Article

A Robust and Adaptive AUV Integrated Navigation Algorithm Based on a Maximum Correntropy Criterion

Pinchi Li [1], Xiaona Sun [2], Ziyun Chen [2], Xiaolin Zhang [1], Tianhong Yan [1,*] and Bo He [2,*]

1 College of Mechanical and Electrical Engineering, China Jiliang University, Hangzhou 310018, China; p22010854068@cjlu.edu.cn (P.L.); s21010811038@cjlu.edu.cn (X.Z.)
2 Department of Information Science and Engineering, Ocean University of China, Qingdao 266100, China; sunxiaona@stu.ouc.edu.cn (X.S.); chenziyun@stu.ouc.edu.cn (Z.C.)
* Correspondence: thyan@cjlu.edu.cn (T.Y.); bhe@ouc.edu.cn (B.H.)

Abstract: In the underwater domain where Autonomous Underwater Vehicles (AUVs) operate, measurements may suffer from the impact of outliers and non-Gaussian noise. These factors can potentially undermine the efficacy of integrated navigation algorithms. The Maximum Correntropy Criterion (MCC) can be utilized to enhance the robustness of AUV integrated navigation algorithms through the construction and maximization of the correntropy function. Notwithstanding, the underwater environment occasionally presents unknown time-varying noise, a situation for which the MCC lacks adaptability. In response to this issue, our study introduces a novel integrated navigation algorithm that synergizes the MCC and the Variational Bayesian approach, thereby augmenting both the robustness and adaptability of the system. Initially, we implement the MCC along with a mixture kernel function in an Unscented Kalman Filter (UKF) to strengthen the robustness of the AUV integrated navigation algorithms amidst the complexities inherent to underwater environmental conditions. Additionally, we utilize the Variational Bayesian method to refine the approximation of measurement noise covariance, thereby boosting the algorithm's adaptability to fluctuating scenarios. We evaluate the performance of our proposed algorithm using both simulation and sea trial datasets. The experimental results reveal a significant enhancement in the Root Mean Square Error (RMSE) and navigation accuracy of our proposed algorithm. Notably, in a complex noise environment, our algorithm achieves, approximately, a 50% improvement in navigation accuracy over other established algorithms.

Keywords: autonomous underwater vehicle (AUV); integrated navigation; maximum correntropy criterion; mixture correntropy; variational bayesian; robustness; adaptability

1. Introduction

To effectively execute missions beneath the sea, Autonomous Underwater Vehicles (AUVs) have been developed [1–3]. AUVs are instrumental in advancing marine scientific research, marine engineering, and underwater archaeology, thereby becoming essential tools for oceanic exploration and study [4]. Accurate navigation and positioning are paramount for AUVs to enhance their performance in executing a diverse array of underwater tasks [2,5].

The AUV integrated navigation algorithm, which integrates various navigation sensors such as an Inertial Navigation System (INS), Doppler Velocity Log (DVL), and pressure sensor through a filtering algorithm, demands fewer resources while facilitating high-precision autonomous underwater navigation. For a nonlinear integrated navigation model of an AUV, the Unscented Kalman Filter (UKF) can be employed [6,7]. The UKF leverages the Unscented Transform (UT) to generate sigma points, providing an accurate approximation of the nonlinear model. Despite its advantages, the efficiency of this traditional integrated navigation algorithm is further compromised in complex noise environments, like unknown underwater environments, that are characterized by non-Gaussian noise, time-varying noise, and outliers.

To enhance the accuracy of estimations and bolster the robustness and adaptability of the filtering algorithm, specific mathematical methods are employed. The M-estimator, which is based on the least squares (LS) criterion, has been successfully introduced to improve the robustness of the KF. The M-estimator can suppress the influence of non-Gaussian noise and outliers by minimizing the cost function based on the measurements residual [8]. The Huber Kalman Filter (HKF) utilizes the Huber function to construct a cost function in the M-estimator. As an improved robust estimation algorithm, the HKF has been widely used and verified to have superior performance. In practical navigation, Crespillo et al. came up with a tightly coupled GNSS/INS integrated algorithm based on the HKF, which was verified to have a better performance in challenging GNSS scenarios [9]. Wang et al. achieved hierarchical water velocity estimation in a middle-water environment using an HKF, enhancing the accuracy of SINS/DVL integrated navigation [10]. In addition, the robust Student's t Kalman Filter (STKF) was proposed, which achieves a robust estimate by approximating the process noise and measurement noise as a heavy-tailed non-Gaussian distribution. Jia et al. designed a SINS/GPS integrated navigation method based on the STKF. Simulations were conducted to prove its superior robustness and performance in complex noise environments [11]. In the field of underwater navigation, Wang et.al came up with an SINS/USBL integrated algorithm based on the STKF. Simulations and field trails were conducted to verify its robustness [12].

Rather than using the least squares criterion and typical filtering algorithms, the Maximum Correntropy Criterion (MCC) was introduced. The MCC focuses on constructing and maximizing a cost function predicated on correntropy, which in turn augments the robustness of the algorithm. Such advancements have provided fresh insights into addressing the challenges associated with solving the correntropy function. Chen et al. introduced an innovative fixed-point method-based iterated maximum correntropy UKF (IMCUKF) [13] and investigated the algorithm's convergence properties [14]. Wang et al. proposed a non-iterative MCUKF, employing a cost function derived from the weighted least squares (WLS) method [15]. Additionally, two iterative versions of the MCUKF based on the Gauss–Newton and Levenberg–Marquardt methods were put forward [16]. These were demonstrated to offer a superior estimation performance and numerical stability.

Although the Maximum Correntropy Criterion (MCC) outperforms traditional methods in handling outliers and non-Gaussian noise, its efficiency can be compromised by unknown time-varying noise. To address this, Huang et al. proposed an adaptive MCC-based Kalman filter with a variable kernel, wherein the kernel width is a function of the residual [17]. Shi et al. introduced a variable kernel bandwidth approach into the MCC to enhance the algorithm's adaptability, devising a novel method to update the kernel bandwidth at each step [18]. More recently, Variational Bayesian methods have been employed to further improve the adaptability of the MCC. Liu et al. developed a Variational Bayesian-based cubature Kalman filter capable of addressing high-dimensional nonlinear problems [19]. Additionally, Li et al. proposed a Variational Bayesian-based Kalman filter that has demonstrated enhanced efficiency in environments with non-stationary noise [20]. By approximating the measurement noise through Variational Bayesian techniques, the algorithm can achieve improved adaptability to unknown time-varying noise.

In practical applications in the navigation field, the MCC-based filter has shown significant promise. Li et al. applied the MCC-based filter to INS/GNSS integrated navigation systems [21], conducting experiments to verify its performance against existing filtering algorithms when facing outliers. Sirish et al. employed the correntropy Kalman filter to improve GPS position estimation during periods of reduced GPS signal visibility [22]. Moreover, while studies on underwater navigation using the MCC are sparse, Wang et al. implemented an MCC-based Kalman filter for underwater SINS/USBL navigation to cope with non-Gaussian measurement noise [23].

In this study, we address challenges posed by non-Gaussian and time-varying measurement noise in the navigation systems of Autonomous Underwater Vehicles (AUVs) operating in complex underwater environments. We introduce a novel approach, the

Variational Bayesian and Gaussian–Newton method-based Iterated Maximum Mixture Correntropy Unscented Kalman Filter (VBGN-IMMCUKF) and integrated it into the navigation algorithm for AUVs. To evaluate the efficacy of our proposed algorithm, we conducted both simulations and sea trials, comparing its performances against those of conventional filters. The contributions of this paper are threefold:

(1)   We improve the MCUKF using a mixture kernel and Gaussian–Newton method, specifically for AUV integrated navigation. These significantly improve the robustness of navigation within complex environments.

(2)   The VBGN-IMMCUKF is introduced for AUV integrated navigation, employing Variational Bayesian techniques to approximate unknown time-varying noise, thereby increasing the navigation system's adaptability.

(3)   We conducted simulation experiments and sea trials to validate the navigation accuracy and robustness of our proposed algorithm. The results demonstrate that our algorithm surpasses both conventional algorithms and existing MCC-based algorithms in performance.

The structure of this paper is as follows. Section 2 introduces the AUV integrated navigation model and the Maximum Correntropy Criterion. Section 3 details the existing IMCUKF and our proposed algorithm. The performance of our proposed algorithm is analyzed based on the simulation and sea trial results in Section 4 and Section 5, respectively. Finally, Section 6 presents our overall conclusions.

2. Preliminaries

Before introducing the proposed algorithm based on the MCC, it is necessary to have some preliminary knowledge about AUV integrated navigation systems and the Maximum Correntropy Criterion.

2.1. AUV Integrated Navigation System

Currently, the integrated navigation system for AUVs primarily utilizes an INS and DVL. The INS provides vital data on the AUV's current attitude and acceleration, which are two critical components of navigation. The DVL, on the other hand, supplies velocity information in the carrier coordinate system. By fusing this data within a filtering algorithm, the AUV can effectively navigate and locate objects underwater.

Before introducing the proposed navigation algorithm, it is necessary to define the AUV's navigation coordinate system specifically: a north–east–down coordinate system. And the body-frame's forward–starboard–down coordinate system is also defined for the AUV. These systems of reference are depicted in Figure 1.
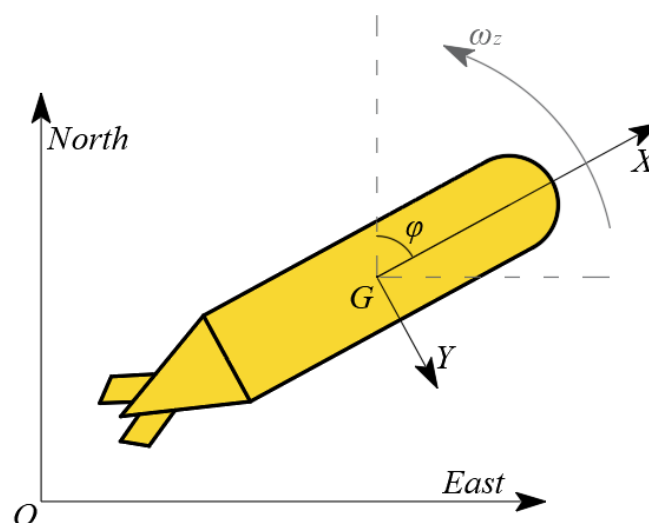
Figure 1. Navigation coordinate system of AUV.

Subsequently, we construct the integrated navigation model for the AUV based on a discrete-time state-space framework [24,25]. The state vector of the AUV's integrated navigation system at time step k is described as follows:

$$X_k = [x_k \quad y_K \quad \varphi_k \quad u_k \quad v_k \quad a_{x,k} \quad a_{y,k} \quad \omega_{z,k}]^T \tag{1}$$

where $\{x_k, y_k\}$ are the current north and earth position at time k, and φ_k denotes the current heading of the AUV at time k. $\{u_k, v_k\}$ are the current forward and starboard bottom-track velocities in body coordinates at time k. $\{a_{x,k}, a_{y,k}\}$ denote the forward and starboard acceleration in body coordinates at time k. $\omega_{z,k}$ denotes the angular velocity of the heading at time k.

The state equation of the navigation system is given by

$$X_k = f(X_{k-1}, w_{k-1}) \tag{2}$$

$$\begin{bmatrix} x_k \\ y_k \\ \varphi \\ u_k \\ v_k \\ a_{x,k} \\ a_{y,k} \\ \omega_{z,k} \end{bmatrix} = \begin{bmatrix} x_{k-1} + (u_{k-1}t + \frac{1}{2}a_{x,k-1}t^2)cos(\varphi) - (v_{k-1}t + \frac{1}{2}a_{y,k-1}t^2)sin(\varphi) + q_{x,k-1} \\ y_{k-1} + (u_{k-1}t + \frac{1}{2}a_{x,k-1}t^2)sin(\varphi) + (v_{k-1}t + \frac{1}{2}a_{y,k-1}t^2)cos(\varphi) + q_{y,k-1} \\ \varphi_{k-1} + \omega_{z,k-1}t + q_{\varphi,k-1} \\ u_{k-1} + a_{x,k-1}t + q_{u,k-1} \\ v_{k-1} + a_{y,k-1}t + q_{v,k-1} \\ a_{x,k-1} + q_{a_x,k-1} \\ a_{y,k-1} + q_{a_y,k-1} \\ \omega_{z,k-1} + q_{\omega_z,k-1} \end{bmatrix} \tag{3}$$

where w_{k-1} denotes the process noise at time $k-1$. Here, in assuming that they agree with the zero-mean Gaussian distribution,

$$q \sim \mathcal{N}(0, Q) \tag{4}$$

where Q is the covariance matrix of the process noise:

$$Q = diag[\sigma_{w_x}^2 \quad \sigma_{w_y}^2 \quad \sigma_{w_\varphi}^2 \quad \sigma_{w_u}^2 \quad \sigma_{w_v}^2 \quad \sigma_{w_{a_x}}^2 \quad \sigma_{w_{a_y}}^2 \quad \sigma_{w_{\omega_z}}^2] \tag{5}$$

The measurement vector at time k is given by

$$Z_k = [\varphi_k^{INS} \quad u_k^{DVL} \quad v_k^{INS} \quad a_{x,k}^{DVL} \quad a_{y,k}^{INS} \quad \omega_{z,k}^{INS}]^T \tag{6}$$

where $\{\varphi_k^{INS}, a_{x,k}^{INS}, a_{y,k}^{INS}, \omega_{z,k}^{INS}\}$ are the yaw, body frame acceleration, and yaw angular velocity measured by the INS respectively, $\{u_k^{DVL}, v_k^{DVL}\}$ denote the body frame bottom-track velocity measured by the DVL.

And the measurement equation is defined as

$$Z_k = H_k X_k + r_k \tag{7}$$

$$H_k = [0_{6\times2} \quad I_{6\times6}] \tag{8}$$

where r_k denotes the measurement noise at time k. Here, in assuming that they agree with the zero-mean Gaussian distribution.

$$r \sim \mathcal{N}(0, R) \tag{9}$$

where R is the covariance matrix of the measurement noise:

$$R = diag[\sigma_{r_\varphi}^2 \quad \sigma_{r_u}^2 \quad \sigma_{r_v}^2 \quad \sigma_{r_{a_x}}^2 \quad \sigma_{r_{a_y}}^2 \quad \sigma_{r_{\omega_z}}^2] \tag{10}$$

2.2. Max Correntropy Criterion

Correntropy is a newly presented quantity that represents the difference between two random variables [26]. In assuming that there are two random variables $A_k \in \mathbb{R}^d$ and $B_k \in \mathbb{R}^d$, their correntropy $V(A, B)$ can be expressed as

$$V(A, B) = \mathbb{E}(\kappa(A, B)) = \iint \kappa(a, b) dF_{AB}(a, b) \tag{11}$$

where *E[.]* denotes the expectation operation, and $F_{AB}(a, b)$ is the joint probability distribution of *A* and *B*. $\kappa(A, B)$ denotes a shift-invariant kernel function that follows the Mercer condition. The Gaussian kernel is utilized as the kernel function in this paper:

$$\kappa(a, b) = G_\sigma(e) = exp(-\frac{e^2}{2\sigma^2}) \tag{12}$$

where $e = a - b$, and $\sigma > 0$ denotes the bandwidth of the Gaussian kernel function.

The joint probability distribution $F_{AB}(a, b)$ can only be approximated through samples in practice:

$$\widehat{V}(A, B) = \frac{1}{N} \sum_{i=1}^{N} G_\sigma(e(i)) \tag{13}$$

where $e(i) = a(i) - b(i)$. $\{a(i), b(i)\}_{i=1}^{N}$ are N samples obtained from the joint density $F_{AB}(a, b)$.

The Taylor series expansion of Equation (13) is given by

$$V(A, B) = \sum_{n=0}^{\infty} \frac{(-1)^n}{2^n \sigma^{2n} n!} E[(A - B)^{2n}] \tag{14}$$

Equation (14) demonstrates that correntropy represents the weighted sum of the second moment and all higher-order moments of $(A - B)$. In adjusting the Gaussian kernel bandwidth σ, the weights assigned to these moments can be modulated; as σ increases, the influence of the higher-order moments in the calculation decreases significantly. Only when $(A = B)$, the correntropy function reaches its maximum.

Inspired by the correntropy function, the Maximum Correntropy Criterion (MCC) is proposed. This criterion achieve the optimal estimate by constructing and maximizing a correntropy function that expresses the discrepancy between estimated and observed values. Equation (14) illustrates the capability of the correntropy to accommodate estimations at higher-order moments, which is particularly effective in addressing non-Gaussian noise.

In AUV positioning and navigation, sensor measurements are susceptible to disturbances in complex underwater environments, which may introduce outliers or non-Gaussian noise. The typical assumption of Gaussian noise in traditional navigation algorithms may be not effective under these conditions. The Maximum Correntropy Criterion is particularly beneficial for addressing such challenges, enhancing the robustness and accuracy of navigation algorithms in complex noise environments.

3. The Variational Bayesian and Gaussian-Newton Method Based IMMCUKF

The existing Iterated Maximum Correntropy Unscented Kalman Filter (IMCUKF) [13] has demonstrated proficiency in managing non-Gaussian noise issues, yet there are specific areas where enhancements are feasible. Firstly, within the IMCUKF algorithm, despite the accommodation for non-Gaussian noise distributions, the use of a static measurement noise covariance matrix *R* may not be sufficiently adaptive to unknown or dynamic noise conditions. Secondly, the reliance on a Gaussian kernel bandwidth can lead to a diminished estimation performance in increasingly complex engineering scenarios. Thirdly, refining the efficiency of the iterative process could significantly improve the practical deployment of the algorithm.

To address these limitations, the Variational Bayesian and Gaussian–Newton method-based Iterated Maximum Mixture Correntropy Unscented Kalman Filter (VBGN-IMMCUKF) is introduced. This advanced algorithm aims to enhance adaptability to varying noise profiles, optimize the estimation performance in multifaceted environments, and streamline the iterative efficiency, thus improving its practical applicability.

3.1. Gaussian–Newton Method-Based Iterated Maximum Mixture Correntropy UKF

Suppose that there is a nonlinear model

$$\begin{cases} X_k = f(X_{k-1}, q_{k-1}) \\ Z_k = h(X_k, r_k) \end{cases} \tag{15}$$

where $f(.)$ is the dynamic model, and $h(.)$ is the measurement model. $X_k \in \mathbb{R}^N$ and $Z \in \mathbb{R}^M$ are the state and measurement of the system at time k. q_{k-1} and r_k are the process noise and measurement noise at time $k-1$, where $q_{k-1} \sim \mathcal{N}(0, Q_{k-1})$, and $r_{k-1} \sim \mathcal{N}(0, R_{k-1})$.

Both the MCC-based Unscented Kalman Filter and the traditional Unscented Kalman Filter (UKF) utilize the same process for prior estimates. In the UKF, the state distribution is represented through an Unscented Transform, which involves the weighted sampling of a set of sigma points. The prior state estimate and its corresponding covariance are derived through updating these sample points and computing their weighted sum, as illustrated in Equations (16) and (17).

$$\begin{cases} X_{0,k-1|k-1} = \widehat{X}_{k-1|k-1} \\ X_{i,k-1|k-1} = \widehat{X}_{k-1|k-1} + \sqrt{(n+k)P_{k-1|k-1}} & i = 1, 2, \ldots n, \\ X_{i,k-1|k-1} = \widehat{X}_{k-1|k-1} - \sqrt{(n+k)P_{k-1|k-1}} & i = n+1, n+2 \ldots 2n, \end{cases} \tag{16}$$

where $X_{i,k-1}, i = 1, 2, \ldots n$ are sigma points generated by the posterior estimation of state at time $k-1$, and $w_m^{(i)}$ and $w_c^{(i)}$ are the weights of the sigma points.

$$\begin{cases} \widehat{X}_{k|k-1}^- = \sum\limits_{i=0}^{2n} w_m^{(i)} f(X_{i,k-1|k-1}), \\ P_{k|k-1}^- = \sum\limits_{i=0}^{2n} w_c^{(i)} (X_{i,k|k-1} - X_{k|k-1}^-)(X_{i,k|k-1} - X_{k|k-1}^-)^T + Q_k. \end{cases} \tag{17}$$

where $X_{k|k-1}^-$ and $P_{k|k-1}^-$ are the prior estimate and covariance matrix of the state at time k. $w_m^{(i)}$ and $w_c^{(i)}$ are the weights of the sigma points, with $w_m^{(0)} = \frac{\lambda}{n+\lambda}$, $w_c^{(0)} = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta)$, $w_m^{(i)} = w_c^{(i)} = \frac{1}{n+\lambda}, i = 1, 2, 3 \ldots 2n$, and $\lambda = \alpha^2(n+\kappa) - n$.

Once the measurements are acquired, the measurement update process is performed. Firstly, a new set of sigma points is acquired through sampling the states and covariance matrix derived from the prior estimate:

$$\begin{cases} X_{0,k|k-1} = \widehat{X}_{k|k-1} \\ X_{i,k|k-1} = \widehat{X}_{k|k-1} + \sqrt{(n+k)P_{k|k-1}} & i = 1, 2, \ldots n, \\ X_{i,k|k-1} = \widehat{X}_{k|k-1} - \sqrt{(n+k)P_{k|k-1}} & i = n+1, n+2 \ldots 2n, \end{cases} \tag{18}$$

Then, calculate $\widehat{Z}_{k|k-1}$, P_{zz}, and P_{xz}:

$$\begin{cases} \widehat{Z}_{k|k-1} = \sum\limits_{i=0}^{2n} w_m^i h(X_{i,k|k-1}), \\ P_{zz} = \sum\limits_{i=0}^{2n} w_c^i (h(X_{i,k|k-1}) - \widehat{Z}_{k|k-1})(h(X_{i,k|k-1}) - \widehat{Z}_{k|k-1})^T + R_k, \\ P_{xz} = \sum\limits_{i=0}^{2n} w_c^i (X_{i,k|k-1} - X_{k|k-1}^-)(h(X_{i,k|k-1}) - \widehat{Z}_{k|k-1})^T. \end{cases} \tag{19}$$

As introduced in Section 2.2, for an MCC-based KF, the correntropy cost function is constructed. To overcome the limitation of single-kernel correntropy in dealing with complex problems, mixture correntropy is utilized. The mixture correntropy is defined as

$$V(X, Y) = \mathbb{E}[\mu G_{\sigma_1}(e) + (1 - \mu)G_{\sigma_2}(e)] \tag{20}$$

where σ_1 and σ_2 are different Gaussian kernel bandwidths of the mixture correntropy, and $0 \leq \mu \leq 1$ is the mixture weight.

Equation (20) reveals that the mixture correntropy incorporates different kernel functions, each with distinct bandwidths. The curve of the mixture correntropy distribution, characterized by bandwidth parameters $\sigma_1 = 1$ and $\sigma_2 = 10$ for varying weights μ, is depicted in Figure 2. This illustration demonstrates that the application of mixture correntropy can improve the Maximum Correntropy Criterion's effectiveness in managing complex noise environments. It is evident that the mixture correntropy reduces to the original correntropy when μ equals 0 or 1.
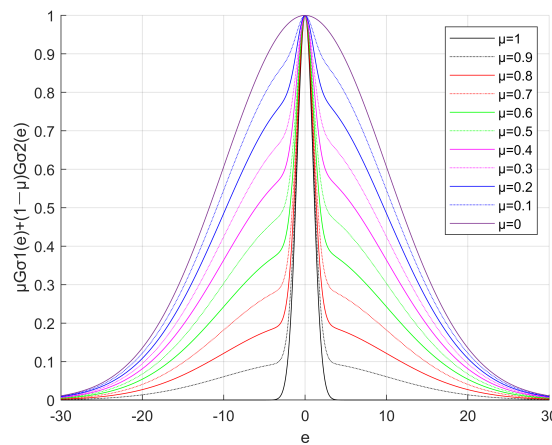
Figure 2. Mixed Gaussian kernel function with different weights μ.

Then, inspired by the weighted least squares (WLS) method, the mixture correntropy cost function is constructed as follows:

$$\begin{aligned} J(X_k) = {}& \alpha_1 \mu G_{\sigma_1}(\|X_k - \widehat{X}_{k|k-1}\|_{P_{k|k-1}^{-1}}) + \alpha_2(1 - \mu)G_{\sigma_2}(\|X_k - \widehat{X}_{k|k-1}\|_{P_{k|k-1}^{-1}}) \\ &+ \beta_1 \mu G_{\sigma_1}(\|Z_k - h(X_k)\|_{R_k^{-1}}) + \beta_2(1 - \mu)G_{\sigma_2}(\|Z_k - h(X_k)\|_{R_k^{-1}}) \end{aligned} \tag{21}$$

where α_1, α_2, β_1, and β_2 are adjusting weights, and $\|X\|_A^2 = X^T A X$. Then, the posterior estimation can be obtained by solving the following equation:

$$\widehat{X}_{k|k} = argmax\, J(X_k) \tag{22}$$

Due to the nonlinear characteristic of the AUV navigation model, Equation (3) lacks an analytical solution. To address this challenge, the Gaussian–Newton iteration method is employed, as it has demonstrated superior effectiveness and convergence in resolving iterative issues [15]. The Gaussian–Newton method iteration, from steps (t) to $(t + 1)$, performs the following equation:

$$\widehat{X}_{k|k}^{(t+1)} = \widehat{X}_{k|k}^{(t)} - ((\eta(X))^{-1}\nabla J(X))\big|_{X=\widehat{X}_{k|k}^{(t)}} \tag{23}$$

where the terms $\eta(X)$ and $\nabla J(X)$ denote the Hessian and gradient of the cost function $J(X)$, respectively. The partial derivatives of $G_\sigma(\|X_k - \widehat{X}_{k|k-1}\|_{P_{k|k-1}^{-1}})$ and $G_\sigma(\|Z_k - h(X_k)\|_{R_k^{-1}})$ over X_k can be calculated as

$$\frac{\partial G_\sigma(E)}{\partial X_k}\Big|_{E=\|X_k-\widehat{X}_{k|k-1}\|_{P_{k|k-1}^{-1}}} = G_\sigma(E)\frac{E}{\sigma^2}\frac{\partial E}{\partial X_k}\Big|_{E=\|X_k-\widehat{X}_{k|k-1}\|_{P_{k|k-1}^{-1}}}$$

$$= G_\sigma(\|X_k-\widehat{X}_{k|k-1}\|_{P_{k|k-1}^{-1}})\frac{(X_k-\widehat{X}_{k|k-1})P_{k|k-1}^{-1}}{\sigma^2} \tag{24}$$

$$\frac{\partial G_\sigma(E)}{\partial X_k}\Big|_{E=\|Z_k-h(X_k)\|_{R_k^{-1}}} = G_\sigma(E)\frac{E}{\sigma^2}\frac{\partial E}{\partial X_k}\Big|_{E=\|Z_k-h(X_k)\|_{R_k^{-1}}}$$

$$= G_\sigma(\|Z_k-h(X_k)\|_{R_k^{-1}})\frac{\tilde{H}_k(Z_k-h(X_k))R_k^{-1}}{\sigma^2} \tag{25}$$

where $\tilde{H}_k^{(t)} = \frac{\partial h(X_k)}{\partial X_k}$, which can be approximated as follows [27]:

$$H_k = P_{xz}^T(P_{k|k-1})^{-1} \tag{26}$$

Therefore, the gradient $\nabla J(X)\big|_{X=\widehat{X}_{k|k}^{(t)}}$ can be calculated as

$$\nabla J(X)\big|_{X=\widehat{X}_{k|k}^{(t)}} = L_P^{(t)}P_{k|k-1}^{-1}(\widehat{X}_{k|k}^{(t)}-\widehat{X}_{k|k-1}) - L_R^{(t)}(\tilde{H}_k^{(t)})^T R_k^{-1}(Z_k-h(\widehat{X}_{k|k}^{(t)})) \tag{27}$$

The $L_P^{(t)}$ and $L_R^{(t)}$ are given by

$$L_P^{(t)} = [\frac{\alpha_1\mu}{\sigma_1^2}G_{\sigma_1}(\|\widehat{X}_{k|k}^{(t)}-\widehat{X}_{k|k-1}\|_{P_{k|k-1}^{-1}}) + \frac{\alpha_2(1-\mu)}{\sigma_2^2}G_{\sigma_2}(\|\widehat{X}_{k|k}^{(t)}-\widehat{X}_{k|k-1}\|_{P_{k|k-1}^{-1}})], \tag{28}$$

$$L_R^{(t)} = [\frac{\beta_1\mu}{\sigma_1^2}G_{\sigma_1}(\|Z_k-h(\widehat{X}_{k|k}^{(t)})\|_{R_k^{-1}}) + \frac{\beta_2(1-\mu)}{\sigma_2^2}G_{\sigma_2}(\|Z_k-h(\widehat{X}_{k|k}^{(t)})\|_{R_k^{-1}})]. \tag{29}$$

In defining weights $\alpha_1 = \beta_1 = \sigma_1^2$ and $\alpha_2 = \beta_2 = \sigma_2^2$,

$$L_P^{(t)} = [\mu G_{\sigma_1}(\|\widehat{X}_{k|k}^{(t)}-\widehat{X}_{k|k-1}\|_{P_{k|k-1}^{-1}}) + (1-\mu)G_{\sigma_2}(\|\widehat{X}_{k|k}^{(t)}-\widehat{X}_{k|k-1}\|_{P_{k|k-1}^{-1}})], \tag{30}$$

$$L_R^{(t)} = [\mu G_{\sigma_1}(\|Z_k-h(\widehat{X}_{k|k}^{(t)})\|_{R_k^{-1}}) + (1-\mu)G_{\sigma_2}(\|Z_k-h(\widehat{X}_{k|k}^{(t)})\|_{R_k^{-1}})]. \tag{31}$$

Then, the Hessian $\eta(X)$ can be calculated by

$$\eta(X)\big|_{X=\widehat{X}_{k|k}^{(t)}} = L_P^{(t)}P_{k|k-1}^{-1} + L_R^{(t)}(\tilde{H}_k^{(t)})^T R_k^{-1}\tilde{H}_k^{(t)} \tag{32}$$

It is noteworthy that the MCC based algorithm will encounter issues of numerical instability when large measurement noise occurs, which results in the values of L_P and L_R being excessively small. To mitigate this, a minimum threshold ε is established for the computation of L_P and L_R . The modified $\widehat{L}_P$ and $\widehat{L}_R$ are defined as follows:

$$\widehat{L}_P = Max\{L_P,\varepsilon\}, \widehat{L}_R = Max\{L_R,\varepsilon\}. \tag{33}$$

The Gaussian–Newton method iteration from steps (t) to (t + 1) is given by

$$\begin{aligned}\widehat{X}_{k|k}^{(t+1)} =\,&\widehat{X}_{k|k}^{(t)} - (L_P^{(t)}P_{k|k-1}^{-1} + L_R^{(t)}(\tilde{H}_k^{(t)})^T R_k^{-1}\tilde{H}_k^{(t)})^{-1}(L_P^{(t)}P_{k|k-1}^{-1}(\widehat{X}_{k|k}^{(t)}-\widehat{X}_{k|k-1}) \\ &- L_R^{(t)}(\tilde{H}_k^{(t)})^T R_k^{-1}(Z_k-h(X_{k|k}^{(t)})))\end{aligned} \tag{34}$$

Substituting

$$\widehat{X}_{k|k}^{(t)} = (L_P^{(t)} P_{k|k-1}^{-1} + L_R^{(t)} (\tilde{H}_k^{(t)})^T R_k^{-1} \tilde{H}_k^{(t)})^{-1} (L_P^{(t)} P_{k|k-1}^{-1} + L_R^{(t)} (\tilde{H}_k^{(t)})^T R_k^{-1} \tilde{H}_k^{(t)}) \widehat{X}_{k|k}^{(t)} \quad (35)$$

into Equation (34), we have

$$\begin{aligned}
\widehat{X}_{k|k}^{(t+1)} =& (L_P^{(t)} P_{k|k-1}^{-1} + L_R^t (\tilde{H}_k^{(t)})^T R_k^{-1} \tilde{H}_k^{(t)})^{-1} (L_P^{(t)} P_{k|k-1}^{-1} \widehat{X}_{k|k-1} \\
&+ L_R^{(t)} (\tilde{H}_k^{(t)})^T R_k^{-1} \tilde{H}_k^{(t)} \widehat{X}_{k|k}^{(t)} + L_R^{(t)} (\tilde{H}_k^{(t)})^T R_k^{-1} (Z_k - h(X_{k|k}^{(t)})))
\end{aligned} \quad (36)$$

Then, adding and subtracting $(L_R^{(t)} (\tilde{H}_k^{(t)})^T R_k^{-1}) \tilde{H}_k^{(t)} \widehat{X}_{k|k-1}$ from the left-hand side of the equation, we have

$$\begin{aligned}
\widehat{X}_{k|k}^{(t+1)} =& (L_P^{(t)} P_{k|k-1}^{-1} + L_R^{(t)} (\tilde{H}_k^{(t)})^T R_k^{-1} \tilde{H}_k^{(t)})^{-1} ((L_P^{(t)} P_{k|k-1}^{-1} + L_R^{(t)} (\tilde{H}_k^{(t)})^T R_k^{-1} \tilde{H}_k^{(t)}) \widehat{X}_{k|k-1} \\
&- (L_R^{(t)} (\tilde{H}_k^{(t)})^T R_k^{-1} \tilde{H}_k^{(t)}) \widehat{X}_{k|k-1} + L_R^{(t)} (\tilde{H}_k^{(t)})^T R_k^{-1} \tilde{H}_k^{(t)} \widehat{X}_{k|k}^{(t)} \\
&+ L_R^{(t)} (\tilde{H}_k^{(t)})^T R_k^{-1} (Z_k - h(X_{k|k}^{(t)})))
\end{aligned} \quad (37)$$

After reorganization, the posterior estimate from step (t) to step (t + 1) is given by

$$\begin{aligned}
\widehat{X}_{k|k}^{(t+1)} =& \widehat{X}_{k|k-1} + (L_P^{(t)} P_{k|k-1}^{-1} + L_R^{(t)} (\tilde{H}_k^{(t)})^T R_k^{-1} \tilde{H}_k^{(t)})^{-1} (-(L_R^{(t)} (\tilde{H}_k^{(t)})^T R_k^{-1} \tilde{H}_k^{(t)}) \widehat{X}_{k|k-1} \\
&+ L_R^{(t)} (\tilde{H}_k^{(t)})^T R_k^{-1} \tilde{H}_k^{(t)} \widehat{X}_{k|k}^{(t)} + L_R^{(t)} (\tilde{H}_k^{(t)})^T R_k^{-1} (Z_k - h(X_{k|k}^{(t)})))
\end{aligned} \quad (38)$$

The Gaussian–Newton method iteration of the posterior estimate from step (*t*) to step (*t* + 1) can be summarized as follows:

$$\begin{cases}
\widehat{X}_{k|k}^{(t+1)} = \widehat{X}_{k|k-1} + K_k^{(t)} (Z_k - h(X_{k|k}^{(t)}) + \tilde{H}_k^{(t)} \widehat{X}_{k|k}^{(t)} - \tilde{H}_k^{(t)} \widehat{X}_{k|k-1}) \\
K_k^{(t)} = (L_P^{(t)} P_{k|k-1}^{-1} + L_R^{(t)} (\tilde{H}_k^{(t)})^T R_k^{-1} \tilde{H}_k^{(t)})^{-1} L_R^{(t)} (\tilde{H}_k^{(t)})^T R_k^{-1}
\end{cases} \quad (39)$$

In addition, the estimate error $\widetilde{X}_{k|k}^{(t+1)}$ can be expressed as

$$\begin{aligned}
\widetilde{X}_{k|k}^{(t+1)} &= X_k - \widehat{X}_{k|k}^{(t+1)} \\
&= X_k - \widehat{X}_{k|k-1} - K_k^{(t)} [Z_k - h(X_{k|k}^{(t)}) + \tilde{H}_k^{(t)} (\widehat{X}_{k|k}^{(t)} - \widehat{X}_{k|k-1})]
\end{aligned} \quad (40)$$

In substituting $Z_k - h(X_{k|k}^{(t)}) = \tilde{H}_k^{(t)} X_k^{(t)} + r_k - \tilde{H}_k^{(t)} \widehat{X}_{k|k}^{(t)}$,

$$\begin{aligned}
\widetilde{X}_{k|k}^{(t+1)} &= X_k - \widehat{X}_{k|k} - K_k^{(t)} [\tilde{H}_k^{(t)} (X_k^{(t)} - \widehat{X}_{k|k}^{(t)}) + r_k + \tilde{H}_k^{(t)} (\widehat{X}_{k|k}^{(t)} - \widehat{X}_{k|k-1})] \\
&= (I - K_k^{(t)} \tilde{H}_k^{(t)}) (X_k - \widehat{X}_{k|k-1}) + K_k^{(t)} r_k
\end{aligned} \quad (41)$$

Therefore, the covariance matrix of the posterior estimated state $P_{k|k}$ can be approximated as follows:

$$P_{k|k}^{(t+1)} = (I - K_k^{(t)} \tilde{H}_k^{(t)}) P_{k|k-1} (I - K_k^{(t)} \tilde{H}_k^{(t)})^T + K_k^{(t)} R_k (K_k^{(t)})^T \quad (42)$$

3.2. Variational Bayesian Improvement

In practice, the constant measurement noise covariance *R* constrains the algorithm's performance when dealing with time-varying measurements. To overcome this limitation,

we integrate the Variational Bayesian (VB) method with the MCC to adaptively estimate R, thereby enhancing the adaptability of the navigation system.

The VB method is a statistical approach commonly employed for managing complex probabilistic models. Utilizing the VB method, we can approximate the joint posterior probability distribution of the system state and measurement noise at time k as follows:

$$p(X_k, R_k | Z_{1:k}) \approx q_X(X_k) q_R(R_k) \tag{43}$$

where $q_X(X_k)$ and $q_R(R_k)$ are the approximating density of system state and measurement noise covariance. Then, the VB-approximated distribution can be calculated by minimizing the Kullback–Leibler (*KL*) divergence between the approximation and true distribution. The *KL* divergence is defined as

$$KL[q_X(X_k) q_R(R_k) \| p(X_k, R_k | Z_{1:K})] = \int q_X(X_k) q_R(R_k) log(\frac{q_X(X_k) q_R(R_k)}{p(X_k, R_k | Z_{1:K})}) dX_k dR_k \tag{44}$$

Minimizing the *KL* divergence, we obtain the following equations:

$$q_X(X_k) \propto exp(\int log(p(Z_k, X_k, R_k | Z_{1:K-1}) q_R(R_k)) dR_k) \tag{45}$$

$$q_R(R_k) \propto exp(\int log(p(Z_k, X_k, R_k | Z_{1:K-1}) q_X(X_k)) dX_k) \tag{46}$$

Therefore, the posterior probability distribution can be approximated as a product of the Gaussian distribution and Inverse Wishart (IW) distribution:

$$p(X_k, R_k | Z_{1:k}) \approx \mathcal{N}(X_k | \widehat{X}_k, P_k) IW(R_k | \gamma_k, V_k) \tag{47}$$

$$\mathcal{N}(X_k | \widehat{X}_k, P_k) \propto |P_k|^{-\frac{1}{2}} exp(-\frac{1}{2}(X_k - \widehat{X}_k)^T P_k^{-1}(X_k - \widehat{X}_k)) \tag{48}$$

$$IW(R_k | \gamma_k, V_k) \propto |R_k|^{-\frac{\gamma_k + n + 1}{2}} exp(-\frac{1}{2} tr(V_k R_k^{-1})) \tag{49}$$

where γ_k and V_k are the degree-of-freedom parameter and scale matrix of the Inverse Wishart distribution, respectively, and n denotes the dimension of the state. tr(.) denotes the trace of the matrix. Therefore, the approximated measurement noise covariance $\widehat{R}_k$ can be calculated as the mean of the IW distribution:

$$\widehat{R}_k = (\gamma_k - n - 1)^{-1} V_k \tag{50}$$

The dynamic model of the IW distribution is given by

$$\gamma_{k|k-1} = \rho(\gamma_{k-1|k-1} - n - 1) + n + 1 \tag{51}$$

$$V_{k|k-1} = B V_{k-1|k-1} B^T \tag{52}$$

where ρ is the forgetting factor $0 < \rho < 1$, and B $= \sqrt{\rho} I$.

The posterior update process $P(R_k | Z_{1:k})$ is given by

$$\gamma_{k|k} = \gamma_{k|k-1} + 1 \tag{53}$$

$$V_{k|k} = V_{k|k-1} + \sum_{i=0}^{2n} \omega_i (Z_k - Z_{i,k|k-1})(Z_k - Z_{i,k|k-1})^T \tag{54}$$

Therefore, we can approximate R_k using the VB method during iteration. The VBGN-IMMCUKF algorithm is summarized in Algorithm 1.

Algorithm 1 VBGN-IMMCUKF

1: Initialize $\widehat{X}_{0|0}$, $P_{0|0}$, γ_0, Y_0

2: **for** time $k = 1, 2, 3 \dots$ **do**

3:      **Prior Predict**

4:      Calculate $\widehat{X}_{k|k-1}$ and $P_{k|k-1}$ by (16), (17);

5:      Calculate $\gamma_{k|k-1}$ and $V_{k|k-1}$ by (50), (51);

6:      **Posterior Update**

7:      set $X_{k|k}^{(0)} = \widehat{X}_{k|k-1}$, $P_{k|k}^{(0)} = P_{k|k-1}$, $\gamma_{k|k} = 1 + \gamma_{k|k-1}$, $V_{k|k}^{(0)} = V_{k|k-1}$;

8:      **for** iteration $t = 0, 1, 2 \dots, N$ **do**

9:          Calculate $X_{i,k|k-1}$, $\widehat{Z}_{k|k-1}$, P_{ZZ} and P_{XZ} by (18) and (19);

10:          Calculate $\widehat{R}_k^{(t)}$, $H_k^{(t)}$ by (50) and (26);

11:          Calculate $L_P^{(t)}$, $L_R^{(t)}$ by (30) and (31);

12:          Calculate $\widehat{X}_{k|k}^{(t+1)}$, $P_{k|k}^{(t+1)}$, $V_{k|k}^{(t+1)}$ by (39), (42), (54);

13:          **if** $\frac{\|\widehat{X}_{k|k}^{(t+1)} - \widehat{X}_{k|k}^{(t)}\|}{\|\widehat{X}_{k|k}^{(t)}\|} < \epsilon$ **then**

14:              $\widehat{X}_{k|k} = \widehat{X}_{k|k}^{(t+1)}$, $P_{k|k} = P_{k|k}^{(t+1)}$, $V_{k|k} = V_{k|k}^{(t+1)}$, break.

15:          **end if**

16:      **end for**

17: **end for**

---

4. Simulation Analysis

Simulations were conducted to evaluate the performance of the proposed algorithm, the Variational Bayesian and Gaussian–Newton method-based Iterated Maximum Mixture Correntropy Unscented Kalman Filter (VBGN-IMMCUKF), in comparison with several established algorithms: the Extended Kalman Filter (EKF), the Unscented Kalman Filter (UKF), the Iterated Maximum Correntropy Unscented Kalman Filter (IMCUKF), the Iterated Maximum Mixture Correntropy Unscented Kalman Filter (IMMCUKF), and the Gaussian–Newton method-based Iterated Maximum Mixture Correntropy Unscented Kalman Filter (GN-IMMUKF). Each algorithm was implemented on the AUV navigation model introduced in Section 2.1. Additionally, the AUV dynamic models presented in [28] were employed to generate the simulation data.

4.1. Simulation Settings

To comprehensively assess the performance of each algorithm, three simulations were executed. The algorithms' performances were quantified using the Root Mean Square Error (*RMSE*) and navigation accuracy. The *RMSE*, Average Root Mean Square Error (*ARMSE*), and accuracy were calculated according to the following equations:

$$RMSE_{pos}(\mathrm{m}) = \sqrt{\frac{1}{N} \sum_{k=1}^{N} ((\widehat{x}_k - x_k)^2 + (\widehat{y}_k - y_k)^2)} \tag{55}$$

$$RMSE_{vel}(\mathrm{m/s}) = \sqrt{\frac{1}{N} \sum_{k=1}^{N} ((\widehat{u}_k - u_k)^2 + (\widehat{v}_k - v_k)^2)} \tag{56}$$

$$ARMSE_{pos}(\mathrm{m}) = \frac{1}{M} \sum_{i=1}^{M} RMSE_{pos,i} \tag{57}$$

$$ARMSE_{vel}(\text{m/s}) = \frac{1}{M}\sum_{i=1}^{M} RMSE_{vel,i} \tag{58}$$

$$Accuracy(\%) = \frac{RMSE}{Total\ Distance} \times 100 \tag{59}$$

where N is the total time of one simulation, and M is the number of Monte Carlo runs.

The details of each simulation case are presented below. Firstly, some filter parameters for conventional algorithms in each simulation case were configured as follows.

The covariance matrix of the noise Q_k, R_k, and initial state covariance matrix P_0 are given by

$$\begin{cases} Q_k = 0.1.diag(I_{8\times1}), \\ R_k = 0.001.diag(I_{6\times1}), \\ P_0 = 0.1.diag(I_{8\times1}). \end{cases} \tag{60}$$

The parameters of the UKF in Equation (17) are given by

$$\begin{cases} \alpha = 1, \\ \beta = 2, \\ \kappa = 0. \end{cases} \tag{61}$$

The bandwidth of a single-Gaussian kernel σ was set to 1; two kernel bandwidth and mixture weights of the mixture Gaussian kernel were set as follows:

$$\begin{cases} \sigma_1 = 2, \\ \sigma_2 = 10, \\ \mu = 0.5. \end{cases} \tag{62}$$

The initial parameters of the Variational Bayesian method were set as follows:

$$\begin{cases} \gamma = 10, \\ V = diag(I_{6\times6}). \end{cases} \tag{63}$$

The flag for the end of the iteration was set as $\epsilon = 10^{-6}$, and the minimum threshold in Equation (33) was set as $\varepsilon = 10^{-10}$.

In each simulation case, the AUV ran 1000 steps, with each step representing one second of motion, starting from an initial position of $x_0 = y_0 = 0$, with an initial heading of $0°$, and sustains a constant forward velocity of 1 m/s.

Different from other cases, the AUV in simulation case 1 had angular velocity of $4.5°$/s in 240–260 s, 490–510 s, and 740–760 s, resulting in a box trajectory of the AUV. A multi-Gaussian noise model $r \sim [0.99\mathcal{N}(0,0.1) + 0.01\mathcal{N}(1,10)]$ was added to the simulated DVL measurements to simulate the harsh measurement noise environment. The simulated DVL forward velocity for simulation case 1 is illustrated in Figure 3b. As can be seen from the figure, the analogous DVL measurements suffer from severe pollution due to the added harsh noise. So, the trajectory of each method was polluted, as illustrated in Figure 3a.

In simulation case 2, the AUV sustains an angular velocity of $0.36°/s$, leading to the circular trajectory of the AUV. The Gaussian noise with a time-varying covariance $r \sim [\mathcal{N}(0.5, \tilde{R})]$ was introduced to the DVL measurements, where covariance $\tilde{R} = 0.5$ during 100–200 s, $\tilde{R} = 0.4$ during 600–700 s, and $\tilde{R} = 0.1$ for the remaining time. The analogous DVL forward velocity for simulation case 2 is depicted in Figure 4b, and the simulated trajectory of each method is illustrated in Figure 4a.

For simulation case 3, the AUV had an angular velocity of $0.72°/s$ in 0–250 s and 500–750 s and $-0.72°/s$ in 250–500 s and 750–1000 s, which resulted in a lawnmower trajectory of the AUV. Different from other cases, a multi-Gaussian noise r with time-varying covariance was added to the DVL measurements: $r \sim [0.99\mathcal{N}(0, 0.1) + 0.01\mathcal{N}(1, \tilde{R})]$, where $\tilde{R} = 10$ during 100–200 s, $\tilde{R} = 9$ during 400–500 s, $\tilde{R} = 8$ during 600–700 s, and $\tilde{R} = 7$ for the remaining time. As shown in Figure 5b, due to the harsh noise, the analogous DVL measurement in simulation case 3 was severely disturbed.
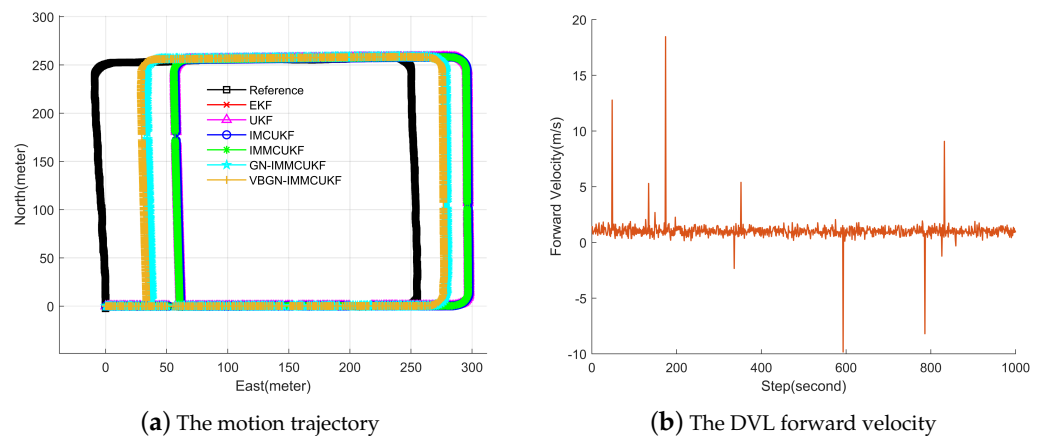

(**a**) The motion trajectory

(**b**) The DVL forward velocity

Figure 3. Trajectory and DVL forward velocity of simulation case 1.

(**a**) The motion trajectory

(**b**) The DVL forward velocity
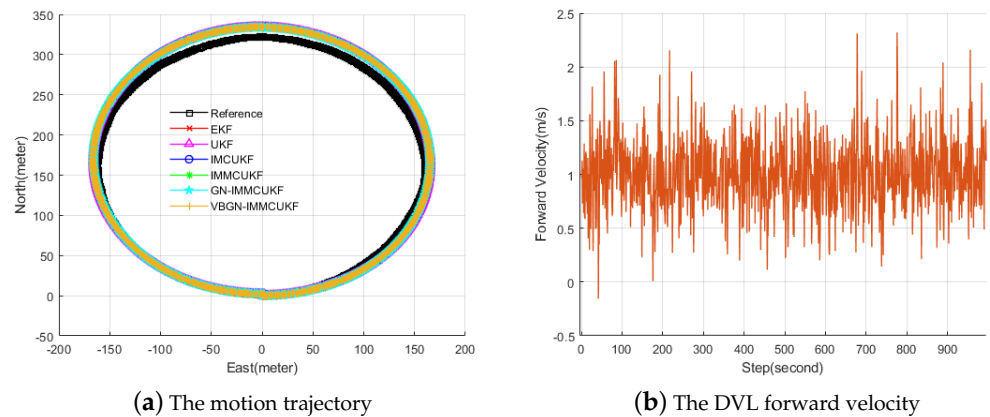
Figure 4. Trajectory and DVL forward velocity of simulation case 2.

(**a**) The motion trajectory

(**b**) The DVL forward velocity
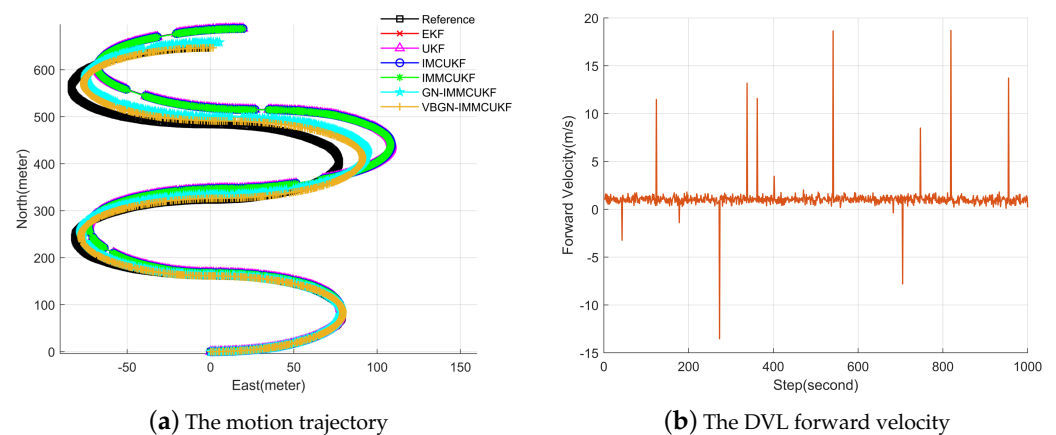
Figure 5. Trajectory and DVL forward velocity of simulation case 3.

4.2. Simulation Results

The experimental results for simulation case 1 are depicted in Figure 6. Figure 6a,c depict the *RMSE* changing during one simulation. As we can see from the diagrams, due to the added severe noise, outliers occurred. Upon the occurrence of outliers, the *RMSE* changes significantly; however, the proposed algorithm effectively suppress the change. The overall *RMSE* of 30 runs of the simulation is presented in Figure 6b,d. As can be seen from the diagrams, the proposed algorithm's overall *RMSE* is evidently lower than those of the other algorithms. The statistical results are recorded in Table 1, which reveals that for simulation case 1, the proposed algorithm shows certain superiority in both accuracy and *ARMSE*. Numerically, the proposed algorithm achieves a 44.67% improvement in the position *ARMSE* over the EKF, 44.74% over the UKF, 44.68% over the IMCUKF, 44.31% over the IMMCUKF, and 10.25% over the GN-IMMCUKF. In terms of navigation accuracy, the improvement is 43.28% over the EKF, 41.39% over the UKF, 42.35% over the IMCUKF, 41.19% over the IMMCUKF, and 9.42% over the GN-IMMCUKF. These statistical results substantiate the superior robustness of the proposed algorithm.

Figure 7 illustrates the experimental results of simulation case 2, which featured no outliers but was influenced by Gaussian noise with time-varying covariance. As can be seen from the figure, the overall velocity *RMSE* of proposed algorithm is evidently lower than those of the other algorithms, which proves that the proposed algorithm has a better estimation performance when the measurement noise changes. Table 1 records the statistical results of simulation case 2. It can be seen from the data that the proposed algorithm achieves a 3.61% improvement in accuracy over the EKF, 3.76% over the UKF, 3.04% over the IMCUKF, 2.46% over the IMMCUKF, and 2.25% over the GN-IMMCUKF. These findings confirm the proposed algorithm's improved navigation accuracy, which further proves that the proposed algorithm can better estimate under time-varying measurement noise, improving the adaptability of the navigation system.

The experimental results of simulation case 3 are shown in Figure 8. Figure 8a,c illustrate the *RMSE* variation during a simulation. As can be seen from the figures, the estimation performance of each algorithm may be affected by outliers; however, the proposed algorithm can effectively suppress this adverse affect. The overall *RMSE* of 30 Monte Carlo runs of simulation case 3 can be seen in Figure 8b,d. It can be seen that the *RMSE* of the proposed algorithm is evidently lower than those of the other algorithms, which further proves the superior robustness of the proposed algorithm. The statistical results of simulation case 3 are recorded in the Table 1. We can see that the position *ARMSE* of the proposed algorithm (VBGN-IMMCUKF) is improved by 74.48% compared to the EKF, 74.12% compared to the UKF, 74.11% compared to the IMCUKF, 73.81% compared to the IMMCUKF, and 43.37% compared to the GN-IMMCUKF. And the navigation accuracy of the proposed algorithm is improved by 92.42% compared to the EKF, 92.30% compared

to the UKF, 91.76% compared to the IMCUKF, 91.14% compared to the IMMCUKF, and 75.12% compared to the GN-IMMCUKF. The statistical results shows that the proposed VBGN-IMMCUKF can improve the robustness of the navigation system and has a superior performance in complex noise environments.
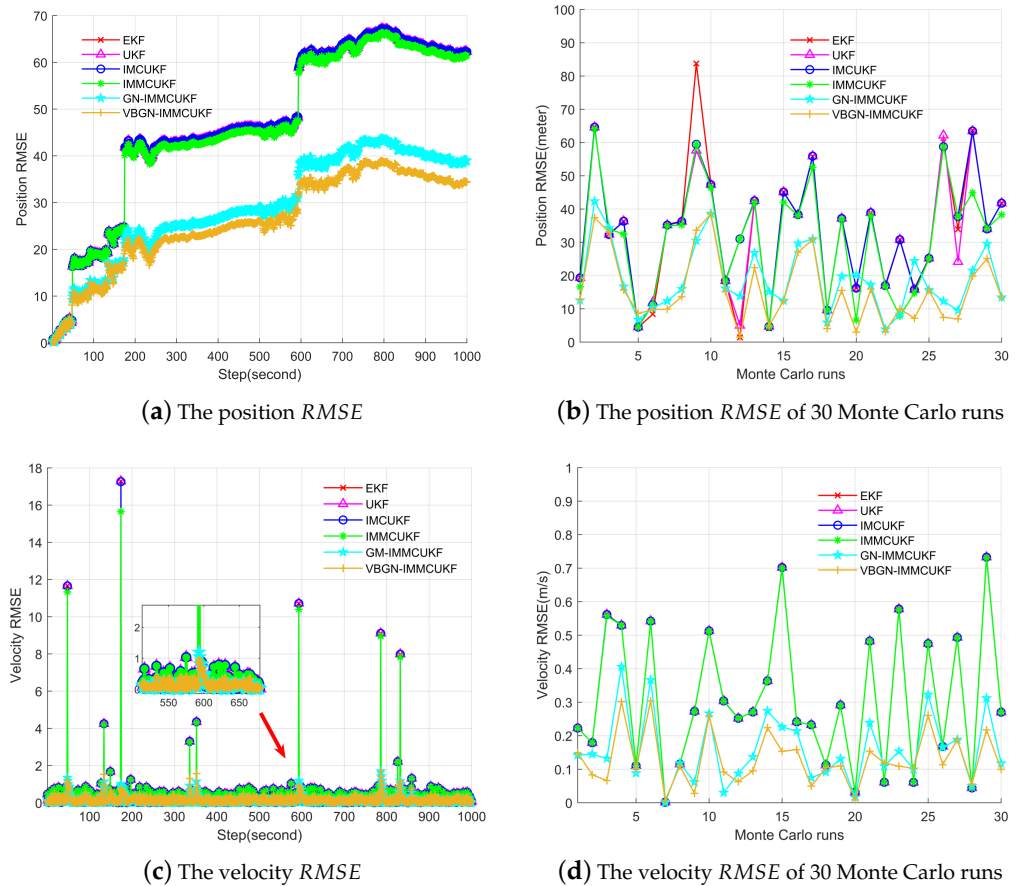

(**a**) The position *RMSE*

(**b**) The position *RMSE* of 30 Monte Carlo runs

(**c**) The velocity *RMSE*

(**d**) The velocity *RMSE* of 30 Monte Carlo runs

Figure 6. The experimental results of simulation case 1.

(**a**) The position *RMSE* of 30 Monte Carlo runs

(**b**) The velocity *RMSE* of 30 Monte Carlo runs
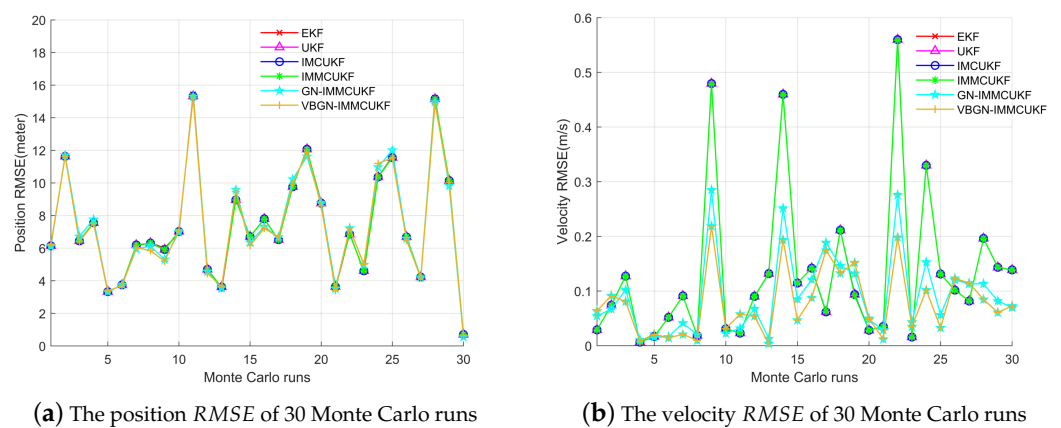
Figure 7. The experimental results of simulation case 2.

(**a**) The position *RMSE*

(**b**) The position *RMSE* of 30 Monte Carlo runs

(**c**) The velocity *RMSE*

(**d**) The velocity *RMSE* of 30 Monte Carlo runs
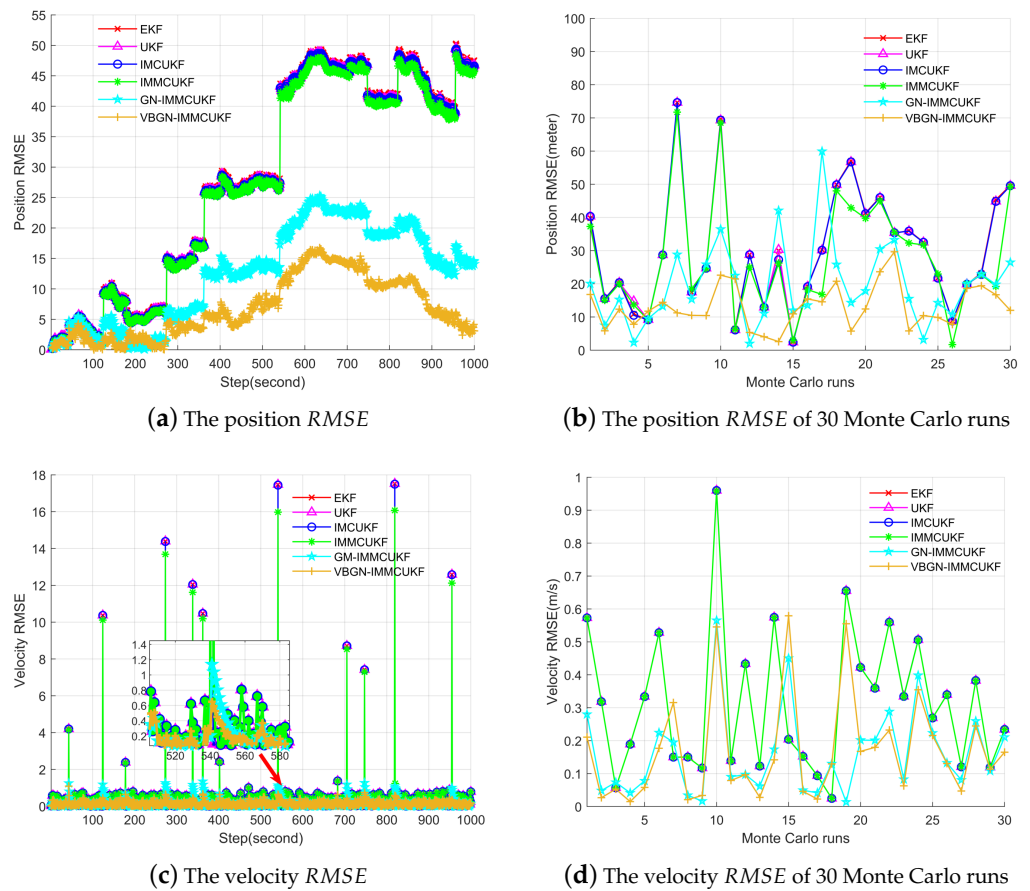
Figure 8. The experimental results of simulation case 3.

**Table 1.** Statistical comparison of simulation for different algorithms.

Dataset		EKF	UKF	IMCUKF	IMMCUKF	GN-IMMCUKF	VBGN-IMMCUKF
simulation case 1	$ARMSE_{pos}$ (m)	19.6300	19.6554	19.6335	19.5011	12.1014	10.8675
	$ARMSE_{vel}$ (m/s)	0.2743	0.2743	0.2643	0.2743	0.0734	0.0719
	Accuracy (%)	6.1983	5.9987	6.0985	5.9786	3.8817	3.5158
	Time (ms/step) (Outliers occur)	0.0029	0.0027	0.0042	0.0045	0.0541	0.0467
simulation case 2	$ARMSE_{pos}$ (m)	7.4309	7.4149	7.4049	7.4048	7.3878	7.3740
	$ARMSE_{vel}$ (m/s)	0.1338	0.1338	0.1337	0.1338	0.0926	0.0719
	Accuracy (%)	1.3546	1.3578	1.3478	1.3390	1.3358	1.3060
	Time (ms/step) (Outliers occur)	0.0023	0.0031	0.0048	0.0031	0.0496	0.0467
simulation case 3	$ARMSE_{pos}$ (m)	28.5191	28.1079	28.1083	27.7914	12.8405	7.2759
	$ARMSE_{vel}$ (m/s)	0.7853	0.7853	0.7850	0.7822	0.1466	0.1826
	Accuracy (%)	4.7532	4.6786	4.3787	4.0088	1.4478	0.3602
	Time (ms/step) (Outliers occur)	0.0038	0.0039	0.0054	0.0050	0.0574	0.0479

The simulation program was conducted using MATLAB R2021b on a PC equipped with a 12th Gen Intel(R) Core(TM) i7-12700H 2.30 GHz processor and 16.0 GB of RAM. Moreover, the computation time for one step when outliers occurred is recorded in Table 1. As can be seen in Table 1, although the proposed algorithm requires a longer processing time, it remains within the acceptable bounds for practical applications.

5. Sea Trial

Two groups of sea trial experiment data collected by a PX-210 AUV were utilized to compare the performance of the proposed algorithm. The PX-210 AUV was independently developed by the Underwater Vehicle Laboratory in Qingdao, Shandong, China.

As shown in Figure 9, the navigation system of the PX-210 AUV is equipped with navigation sensors, such as an INS, DVL, pressure sensor, etc., which can collect real-time navigation information during an AUV mission. The performances of the main navigation sensors on the PX-210 are given in Table 2. A Yanhua aimb-205 Industrial PC with a 12th Gen Intel(R) Core(TM) i5-6500 3.20 GHz processor and 16.0 GB of RAM was equipped on the PX-210 AUV. And communication with the sensors and computation in real time are achieved through a serial port and the MOOS-Ivp software 12.2 platform, which helps achieve the autonomous positioning and navigation of the AUV.
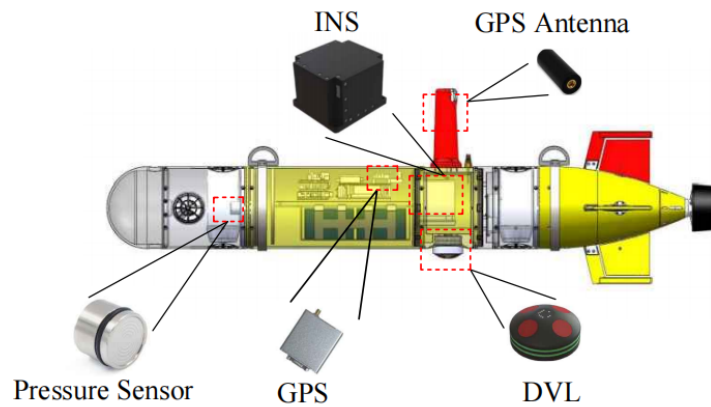
Figure 9. Main navigation sensors equipped on PX-210.

Table 2. Performances of the main navigation sensors on the PX-210.

Sensor Type		Performance	Update Rate
INS	Gyroscope	Constant ≤ 0.1 (°)/h Random ≤ 0.02 (°)/$\sqrt{h}$	50 Hz
	Accelerometer	Constant ≤ 0.1 (°)/h	
DVL		Bottom-Track Accuracy 0.4% ± 2 mm/s	1 Hz
GPS		Horizontal Position Accuracy 2.5 m	10 Hz
PS		Accuracy 0.09%	10 Hz

The sea trial was conducted in an open sea area of Tuandao Bay as illustrated in Figure 10. GPS position information was utilized as a reference value to evaluate and compare the performance of the proposed algorithms. The filter parameters during the sea trial were set the same as those introduced in Section 5 for the simulation case.

To verify the superior performance of the proposed algorithm, sea trial-1 was conducted. During sea trial-1, the AUV navigated for 844 s, covering a total distance of approximately 775 m. As depicted in Figure 11a, the AUV followed a box trajectory. Furthermore, Figure 11b illustrates that the DVL forward velocity exhibited outliers near 300 s.

To further compare the performance of the proposed algorithm, sea trial-2 was conducted. The trajectory and DVL measurements of sea trial-2 are illustrated in Figure 12. During sea trial-2, the AUV navigated a zigzag trajectory for 1090 s, covering a total distance of approximately 987 m. As shown in Figure 12b, the AUV experienced disturbances near 600 s and 1000 s.
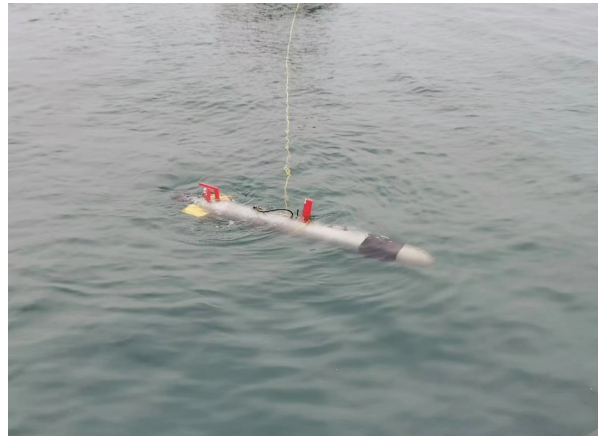
Figure 10. The PX-210 AUV during mission.

The position *RMSE* at each moment of sea trial-1 is depicted in Figure 13a. It is evident that outliers negatively impacted the performance of each algorithm, yet the proposed algorithm (VBGN-IMMCUKF) effectively mitigated this deterioration.

The statistical results are presented in Table 3. It is evident from the table that the end-point position *RMSE* of the proposed algorithm improved by 61.58% compared to the EKF, 61.58% compared to the UKF, 56.32% compared to the IMCUKF, 53.12% compared to the IMMCUKF, and 26.78% compared to the GN-IMCUKF. Additionally, the navigation accuracy of the proposed algorithm improved by 61.34% compared to the EKF, 61.33% compared to the UKF, 56.04% compared to the IMCUKF, 52.82% compared to the IMMCUKF, and 27.06% compared to the GN-IMCUKF. Therefore, based on the statistical results, we conclude that the proposed algorithm demonstrates superior robustness compared to other existing algorithms.

The position *RMSE* at each moment of sea trial-2 is depicted in Figure 13b. It is evident from the diagram that the unknown disturbances worsened the estimation performance of each algorithm. It is obvious that the proposed algorithm effectively mitigated this deterioration.

Table 3 presents the statistical results of sea trial-2. It is observed from the table that the end-point position *RMSE* of the proposed algorithm improved by 53.78% compared to the EKF, 51.71% compared to the UKF, 51.82% compared to the IMCUKF, 50.60% compared to the IMMCUKF, and 38.49% compared to the GN-IMCUKF. Additionally, the navigation accuracy of the proposed algorithm improved by 54.00% compared to the EKF, 53.99% compared to the UKF, 51.93% compared to the IMCUKF, 50.84% compared to the IMM-CUKF, and 38.60% compared to the GN-IMCUKF. Therefore, based on the statistical results, we conclude that the proposed algorithm outperforms other existing algorithms in AUV navigation under disturbance.
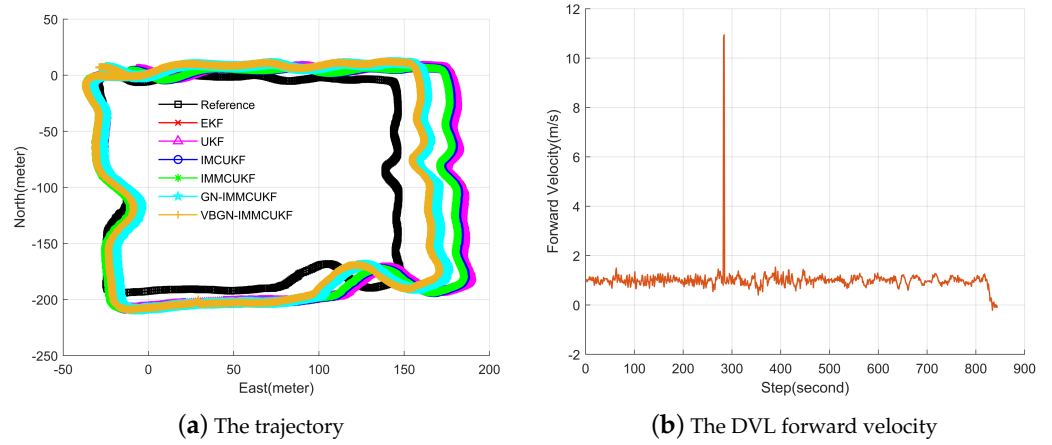
(**a**) The trajectory

(**b**) The DVL forward velocity

Figure 11. Trajectory and DVL forward velocity of sea trial-1.

(**a**) The trajectory

(**b**) The DVL forward velocity
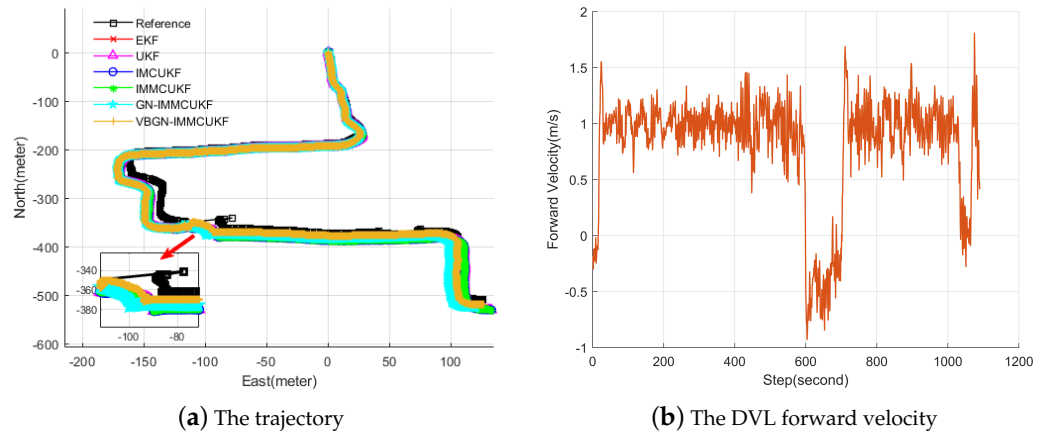
Figure 12. Trajectory and DVL forward velocity of sea trial-2.

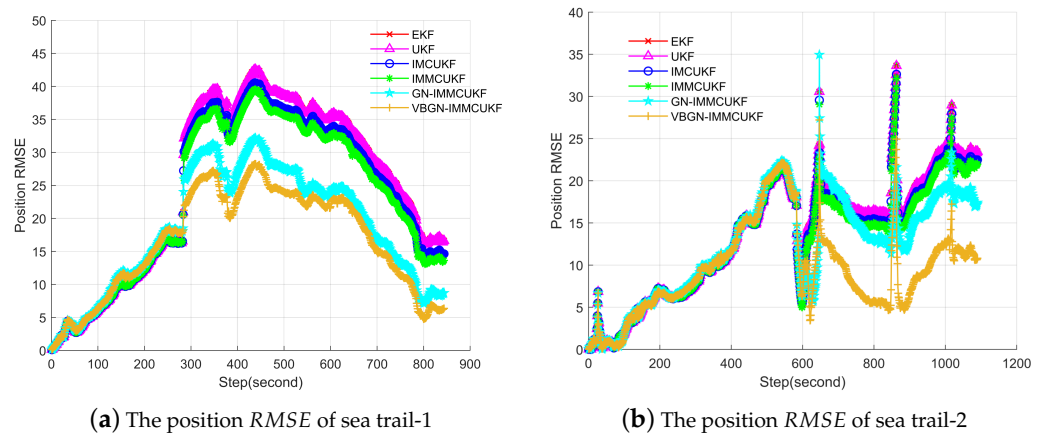(**a**) The position *RMSE* of sea trail-1

(**b**) The position *RMSE* of sea trail-2

Figure 13. The experimental results of sea trails.

Table 3. Statistical comparison of sea trails for different algorithms.

Dataset		EKF	UKF	IMCUKF	IMMCUKF	GN-IMMUKF	VB-GNIMMCUKF
sea trial-1	End position *RMSE* (m)	16.6468	16.6381	14.6413	13.6438	8.7352	6.3961
	Accuracy (%)	1.9257	1.9247	1.6937	1.5783	1.0208	0.7445
	Time (ms/step) (outlier occur)	0.5591	0.5809	0.4684	0.4540	1.2434	54.6370
sea trial-2	End position *RMSE* (m)	23.4197	22.4115	22.4705	21.9109	17.5975	10.8236
	Accuracy (%)	2.3575	2.3567	2.2560	2.2057	1.7662	1.0844
	Time (ms/step) (outlier occur)	0.0472	0.2029	0.2680	0.3533	1.1089	53.8124

Moreover, the computation times of one step when outliers occur are also listed in Table 3. It is worth mentioning that, although the proposed algorithm takes a lot of time when outliers occur, it can still meet the needs of 20 Hz AUV navigation in practice.

6. Conclusions

In this study, we introduced the Variational Bayesian and Gaussian-Newton method-based Iterated Maximum Mixture Correntropy Unscented Kalman Filter (VBGN-IMMCUKF), designed to enhance the navigation precision, fortify robustness against outliers, and increase adaptability to unknown disturbances within the realm of Autonomous Underwater Vehicle (AUV) integrated navigation systems. To assess the performance of the proposed VBGN-IMMCUKF, we conducted three sets of simulations alongside two series of sea trial experiments, benchmarking against established algorithms such as the Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), Iterated Maximum Correntropy Unscented Kalman Filter (IMCUKF), Iterated Maximum Mixture Correntropy Unscented Kalman Filter (IMMCUKF), and the Gaussian–Newton Iterated Maximum Mixture Correntropy Unscented Kalman Filter (GN-IMMCUKF). The experimental outcomes affirm that our proposed algorithm outperforms existing algorithms in terms of robustness and adaptability.

However, it is noteworthy that despite the practical applicability of our algorithm, it requires significantly more computational time than conventional alternatives. Consequently, future work should prioritize enhancing the computational efficiency of the VBGN-IMMCUKF to accommodate the demands of high-frequency navigation applications. Moreover, the kernel function form within the correntropy deserves further exploration to refine the MCC-based algorithm for handling even more complex environments.

References

1. Zhou, J.; Si, Y.; Chen, Y. A Review of Subsea AUV Technology. *J. Mar. Sci. Eng.* **2023**, *11*, 1119. [CrossRef]
2. Zhang, B.; Ji, D.; Liu, S.; Zhu, X.; Xu, W. Autonomous Underwater Vehicle Navigation: A Review. *Ocean. Eng.* **2023**, *273*, 113861. [CrossRef]
3. Mohsan, S.A.H.; Li, Y.; Sadiq, M.; Liang, J.; Khan, M.A. Recent Advances, Future Trends, Applications and Challenges of Internet of Underwater Things (IoUT): A Comprehensive Review. *J. Mar. Sci. Eng.* **2023**, *11*, 124. [CrossRef]
4. Wang, S.; Li, W.; Xing, L. A Review on Marine Economics and Management: How to Exploit the Ocean Well. *Water* **2022**, *14*, 2626. [CrossRef]
5. Li, Z.; Wang, Y.; Yang, W.; Ji, Y. Development Status and Key Navigation Technology Analysis of Autonomous Underwater Vehicles. In Proceedings of the 2020 3rd International Conference on Unmanned Systems (ICUS), Harbin, China, 27 November 2020; pp. 1130–1133. [CrossRef]
6. Krauss, S.T.; Stilwell, D.J. Unscented Kalman Filtering on Manifolds for AUV Navigation—Experimental Results. In Proceedings of the OCEANS 2022, Hampton Roads, VI, USA, 17–20 October 2022; pp. 1–6. [CrossRef]
7. Allotta, B.; Caiti, A.; Chisci, L.; Costanzi, R.; Di Corato, F.; Fantacci, C.; Fenucci, D.; Meli, E.; Ridolfi, A. An unscented Kalman filter based navigation algorithm for autonomous underwater vehicles. *Mechatronics* **2016**, *39*, 185–195. [CrossRef]
8. Chang, G.; Liu, M. M-Estimator-Based Robust Kalman Filter for Systems with Process Modeling Errors and Rank Deficient Measurement Models. *Nonlinear Dyn.* **2015**, *80*, 1431–1449. [CrossRef]
9. Crespillo, O.G.; Medina, D.; Skaloud, J.; Meurer, M. Tightly Coupled GNSS/INS Integration Based on Robust M-estimators. In Proceedings of the 2018 IEEE/ION Position, Location and Navigation Symposium (PLANS), Monterey, CA, USA, 23–26 April 2018; pp. 1554–1561. [CrossRef]
10. Wang, D.; Wang, B.; Huang, H.; Zhang, H. A SINS/DVL Navigation Method Based on Hierarchical Water Velocity Estimation*. *Meas. Sci. Technol.* **2024**, *35*, 015116. [CrossRef]
11. Jia, G.; Zhang, Y.; Huang, Y.; Bai, M.; Wang, G. A New Robust Student's t Based SINS/GPS Integrated Navigation Method. In Proceedings of the 2018 International Conference on Control, Automation and Information Sciences (ICCAIS), Hangzhou, China, 24–27 October 2018; pp. 285–290. [CrossRef]
12. Wang, J.; Zhang, T.; Jin, B.; Zhu, Y.; Tong, J. Student's t-Based Robust Kalman Filter for a SINS/USBL Integration Navigation Strategy. *IEEE Sens. J.* **2020**, *20*, 5540–5553. [CrossRef]
13. Chen, B.; Liu, X.; Zhao, H.; Principe, J.C. Maximum Correntropy Kalman Filter. *Automatica* **2017**, *76*, 70–77. [CrossRef]
14. Chen, B.; Wang, J.; Zhao, H.; Zheng, N.; Principe, J.C. Convergence of a Fixed-Point Algorithm under Maximum Correntropy Criterion. *IEEE Signal Process. Lett.* **2015**, *22*, 1723–1727. [CrossRef]
15. Wang, G.; Li, N.; Zhang, Y. Maximum Correntropy Unscented Kalman and Information Filters for Non-Gaussian Measurement Noise. *J. Frankl. Inst.* **2017**, *354*, 8659–8677. [CrossRef]
16. Wang, G.; Zhang, Y.; Wang, X. Iterated Maximum Correntropy Unscented Kalman Filters for Non-Gaussian Systems. *Signal Process.* **2019**, *163*, 87–94. [CrossRef]
17. Huang, F.; Zhang, J.; Zhang, S. Adaptive Filtering Under a Variable Kernel Width Maximum Correntropy Criterion. *IEEE Trans. Circuits Syst. II Express Briefs* **2017**, *64*, 1247–1251. [CrossRef]
18. Shi, L.; Zhao, H.; Zakharov, Y. An Improved Variable Kernel Width for Maximum Correntropy Criterion Algorithm. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *67*, 1339–1343. [CrossRef]
19. Liu, B.; Zhang, X.; Jia, S.; Zou, S.; Tian, D. Variational Bayesian-Based Adaptive Maximum Correntropy Generalized High-Degree Cubature Kalman Filter. *Circuits Syst. Signal Process.* **2023**, *42*, 7073–7098. [CrossRef]
20. Li, X.; Guo, Y.; Meng, Q. Variational Bayesian-Based Improved Maximum Mixture Correntropy Kalman Filter for Non-Gaussian Noise. *Entropy* **2022**, *24*, 117. [CrossRef] [PubMed]
21. Li, Z.; Chen, W.; Sun, Y.; Chen, Z. An Improved Multiple-Outlier Robust Filter Based on Maximum Correntropy Criterion for Integrated Navigation. *IEEE Sens. J.* **2023**, *23*, 17451–17461. [CrossRef]
22. Pagoti, S.K.; Vemuri, S.I.D. Development and Performance Evaluation of Correntropy Kalman Filter for Improved Accuracy of GPS Position Estimation. *Int. J. Intell. Netw.* **2022**, *3*, 1–8. [CrossRef]
23. Wang, B.; Wang, Z. Adaptive Robust Kalman Filter Based on MCC and Its Application in Underwater Integrated Navigation. In *China Satellite Navigation Conference Proceedings, CSNC 2022, VOL III*; Yang, C., Xie, J., Eds.; Lecture Notes in Electrical Engineering; Springer: Singapore, 2022; Volume 910, pp. 483–492. [CrossRef]
24. Zhang, X.; He, B.; Gao, S.; Mu, P.; Xu, J.; Zhai, N. Multiple Model AUV Navigation Methodology with adaptability and Robustness. *Ocean. Eng.* **2022**, *254*, 111258. [CrossRef]
25. Ribas, D.; Ridao, P.; Tardos, J.D.; Neira, J. Underwater SLAM in Man-Made Structured Environments. *J. Field Robot.* **2008**, *25*, 898–921. [CrossRef]
26. Zhao, H.; Tian, B.; Chen, B. Robust Stable Iterated Unscented Kalman Filter Based on Maximum Correntropy Criterion. *Automatica* **2022**, *142*, 110410. [CrossRef]

27. Battistelli, G.; Chisci, L.; Fantacci, C. Parallel Consensus on Likelihoods and Priors for Networked Nonlinear Filtering. *IEEE Signal Process. Lett.* **2014**, *21*, 787–791. [CrossRef]
28. Liu, F.; Shen, Y.; He, B.; Wang, D.; Wan, J.; Sha, Q.; Qin, P. Drift Angle Compensation-Based Adaptive Line-of-Sight Path Following for Autonomous Underwater Vehicle. *Appl. Ocean. Res.* **2019**, *93*, 101943. [CrossRef]