanimoto K, Umeda N, Stern F (2012) Estimating maneuvering coefficients using system identification methods with experimental, system-based, and cfd free-running trial data. Ocean Eng 51:63–84

44. Jian C, Jiayuan Z, Feng X, Jianchuan Y, Zaojian Z, Hao Y, Tao X, Luchun Y (2015) Parametric estimation of ship maneuvering motion with integral sample structure for identification. Appl Ocean Res 52:212–221

45. Sutulo S, Soares CG (2014) An algorithm for offline identification of ship manoeuvring mathematical models from free-running tests. Ocean Eng 79:10–25

46. Wakita K, Maki A, Umeda N, Miyauchi Y, Shimoji T, Rachman DM, Akimoto Y (2022) On neural network identification for low-speed ship maneuvering model. J Mar Sci Technol 27(1):772–785

47. Miyauchi Y, Maki A, Umeda N, Rachman DM, and Akimoto Y (2021) System parameter exploration of ship maneuvering model for automatic docking/berthing using cma-es. arXiv preprint arXiv:2111.06124

48. Majeed M, Kar IN (2011) Multi sensor data fusion based approach for the calibration of airdata systems. Aeronaut J 115(1164):113–122