

Article

An Improved Inertial Frame Alignment Algorithm Based on Horizontal Alignment Information for Marine SINS

Yanting Che^{1,*}, Qiuying Wang^{2,†}, Wei Gao^{1,*} and Fei Yu^{1,†}

- ¹ College of Automation, Harbin Engineering University, Harbin, China; E-Mail: yufei@hrbeu.edu.cn
- ² College of Information and Communication Engineering, Harbin Engineering University, Harbin, China; E-Mail: wqy869087@163.com
- [†] These authors contributed equally to this work.
- * Authors to whom correspondence should be addressed; E-Mails: cheyanting@hrbeu.edu.cn (Y.C.); gaow@hrbeu.edu.cn (W.G.); Tel.: +86-451-8258-9395 (Y.C. & W.G.).

Academic Editor: Vittorio M. N. Passaro

Received: 13 May 2015 / Accepted: 22 September 2015 / Published: 5 October 2015

Abstract: In this paper, an improved inertial frame alignment algorithm for a marine SINS under mooring conditions is proposed, which significantly improves accuracy. Since the horizontal alignment is easy to complete, and a characteristic of gravity is that its component in the horizontal plane is zero, we use a clever method to improve the conventional inertial alignment algorithm. Firstly, a large misalignment angle model and a dimensionality reduction Gauss-Hermite filter are employed to establish the fine horizontal reference frame. Based on this, the projection of the gravity in the body inertial coordinate frame can be calculated easily. Then, the initial alignment algorithm is accomplished through an inertial frame alignment algorithm. The simulation and experiment results show that the improved initial alignment algorithm performs better than the conventional inertial alignment algorithm, and meets the accuracy requirements of a medium-accuracy marine SINS.

Keywords: SINS; alignment; dimension reduction GHF; inertial coordinate frame

1. Introduction

A strapdown inertial navigation system (SINS) is a dead-reckoning navigation system, and the initial alignment is an essential procedure for a SINS, since it directly affects the precision of navigation parameters (position, velocity, and attitude) [1,2]. The main purpose of the initial alignment is to determine the initial strapdown attitude matrix between the body frame and navigation frame, and its accuracy is especially important for a marine SINS, which usually has to work for a long time [1–4].

Generally, the alignment process can be divided into two phases, the coarse-alignment phase and the fine-alignment phase [4–8]. The coarse-alignment phase is required to estimate the ship's heading within a few degrees and pitch/roll within a few tenths of a degree in order to allow the fine-alignment filter to operate within its linear region [5,6]. The typical coarse-alignment method is analytic coarse-alignment. However, it is unable to handle the in-motion alignment problem [3–5,7]. In order to overcome the difficulties when a marine SINS is under mooring conditions, many methods have been developed and analyzed. Reference [3] proposed an improved alignment based on gravity in an inertial frame, and velocity is used in the calculation to reduce the influence of disturbance acceleration. In mooring conditions, due to the presence of the disturbed acceleration and angular velocities, accurate gravity and Earth rate are difficult to obtain directly, which finally leads to the low precision of the coarse alignment. Since the signal-to-noise ratios of gyros' and accelerometers' output are poor and the frequency bands of disturbed signals are wide, it is unable to separate the pure, useful signals from the interference signals measured by gyros and accelerometers [6,7,9]. In order to remove the high frequency noise, reference [7] used an IIR digital low-pass filter to process the gyro and accelerometer measurements. Although there have been various methods presented so as to obtain the purer Earth rate and gravity signals, the precision is still not high enough.

While in the fine-alignment phase, usually the standard Kalman filter or the compass loop method can be implemented based on the coarse-alignment result, and under the assumption of a small misalignment angle linear error model [1,2]. However, with the increase of SINS application technology and the development of nonlinear filtering estimation technology, its error models are no longer confined to linear models, and new error models are constantly emerging. Therefore, nonlinear filters are used for alignment [10–16]. However, no matter what method is employed, the heading misalignment angle will usually converge over 10 min under mooring conditions, which is slower than the horizontal misalignment angle (within 2 min only) and does not meet the demand for a quick start.

It is well known that the horizontal components of gravity projection in the horizontal coordinate frame are zero and the horizontal alignment is rapid. If an accurate horizontal coordinate frame is established, the interference caused by sway can be easily isolated. Based on this idea, we proposed an improved alignment scheme in [9]. Then, according to the characteristics of system structure, we used a dimensionality reduction Gauss-Hermite filter (GHF) algorithm to establish the accurate horizontal coordinate frame [17]. We specify the details of the algorithm and supplement error analysis in this paper. Because of the low precision of the traditional inertial frame alignment algorithm in the mooring environment, it is usually employed as a coarse alignment for a marine SINS. We use a clever method to improve the traditional inertial alignment algorithm.

Compared with the commonly used nonlinear filtering algorithms such as EKF [18,19], UKF [20] and CKF [21,22], GHQF [23] has the advantages of incomparable precision and stability. However, this approach is infeasible for high-dimensional systems since the computation burden increases exponentially with the index of dimension. This results in the "curse of dimensionality". Fortunately, only misalignment angles suffer from nonlinearity in initial alignment, so that we can apply a dimension reduction nonlinear filter to carry out the alignment.

The remainder of this paper is organized as follows. Firstly, reference frames and parameter definitions are addressed in Section 2. The algorithmic principle for traditional alignment in the inertial frame is presented in Section 3. Section 4 details how to establish an accurate horizontal reference frame and Section 5 details how to accomplish the alignment. Then, the simulation results that validate the proposed approach are presented in Section 6. Section 7 presents the experimental results. Finally, the conclusions are presented in Section 8.

2. Reference Frames and Parameter Definitions

The reference frames are defined in Table 1 and the parameters are defined in Table 2.

Reference Frame	Definition			
. from a	Navigation reference frame which is the local horizontal reference frame. Its axes are aligned with			
<i>n</i> -frame	east-north-up (ENU) geodetic axes.			
<i>h</i> -frame	Horizontal reference frame. Its z_h axis is aligned with z_n axis, but the horizontal axes are			
	arbitrary in the horizontal plane.			
<i>e</i> -frame	Earth-centered Earth-fixed orthogonal reference frame. Its xe axis points to the local longitude.			
<i>b</i> -frame	Body reference frame aligned with inertial measurement unit (IMU) axes.			
: from a	Earth-centered inertial fixed (ECIF) orthogonal reference frame. The axes are fixed with e-frame			
<i>i</i> -frame	at the beginning of the alignment process.			
i from a	Body inertial reference frame. It is formed by fixing the axes of b-frame in the inertial space at the			
l_{b0} -frame	beginning of the alignment process.			

Table 1. The reference frames definitions used in this paper.

The relationship of the frames mentioned in Table 1 is shown in Figure 1.



Figure 1. The relationship of the frames.

Parameter	Definition					
C_{p1}^{p2}	Transform matrix from $p1$ frame to $p2$ frame					
ω _{ie}	Angular rate of Earth rotation					
g	Gravity					
L	Latitude					
V	Velocity					
3	Gyro drift					
∇	Acceleration bias					
ϕ	Misalignment angle					
f	Specific Force					

Table 2. The parameters.

3. The Algorithmic Scheme for an Alignment Algorithm in an Inertial Frame

The alignment algorithm in an inertial frame is based on the consideration that the Earth rate is constant in a body inertial reference frame, and we can get north from the projection of the gravity in the inertial reference frame which defines a cone whose main axis is the rotational axis of the Earth [3,7].

The traditional alignment algorithm in an inertial frame is presented in [3–5,7]. It is usually decomposing the strapdown matrix $C_b^n(t)$ (which represents the orientation of the *b* frame relative to the *n* frame) as per Equation (1):

$$C_b^n(t) = C_e^n C_i^e(t) C_{i_{b0}}^i C_b^{i_{b0}}(t)$$
⁽¹⁾

Under mooring conditions, C_e^n is a function of latitude L, and $C_i^e(t)$ is a function of time t.

$$C_{e}^{n} = \begin{bmatrix} 0 & 1 & 0 \\ -\sin L & 0 & \cos L \\ \cos L & 0 & \sin L \end{bmatrix}$$
(2)

$$C_i^e(t) = \begin{bmatrix} \cos(\omega_{ie}t) & \sin(\omega_{ie}t) & 0\\ -\sin(\omega_{ie}t) & \cos(\omega_{ie}t) & 0\\ 0 & 0 & 1 \end{bmatrix}$$
(3)

where $C_b^{i_{b0}}(t)$ can be updated by the gyro output (its initial value is a unit matrix):

$$\dot{C}_{b}^{i_{b0}}\left(t\right) = C_{b}^{i_{b0}}\left(t\right) \left[\left(\boldsymbol{\omega}_{i_{b0}}^{b}\right) \times \right]$$

$$\tag{4}$$

where $\left[\left(\boldsymbol{\omega}_{i_{b0}}^{b}\right)\times\right]$ is the skew symmetric matrix of the vector $\boldsymbol{\omega}_{i_{b0}}^{b}$ measured by the gyroscopes representing the angular rate of *b* -frame with respect to i_{b0} -frame, and $C_{i_{b0}}^{i}$ is calculated as:

$$C_{i_{b0}}^{i} = \begin{bmatrix} \left[g^{i}(t_{1}) \right]^{T} \\ \left[g^{i}(t_{2}) \right]^{T} \\ \left[g^{i}(t_{1}) \times g^{i}(t_{2}) \right]^{T} \end{bmatrix}^{-1} \begin{bmatrix} \left[g^{i_{b0}}(t_{1}) \right]^{T} \\ \left[g^{i_{b0}}(t_{2}) \right]^{T} \\ \left[g^{i_{b0}}(t_{1}) \times g^{i_{b0}}(t_{2}) \right]^{T} \end{bmatrix}$$
(5)

where, g^i and $g^{i_{b0}}$ represent the projections of the gravity in the *i* frame and the i_{b0} frame, respectively. Since g^n is known, the projection in the inertial frame can be calculated as:

$$g^i = C_e^i(t)C_n^e g^n \tag{6}$$

In order to restrain the interference of disturbing acceleration, we usually use the following equation instead of Equation (5):

$$C_{i_{b0}}^{i} = \begin{bmatrix} \begin{bmatrix} V^{i}(t_{1}) \end{bmatrix}^{T} \\ \begin{bmatrix} V^{i}(t_{2}) \end{bmatrix}^{T} \\ \begin{bmatrix} V^{i}(t_{2}) \end{bmatrix}^{T} \\ \begin{bmatrix} V^{i_{b0}}(t_{2}) \end{bmatrix}^{T} \end{bmatrix}^{-1} \begin{bmatrix} \begin{bmatrix} V^{i_{b0}}(t_{1}) \end{bmatrix}^{T} \\ \begin{bmatrix} V^{i_{b0}}(t_{2}) \end{bmatrix}^{T} \\ \begin{bmatrix} V^{i_{b0}}(t_{1}) \times V^{i_{b0}}(t_{2}) \end{bmatrix}^{T} \end{bmatrix}$$
(7)

where $V^{i}(t_{j}) = \int g^{i}(t_{j})dt$, $V^{i_{b0}}(t_{j}) = \int g^{i_{b0}}(t_{j})dt = \int \hat{C}^{i_{b0}}_{b}f^{b}(t_{j})dt$, (j = 1, 2).

Taking into account that C_e^n , $C_i^e(t)$, $C_b^{i_{b0}}(t)$, and g^i can all be calculated by the parameters known or measured, how to extract the pure gravity projection in the i_{b0} frame from the output of accelerometers is the essential operation we have to carry out.

Figure 2 illustrates the algorithmic scheme.



Figure 2. Flow chart of alignment algorithm in an inertial frame.

4. The Establishment of a Horizontal Reference Frame

In Appendix A, we analyze the propagation of errors, and find it has the same precision as other methods. Therefore, if we can isolate the interference errors, the inertial frame alignment algorithm can be used for fine alignment.

In Section 3, we have pointed out that the essential and difficult process is to extract the pure gravity from measuring signals which are often interfered with by environmental disturbance. In order to obtain

the accurate gravity projection in the i_{b0} frame, we proposed a clever solution by using a feature of gravity. It is easy to obtain the pure gravity once an accurate horizontal reference frame is established, since the projection of gravity in the *h*-frame has nothing to do with the heading, and it is also relatively easy to establish the *h*-frame. In this section, we will show how to establish an accurate horizontal reference frame.

4.1. Nonlinear Error Model of a SINS

Traditional linear differential equations are based on the assumption that the misalignment angles are small. However, for a small misalignment angle model, a coarse alignment is necessary. To improve the accuracy and reduce the time, the nonlinear error model of large misalignment angle for a SINS described in [24] is adopted in this paper (see Section 2.1 in [24]).

Attitude error equation:

$$\dot{\phi} = C_{\omega}^{-1} \left[\left(I - C_{n}^{n'} \right) \hat{\omega}_{in}^{n} + C_{n}^{n'} \delta \omega_{in}^{n} - C_{b}^{n'} \delta \omega_{ib}^{b} \right]$$
(8)

Velocity error equation:

$$\delta \dot{v}^{n} = \left[I - \left(C_{n}^{n'} \right)^{T} \right] C_{b}^{n'} \hat{f}^{b} + C_{b}^{n} \nabla^{b} - \left(2\delta \omega_{ie}^{n} + \delta \omega_{en}^{n} \right) \times \left(\dot{v}^{n} - \delta v^{n} \right) - \left(2\hat{\omega}_{ie}^{n} + \hat{\omega}_{en}^{n} \right) \times \delta v^{n} + C_{b}^{n} w_{a}^{b} \tag{9}$$

Position error equations:

$$\begin{cases} \delta \dot{L} = \frac{\delta v_N^n}{R_M} \\ \delta \dot{\lambda} = \frac{\sec L}{R_N} \delta v_E^n + \frac{v_E^n \sec L \tan L}{R_N} \delta L \end{cases}$$
(10)

where superscript n' donates the calculation navigation reference frame, and $\phi = \begin{bmatrix} \phi_x & \phi_y & \phi_z \end{bmatrix}^T$ is the Euler error angle vector. $c(\phi_i)$ and $s(\phi_i)$ denote $\cos \phi_i$ and $\sin \phi_i$, respectively.

$$C_{n}^{n'} = \begin{vmatrix} c(\phi_{y})c(\phi_{z}) - s(\phi_{y})s(\phi_{x})s(\phi_{z}) & c(\phi_{y})s(\phi_{z}) + s(\phi_{y})s(\phi_{x})c(\phi_{z}) & -s(\phi_{y})c(\phi_{x}) \\ -c(\phi_{x})s(\phi_{z}) & c(\phi_{x})c(\phi_{z}) & s(\phi_{x}) \\ s(\phi_{y})c(\phi_{z}) + c(\phi_{x})s(\phi_{z}) & s(\phi_{y})s(\phi_{z}) - c(\phi_{y})s(\phi_{x})c(\phi_{z}) & c(\phi_{x})c(\phi_{y}) \end{vmatrix}$$
(11)

$$C_{\omega} = \begin{bmatrix} c(\phi_{y}) & 0 & -s(\phi_{y})c(\phi_{x}) \\ 0 & 1 & s(\phi_{x}) \\ s(\phi_{y}) & 0 & c(\phi_{y})c(\phi_{x}) \end{bmatrix}$$
(12)

$$\hat{\omega}_{in}^n = \hat{\omega}_{ie}^n + \hat{\omega}_{en}^n \tag{13}$$

$$\delta\omega_{in}^{n} = \delta\omega_{ie}^{n} + \delta\omega_{en}^{n} \tag{14}$$

$$\delta \omega_{ib}^b = \varepsilon^b + w_g^b \tag{15}$$

where $\hat{\omega}_{ie}^{n}$ is the calculated Earth's rotating angular rate, $\hat{\omega}_{en}^{n}$ is the calculated angular rate vector, and $\delta \omega_{in}^{n}$ is the calculated error vector of ω_{in}^{n} . $\delta \omega_{ie}^{n}$ and $\delta \omega_{en}^{n}$ are respectively the error vectors of $\hat{\omega}_{ie}^{n}$ and $\hat{\omega}_{en}^{n} \varepsilon^{b}$ and W_{g}^{b} are the gyro constant drift vector and the zero-mean Gaussian white noise vector, respectively. \hat{f}^{b} and δf^{b} denote the specific force vector and its corresponding error vector, respectively. \hat{v}^{n} and δv^{n} are

calculated velocity vector and its corresponding error vector, respectively. R_M and R_N are the Earth's radii of the meridian circle and the prime vertical circle, respectively.

4.2. The Dimension Reduction Gauss-Hermite Filter

The Gauss-Hermite filter (GHF) is one of the sigma point filters. It has proved to be efficient and successful in solving estimation problems when the state and noise distributions are Gaussian. It is usually used as a benchmark algorithm, since its accuracy and stability are the highest among numerous Gaussian approximation filters [23,25,26] (the algorithm framework see Appendix B). However, the "curse of dimensionality" would seriously affect the real-time performance for high dimensional systems [23,27]. From Section 4.1, it is known that only misalignment angle suffers from nonlinearity in the nonlinear error model. This means it is possible to employ a dimension reduction GHF to deal with the alignment task.

In order to establish an accurate horizontal reference frame and reduce the amount of calculation, we employ the dimension reduction GHF algorithm.

The large misalignment angle error model of SINS alignment is a typical nonlinear model that can be described as a general form:

$$x_{k+1} = F_k(\xi_k) x_k + g_k(\xi_k) + w_k$$
(16)

$$y_k = H_k(\varsigma_k)x_k + h_k(\varsigma_k) + v_k \tag{17}$$

where ξ_k is the first *l* components of x_k , ζ_k is arbitrary *s* components of x_k .

The dimension reduction GHF algorithm is shown in Figure 3.

$$\begin{array}{c}
 \text{Initialized state variable and covariance matrix} \\
 \hat{x}_{0} = E[x_{0}], \xi_{0} \\
 P_{0} = E[(x_{0} - \hat{x}_{0})(x_{0} - \hat{x}_{0})^{T}], P_{0}^{\xi} \\
\end{array}$$

$$\begin{array}{c}
 \text{Predicttion 1.Calculate sigma points } \chi_{i,k-1}^{\xi} & \chi_{i,k-1}^{\xi} \\
 2. \quad \hat{x}_{k/k-1} = \sum_{l_{1}}^{m} \cdots \sum_{l_{l}}^{m} \Phi_{k-1}(\xi_{l_{1},k-1}, \xi_{l_{2},k-1}, \cdots, \xi_{l_{l},k-1})\omega_{l_{1}}\omega_{l_{2}} \cdots \omega_{l} \\
 3. \quad P_{k/k-1} = \sum_{l_{1}}^{m} \cdots \sum_{l_{l}}^{m} \left(\Psi_{k-1}(\xi_{l_{1},k-1}, \xi_{l_{2},k-1}, \cdots, \xi_{l_{l},k-1})\omega_{l_{0}}\omega_{l_{2}} \cdots \omega_{l}\right) \\
 4. \quad \hat{y}_{k/k-1} = \sum_{l_{1}}^{m} \cdots \sum_{l_{l}}^{m} \left(\omega_{l_{1}} \cdots \omega_{l_{l}} \Pi_{k}\left(\gamma_{l,k-1}^{\xi}\right)\right) \\
 5. \quad P_{k/k-1}^{y} = \sum_{l_{1}}^{m} \cdots \sum_{l_{l}}^{m} \left(\omega_{l_{1}} \cdots \omega_{l_{l}} \Pi_{k}\left(\gamma_{l,k-1}^{\xi}\right)\right) \\
 6. \quad P_{k/k-1}^{xy} = \sum_{l_{1}}^{m} \cdots \sum_{l_{l}}^{m} \left(\omega_{l_{1}} \cdots \omega_{l_{l}} \Pi_{k}\left(\gamma_{l,k-1}^{\xi}\right)\right) \\
 (F_{k/k-1}^{xy} = \sum_{l_{1}}^{m} \cdots \sum_{l_{l}}^{m} \left(\omega_{l_{1}} \cdots \omega_{l_{l}} \Pi_{k}\left(\gamma_{l,k-1}^{\xi}\right)\right) \\
 (F_{k/k-1}^{xy} = \sum_{l_{1}}^{m} \cdots \sum_{l_{l}}^{m} \left(\omega_{l_{1}} \cdots \omega_{l_{l}} \Pi_{k}\left(\gamma_{l,k-1}^{\xi}\right)\right) \\
 (F_{k/k-1}^{yy} = \sum_{l_{1}}^{m} \cdots \sum_{l_{k}}^{m} \left(\omega_{l_{1}} \cdots \omega_{l_{k}} \Pi_{k}\left(\gamma_{l,k-1}^{\xi}\right)\right) \\
 (F_{k/k-1}^{yy} = \sum_{l_{k}}^{m} \cdots \sum_{l_{k}}^{m} \left(\omega_{l_{1}} \cdots \omega_{l_{k}} \Pi_{k}\left(\gamma_{l,k-1}^{\xi}\right)\right) \\
 (F_{k/k-1}^{yy} = \sum_{l_{k}}^{m} \cdots \sum_{l_{k}}^{m} \left(\omega_{l_{1}} \cdots \omega_{l_{k}} \Pi_{k}\left(\gamma_{l,k-1}^{\xi}\right)\right) \\
 (F_{k/k-1}^{yy} = \sum_{l_{k}}^{m} \cdots \sum_{l_{k}}^{m} \left(\omega_{l_{k}} \cdots \omega_{l_{k}} \Pi_{k}\left(\gamma_{l,k-1}^{\xi}\right)\right) \\
 (F_{k/k-1}^{yy} = \sum_{l_{k}}^{m} \cdots \sum_{l_{k}}^{m} \left(\omega_{l_{k}} \cdots \omega_{l_{k}} \Pi_{k}\left(\gamma_{l,k-1}^{\xi}\right)\right) \\
 (F_{k/k-1}^{yy} = \sum_{l_{k}}^{m} \cdots \sum_{l_{k}}^{m} \left(\omega_{l_{k}} \cdots \omega_{l_{k}} \Pi_{k}\left(\gamma_{l,k-1}^{\xi}\right)\right) \\
 (F_{k/k-1}^{yy} = \sum_{l_{k}}^{yy} = \sum_{l_{k}}^{yy} \cdots \sum_{l_{k}}^{yy} = \sum_{l_{k}^$$

Figure 3. The flow chart of the dimension reduction GHF.

where,

$$\Phi_{k-1}(\xi_{k-1}) = F_{k-1}(\xi_{k-1}) \left(\hat{x}_{k-1} + S_{k-1} \begin{bmatrix} (S_{k-1}^{\xi})^{-1}(\xi_{k-1} - \hat{\xi}_{k-1}) \\ 0 \end{bmatrix} \right) + g_{k-1}(\xi_{k-1})$$
(18)

$$\Psi_{k-1}(\xi_{k-1}) = \Phi_{k-1}(\xi_{k-1})\Phi_{k-1}^{T}(\xi_{k-1}) + F_{k-1}(\xi_{k-1})S_{k-1}\begin{bmatrix} 0 & 0\\ 0 & I_{n-l} \end{bmatrix}S_{k-1}^{T}F_{k-1}^{T}(\xi_{k-1})$$
(19)

$$\Theta_{k}(\varsigma_{k}) = \hat{x}_{k/k-1}' + S_{k/k-1} \begin{bmatrix} (S_{k/k-1}^{\varsigma})^{-1}(\varsigma_{k} - \hat{\varsigma}_{k/k-1}) \\ 0 \end{bmatrix}$$
(20)

$$\Omega_k(\varsigma_k) = \Theta_k(\varsigma_k)\Theta_k^T(\varsigma_k) + S_{k/k-1} \begin{bmatrix} 0 & 0 \\ 0 & I_t \end{bmatrix} S_{k/k-1}^T$$
(21)

$$\mathbf{M}_{k}(\boldsymbol{\varsigma}_{k}) = H_{k}'(\boldsymbol{\varsigma}_{k})\boldsymbol{\Theta}_{k}(\boldsymbol{\varsigma}_{k}) + h_{k}(\boldsymbol{\varsigma}_{k})$$
(22)

$$\Pi_{k}(\varsigma_{k}) = \mathbf{M}_{k}^{T}(\varsigma_{k})\mathbf{M}_{k}^{T}(\varsigma_{k}) + H_{k}'(\varsigma_{k})S_{k/k-1}'\begin{bmatrix}0 & 0\\0 & I_{t}\end{bmatrix}(S_{k/k-1}')^{T}F^{T}(\varsigma_{k})$$
(23)

where, P^{ξ} denotes the first l-th rows and the first l-th columns of the matrix P; S^{ξ} and S are obtained from P^{ξ} and P, respectively, through a Cholesky decomposition; that is $P = SS^{T}$, $P^{\xi} = S^{\xi}(S^{\xi})^{T}$. $\hat{\xi}$ denotes the first l-th components of \hat{x} , and m = n - l. The quadrature points $\{\xi_i\}$ and the associated weights $\{\omega_i\}$ are determined by the Gauss–Hermite quadrature rule (see Appendix C).

4.3. The Horizontal Alignment for the Large Misalignment Angle Model Based on the Dimension Reduction Gauss-Hermite Filter

As mentioned above, the nonlinear model for horizontal alignment is established under the large misalignment angle in this paper. Considering 13 state variables—the east velocity error δv_E and the north velocity error δv_N ; the Euler misalignment angle errors ϕ_x , ϕ_y and ϕ_z ; the latitude error δL and the longitude error $\delta \lambda$; the accelerometer zero-biases ∇_x^b , ∇_y^b and ∇_z^b ; the constant gyro drifts ε_x^b , ε_y^b and ε_z^b —the state vector is built up as

$$X = \begin{bmatrix} \delta v_E & \delta v_N & \phi_x & \phi_y & \phi_z & \delta L & \delta \lambda & \nabla_x^b & \nabla_y^b & \nabla_z^b & \varepsilon_x^b & \varepsilon_y^b & \varepsilon_z^b \end{bmatrix}^T$$

The corresponding state equation is written as:

$$\dot{X}_{k+1} = f(X_k) + W_k$$

The state function $f(\cdot)$ is obtained from Equations (8)–(10).

We choose the quadrature point m = 3, then the total number of points $N_p = 3^{13}$, and that is a great amount of computation. However, according to Section 4.1, we know that only the Euler misalignment angle errors ϕ_x , ϕ_y , and ϕ_z are suffering from nonlinearity; therefore, the dimension reduction Gauss-Hermite filter proposed in Section 4.2. can be adopted, and the number of points will be reduced to $N_p = 3^3$. That greatly reduces the computational burden.

5. Calculation of the Gravity Direction

In Section 5, we present a method to set-up an accuracy horizontal reference frame in detail. After the fine alignment, we obtain the transfer matrix C_b^h , that is to say, the accuracy horizontal reference frame is established. Then, the projection of the gravity in the i_{b0} frame can be calculated as:

$$g^{i_{b0}}(t) = C_b^{i_{b0}}(t) [C_b^h]^{-1} g^h$$
(24)

where, $g^{h} = g^{n} = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^{T}$, $C_{b}^{i_{b0}}(t)$ is updated by the gyros' output in real time, and C_{b}^{h} is obtained from the fine horizontal alignment.

In order to improve the accuracy further, a weighted smoothing algorithm is adopted to inhibit the interference noise caused by the winds and waves. The algorithm is described as follows:

Assume that t_i (i \in (1, N)) is the sampling period, g_i is the corresponding sampling data, and the weight coefficient is 1. Then, the smoothed data is calculated as:

$$\overline{g} = \frac{1}{N+1} \sum_{i=-N/2}^{N/2} g_i$$
(25)

Since it is a coning motion of gravity in an inertial frame (i_{b0} frame or *i* frame), as Figure 4, the projection of $g^{i_{b0}}$ in the i_{b0} frame is a sine curve, and the period is 24 h. Compared with the alignment time, the period is so long that we can consider $g^{i_{b0}}(t)$ as linear *g*.

Then,

$$g_N^{i_{b0}} = C_{i_M}^{t_N} \overline{g} \tag{26}$$

where the transfer matrix from n_M frame (at t_M) to *n* frame (at t_N) could be calculated as follows:

$$C_{t_{M}}^{t_{N}} = C_{b}^{ib0}(t_{N}) \cdot \left[C_{b}^{ib0}(t_{M})\right]^{-1}$$
(27)

We can smooth $g^{i_{b0}}$ at 1 and 5 min respectively with Equation (25) after the fine horizontal alignment, and then $C_i^{i_{b0}}$ can be calculated with Equation (4).

Finally, by substituting $C_i^{i_{b0}}$ into Equation (1), the alignment can be completed.



Figure 4. The coning motion of gravity.

The flow chart of the improved alignment algorithm is shown in Figure 5.



Figure 5. Flow chart of the improved alignment algorithm.

As shown in the dashed box, we use a clever method to construct matrix $V^{i_{b0}}$. This improvement makes the traditional inertial alignment algorithm able to resist interference.

6. Simulation

In order to test the technique proposed in this paper, a simulation is carried out to compare with the conventional inertial frame alignment algorithm. The main parameters are set as follows in Table 3:

Parameters	Gyro	Accelerometer
Constant bias	0.01°/h	$10^{-4} g$
Random noise	$0.05 \circ \sqrt{h}$	$0.5 \times 10^{-4} g / \sqrt{\text{Hz}}$

Table 3. Specifications of IMU.

- (1) The latitude and longitude: $L = 45.7796^{\circ}$, $\lambda = 126.6705^{\circ}$;
- (2) The misalignment angle: $\phi_x = 10^\circ$, $\phi_y = 10^\circ$, $\phi_z = 10^\circ$;

(3) The sampling period is 0.1 s;

(4) The initial attitude is a random value.

(5) The first sampling time is $t_{k1} = 70$ s, and the second is $t_{k2} = 300$ s.

(6) We set up two situations to test the proposed algorithm, and the model is as follows:

Situation 1: Assume the ship is in the state of rest, without any interference.

Situation 2: Assume that the ship is on the berth and is rocked by the surf and wind. The pitch, roll, and yaw models of a marine vehicle are given by:

$$\Psi = \Psi_0 + \Psi_m \sin(2\pi t / T_{\Psi}) \tag{28}$$

$$\theta = \theta_0 + \theta_m \sin(2\pi t / T_\theta)$$
⁽²⁹⁾

$$\gamma = \gamma_0 + \gamma_m \sin(2\pi t / T_\gamma) \tag{30}$$

where, ψ , θ , and γ are yaw, pitch, and roll angles, respectively; the initial attitudes are $\psi_0 = 30^\circ$, $\theta_0 = \gamma_0 = 0$; the sway periods are $T_{\psi} = 6$ s, $T_{\theta} = 10$ s, and $T_{\gamma} = 8$ s; $\psi_m = 1^\circ$, $\theta_m = 5^\circ$, and $\gamma_m = 5^\circ$.

In the mooring condition, the horizontal velocities are small values because the mooring line secures the ship to the wharf, but vertical velocity may be not as small as horizontal velocities since a ship will heave along with the sea level fluctuation. So, velocity interference models are given by:

$$V_{dz} = 0.5 \cdot \sin(\omega_{dz} t + \varphi_{dz}) \tag{31}$$

$$V_{dx} = 0.02 \cdot \sin(\omega_{dx}t + \varphi_{dx}) \tag{32}$$

$$V_{dy} = 0.02 \cdot \sin(\omega_{dy}t + \varphi_{dy}) \tag{33}$$

where $\omega_d = \frac{2\pi}{T_d}$; $T_{dz} = 8$ s; $T_{dx} = T_{dy} = 2$ s; $\varphi_{dx}, \varphi_{dy}, \varphi_{dz}$ are random values in $[0, 2\pi]$.

The parameters of the dimension reduction GHF are chosen as follows:

$$\hat{x}_{0} = \begin{bmatrix} 10^{\circ} & 10^{\circ} & 0.1 \text{m/s} & 0.1 \text{m/s} & 0.1 \text{m} & 0.1 \text{m} \\ 1 \times 10^{-4} g_{0} & 1 \times 10^{-4} g_{0} & 1 \times 10^{-4} g_{0} & 0.01^{\circ}/\text{h} & 0.01^{\circ}/\text{h} & 0.01^{\circ}/\text{h} \end{bmatrix}$$

$$P_{0} = diag \begin{bmatrix} (5^{\circ})^{2} & (5^{\circ})^{2} & (5^{\circ})^{2} & (0.2 \text{m/s})^{2} & (0.2 \text{m/s})^{2} & (0.2 \text{m})^{2} \\ (1 \times 10^{-4} g_{0})^{2} & (1 \times 10^{-4} g_{0})^{2} & (1 \times 10^{-4} g_{0})^{2} & (0.01^{\circ}/\text{h})^{2} & (0.01^{\circ}/\text{h})^{2} \\ Q = diag \begin{bmatrix} (1 \times 10^{-4} g_{0})^{2} & (1 \times 10^{-4} g_{0})^{2} & (0.01^{\circ}/\text{h})^{2} & (0.01^{\circ}/\text{h})^{2} \end{bmatrix}$$

$$R = diag \begin{bmatrix} (0.01 \text{m/s})^{2} & (0.01 \text{m/s})^{2} \end{bmatrix}$$

For a fair comparison, 100 independent Monte Carlo runs are carried out. The results are as follows; the RMS error results are used to test the horizontal alignment accuracy and time consumed by dimensionality reduction GHF. RMS error is defined in Appendix D.

From Figures 6 and 7, it can be noted that the pitch error and roll error converge to the order of minutes within a few seconds. Figures 8–10 show the pitch error, roll error, and yaw error calculated by the traditional algorithm and the proposed algorithm, respectively. Their statistical results are shown in Table 4. From Table 4, we can tell that the accuracy of heading alignment is roughly the same (mean value, standard deviation and maximum value). However, the horizontal alignment results from the proposed algorithm are obvious better, since the standard deviation and the maximum value are smaller; the standard deviation in particular is an order of magnitude smaller than the traditional algorithm.

	Traditional Algorithm (min)			Improved Algorithm (min)		
	Pitch Error	Roll Error	Yaw Error	Pitch Error	Roll Error	Yaw Error
Mean	-0.1677	0.0156	0.3089	-0.0001	-0.2679	0.6363
Std	0.3475	0.3467	3.0679	0.0231	0.0163	4.2559
Max	0.6472	0.5212	7.1117	0.0723	0.3059	11.9556

Table 4. Statistics of situation 1.



Figure 6. The RMS error of pitch misalignment angle in situation 1.



Figure 7. The RMS error of roll misalignment angle in situation 1.



Figure 8. Pitch error result in situation 1.



Figure 10. Yaw error result in situation 1.

Figures 11 and 12 show the horizontal alignment results of the nonlinear filter in situation 2. The pitch error and roll error also converge rapidly (within tens of seconds) even though they converge slower than situation 1. Figures 13–15 show the pitch error, roll error, and yaw error calculated by traditional algorithm and the proposed algorithm in situation 2, respectively. Their statistical results are shown in Table 5. From Table 5, we can draw the conclusion that the proposed algorithm performs much better than the traditional one, because not only is the horizontal alignment accuracy an order of magnitude higher than the traditional one's (standard deviation and maximum value), but also the heading alignment accuracy of the proposed algorithm is better than the traditional one's. Compared with Tables 4 and 5, we can see that angular velocities and velocities have little effect on the proposed algorithm, but will greatly affect the precision of the traditional algorithm.



Figure 11. The RMS error of pitch misalignment angle in situation 2.



Figure 12. The RMS error of roll misalignment angle in situation 2







Figure 15. Yaw error result in situation 2.

Table 5.	Statistics	of situatio	on 2.
----------	------------	-------------	-------

	Traditional Algorithm (angular minute)			Improved Algorithm (angular minute)		
	Pitch Error	Roll Error	Yaw Error	Pitch Error	Roll Error	Yaw Error
Mean	-1.6843	-0.0872	2.5054	-0.0172	-0.2397	4.1585
Std	0.5111	0.5033	29.3313	0.02248	0.0174	4.2575
Max	2.4103	0.7588	44.1109	0.0680	0.2863	14.1987

7. Experiments

In order to evaluate the performance of the proposed self-alignment method for SINS, in this section, the mooring experiment was conducted in the East Sea of China. In this experiment, the ship was moored to the pier. A self-made SINS was used for the experiment, and the attitude reference was given by a PHINS (made by iXBlue Company) as shown in Figure 16. The self-made SINS and the PHINS were fixed on a rigid aluminum alloy board, and then the installation error was measured and compensated

for in the stationary state. The data acquisition computer collected the data of the self-made SINS and the PHINS synchronously. We carried out the alignment experiments three times.

The PHINS worked in GPS aided mode, its performance is as follows: pitch and roll errors are less than 0.01°, and heading error is less than 0.02°. In this experiment, we used the dimension reduction GHF mentioned in Section 4.2 to implement horizontal alignment. During the mooring experiments, the parameters of the dimension reduction GHF were optimally chosen as follows:

$$\hat{x}_{0} = \begin{bmatrix} 0^{\circ} & 0^{\circ} & 0^{\circ} & 0 \text{m/s} & 0 \text{m/s} & 0 \text{m} & 0 \text{m} \\ 1 \times 10^{-4} g_{0} & 1 \times 10^{-4} g_{0} & 0.01^{\circ}/\text{h} & 0.01^{\circ}/\text{h} & 0.01^{\circ}/\text{h} \end{bmatrix}$$

$$P_{0} = diag \begin{bmatrix} (1^{\circ})^{2} & (1^{\circ})^{2} & (3^{\circ})^{2} & (0.2 \text{m/s})^{2} & (0.2 \text{m/s})^{2} & (1 \text{m})^{2} \\ (1 \times 10^{-4} g_{0})^{2} & (1 \times 10^{-4} g_{0})^{2} & (1 \times 10^{-4} g_{0})^{2} & (0.01^{\circ}/\text{h})^{2} & (0.01^{\circ}/\text{h})^{2} \end{bmatrix}$$

$$Q = diag \begin{bmatrix} (1 \times 10^{-4} g_{0})^{2} & (1 \times 10^{-4} g_{0})^{2} & (0.01^{\circ}/\text{h})^{2} & (0.01^{\circ}/\text{h})^{2} \end{bmatrix}$$

$$R = diag \begin{bmatrix} (0.1 \text{m/s})^{2} & (0.1 \text{m/s})^{2} \end{bmatrix}$$



Figure 16. Self-made SINS, PHINS, and data acquisition computer.

The results are shown as follows. Figure 17 shows the estimations of misalignment angle.

From Figure 17, the estimations of misalignment angle converge within a few seconds. In order to ensure the estimations of misalignment angle are available, the pitch error and roll error are compensated for after 60 s. Figure 18 shows the attitude of the PHINS and the self-made SINS. For convenience, the initial 60 s part was omitted. We note that the difference between the two curves is very small (less than 0.05'). This indicates that the accurate horizontal reference frame is already established successfully. It is also proved that the estimations of the pitch error and roll error are accurate. From the attitude curves in Figure 18 and the velocity curves in Figure 19, we can also find that there are periodic disturbances during the alignment process.



Figure 17. Estimations of misalignment angle.



Figure 18. Attitude of PHINS and self-made SINS.



Figure 19. Velocity of PHINS.

Figure 20 presents the alignment results for the 3-time experiment. It is clear from Figure 20 that the alignment results of the improved algorithm are better than the traditional algorithm. The pitch errors and the roll errors of the improved algorithm are less than 0.06°, and yaw errors are less than 0.2°. That meets the accuracy requirements of a medium-accuracy marine SINS.



Figure 20. Comparison of the misalignment angles. (a) Pitch error; (b) roll error; (c) yaw error.

8. Conclusions

An improved inertial frame alignment algorithm based on horizontal alignment for marine SINS is proposed in this paper. The major improvement of this work is establishing a horizontal reference frame using a dimension reduction Gauss-Hermite filter. Based on that, the projection of gravity in the body inertial reference frame can be calculated and take the place of the accelerometer output to calculate the attitude matrix.

The dimension reduction Gauss-Hermite filter algorithm is detailed in this paper. The simulation and experimental results indicate that it can quickly and accurately complete the horizontal alignment. However, the parameters P_{0} , Q, and R which affect the filtering performance are usually chosen according to prior knowledge, as we have not yet found a mathematical method to get the optimal parameters. This means the selection of P_{0} , Q, and R is rather a matter of tuning. We also derive the error propagation equation and point out that the inertial frame alignment algorithm has the same theoretical accuracy as other algorithms.

The results of the simulation and the experiment also show that, compared with the traditional inertial frame alignment algorithm, the proposed algorithm can resist velocity and angular velocity interference to obtain higher accuracy, and meets the requirement of a medium-accuracy inertial navigation system.

Acknowledgments

The authors would like to thank Yalong liu for the helpful comments. This work was supported in part by the National Natural Science Foundation of China (51379042, 51379047) and the Fundamental Research Funds for the Central Universities, No. HEUCF150810, and the Postdoctoral grants of Heilongjiang Province 3236310313.

Conflicts of Interest

The authors declare no conflict of interest.

Appendix

A. Error Analysis

From Equation (1), the errors of the strapdown matrix $C_b^n(t)$ can be expressed as:

$$\delta C_b^n = \delta C_e^n C_i^e(t) C_{i_{b0}}^i C_b^{i_{b0}}(t) + C_e^n \delta C_i^e(t) C_{i_{b0}}^i C_b^{i_{b0}}(t) + C_e^n C_i^e(t) \delta C_{i_{b0}}^i C_b^{i_{b0}}(t) + C_e^n C_i^e(t) C_{i_{b0}}^i \delta C_b^{i_{b0}}(t)$$
(A1)

Because Equations (2) and (3) are respectively the functions of latitude and time, and they are exactly known during the alignment process, so that δC_e^n and $\delta C_i^e(t)$ can be ignored, the alignment error is caused by $\delta C_{i_{b0}}^i$ and $\delta C_b^{i_{b0}}(t)$.

Firstly, let us analyze the error of $C_b^{i_{b0}}(t)$. Its initial value is a constant matrix and contains no errors, so we only need to consider its updating error. In practice, $C_b^{i_{b0}}(t)$ is calculated through a differential equation:

$$\hat{\mathbf{C}}_{b}^{i_{b0}} = \hat{\mathbf{C}}_{b}^{i_{b0}} \left(\hat{\boldsymbol{\omega}}_{bb}^{b} \times \right) \tag{A2}$$

where $(\hat{\omega}_{ib}^{b} \times)$ is the skew symmetric matrix of the vector; $\hat{\omega}_{ib}^{b} = \begin{bmatrix} \hat{\omega}_{ibx}^{b} & \hat{\omega}_{iby}^{b} & \hat{\omega}_{ibz}^{b} \end{bmatrix}^{T}$ which is measured by the gyros, represents the turn rate of the *b* frame with respect to the *i* frame; and $\hat{\omega}_{ib}^{b} = \omega_{ib}^{b} + \varepsilon^{b}$, ε^{b} is the gyro bias. Obviously, ε^{b} directly produces the error in the calculation of $C_{b}^{i_{b0}}(t)$. We note that to solve Equation (7) we must carry out two integrations of ε^{b} , which will certainly increase the calculation error. In order to facilitate analysis, we define ϕ as the attitude error angle,

$$\hat{C}_{b}^{n} = \left(I - \left[\left(\phi^{n}\right)\times\right]\right)C_{b}^{n}$$
(A3)

$$\hat{C}^{i}_{i_{b0}} = \left(I - \left[\left(\phi^{i}\right)\times\right]\right)C^{i}_{i_{b0}}$$
(A4)

$$\hat{C}_{b}^{i_{b0}} = \left(I - \left[\left(\phi^{i_{b0}}\right) \times\right]\right) C_{b}^{i_{b0}} \tag{A5}$$

Substitute Equations (A3)–(A5) into Equation (1), and ignore the second-order small items. We get:

$$\hat{C}_{b}^{n} = C_{i}^{n} \hat{C}_{i_{b0}}^{i} \hat{C}_{b}^{i_{b0}}
= C_{i}^{n} \left(I - \left[\left(\phi^{i} \right) \times \right] \right) C_{i_{b0}}^{i} \left(I - \left[\left(\phi^{i_{b0}} \right) \times \right] \right) C_{b}^{i_{b0}}
\approx C_{i}^{n} \left(C_{b}^{i} - C_{i_{b0}}^{i} \left[\left(\phi^{i_{b0}} \right) \times \right] C_{i}^{i_{b0}} C_{b}^{i} - \left[\left(\phi^{i} \right) \times \right] C_{b}^{i} \right)
= \left(I - \left[\left(\phi_{2}^{n} \right) \times \right] - \left[\left(\phi_{3}^{n} \right) \times \right] \right) C_{b}^{n}$$
(A6)

where the similarity transformation theorem for matrix is used, and

$$\phi_2^n = C_{i_{b0}}^n \phi^{i_{b0}} \tag{A7}$$

$$\phi_3^n = C_i^n \phi^i \tag{A8}$$

Comparing Equation (A3) and Equation (A6), we know that:

$$\phi^n = \phi_2^n + \phi_3^n \tag{A9}$$

Considering that $\hat{\omega}_{i_{b0}}^{b} = \omega_{i_{b0}}^{b} + \varepsilon^{b}$, ε^{b} is gyros' bias, substitute Equation (A5) into Equation (4) and ignore the second-order small items, and we can obtain:

$$\dot{\phi}^{i_{b0}} = -C_b^{i_{b0}}\varepsilon^b \tag{A10}$$

By solving Equation (17), we get:

$$\phi^{i_{b0}}(t) = -\int_{0}^{t} C_{b}^{i_{b0}}(t) \varepsilon^{b} dt$$
(A11)

Then, by substituting Equation (18) into Equation (14), ϕ_2^n can be expressed as follows:

$$\begin{split} \phi_{2}^{n} &= -C_{i_{b0}}^{n}\left(t\right) \int_{0}^{t} C_{b}^{i_{b0}}\left(t\right) \varepsilon^{b} dt \\ &= -C_{i}^{n}\left(t\right) C_{i_{b0}}^{i} \int_{0}^{t} C_{b}^{i_{b0}}\left(t\right) \varepsilon^{b} dt \\ &= -C_{i}^{n}\left(t\right) \int_{0}^{t_{k}} C_{i_{b0}}^{i} C_{n}^{i_{b0}}\left(t\right) C_{b}^{n} \varepsilon^{b} dt \\ &= -C_{i}^{n}\left(t\right) \int_{0}^{t_{k}} C_{n}^{i}\left(t\right) \varepsilon^{n} dt \end{split}$$
(A12)

where

$$C_{i}^{n}(t) = C_{e}^{n}C_{i}^{e}(t) = \begin{bmatrix} \cos(\omega_{ie}t) & \sin(\omega_{ie}t) & 0\\ -\sin L\sin(\omega_{ie}t) & \sin L\cos(\omega_{ie}t) & \cos L\\ \cos L\sin(\omega_{ie}t) & -\cos L\cos(\omega_{ie}t) & \sin L \end{bmatrix}$$
(A13)

$$C_{n}^{i}(t) = \left(C_{i}^{n}(t)\right)^{T} = \begin{bmatrix} \cos(\omega_{ie}t) & -\sin L \sin(\omega_{ie}t) & \cos L \sin(\omega_{ie}t) \\ \sin(\omega_{ie}t) & \sin L \cos(\omega_{ie}t) & -\cos L \cos(\omega_{ie}t) \\ 0 & \cos L & \sin L \end{bmatrix}$$
(A14)

$$\varepsilon^{n} = \begin{bmatrix} \varepsilon_{e} & \varepsilon_{n} & \varepsilon_{u} \end{bmatrix}^{T}$$
(A15)

Then, substituting Equations (A13)–(A15) into Equation (A12), we obtain:

$$\phi_{2}^{n} = \begin{bmatrix} -\frac{\sin L \cdot \varepsilon_{n}}{\omega_{ie}} + \frac{\cos L \cdot \varepsilon_{u}}{\omega_{ie}} \\ \frac{\sin L \cdot \varepsilon_{e}}{\omega_{ie}} - \cos^{2} L \cdot t \cdot \varepsilon_{n} - \cos L \cdot \sin L \cdot t \cdot \varepsilon_{u} \\ -\frac{\cos L \cdot \varepsilon_{e}}{\omega_{ie}} - \cos L \cdot \sin L \cdot t \cdot \varepsilon_{n} - \sin^{2} L \cdot t \cdot \varepsilon_{u} \end{bmatrix}$$
(A16)

Next, let us analyze the error of $C_{i_{b_0}}^i$. In practice, accelerometer measurements contain interference error. Therefore, the $g^{i_{b_0}}$ used in Equation (5) is relatively difficult to get. For purposes of analysis, consider the static base condition; let $\hat{f}^b = f^b + \nabla^b$, where ∇^b is the accelerometer bias. If neglecting the second-order small items, we have

$$\hat{f}^{i_{b0}} = \hat{C}^{i_{b0}}_{b} \hat{f}^{b} = \left[I - (\phi^{i_{b0}} \times) \right] C^{i_{b0}}_{b} \left(f^{b} + \nabla^{b} \right) \approx f^{i_{b0}} - (\phi^{i_{b0}} \times) f^{i_{b0}} + \nabla^{i_{b0}}$$
(A17)

Then, substitute Equation (A17) into Equation (5) and define

$$\left(\mathbf{G}^{i} \right)^{-1} = \begin{bmatrix} \left[g^{i}(t_{1}) \right]^{T} \\ \left[g^{i}(t_{2}) \right]^{T} \\ \left[g^{i}(t_{2}) \right]^{T} \end{bmatrix}^{-1}$$
(A18)
$$F^{i_{b0}} = \begin{bmatrix} \left[f^{i_{b0}}(t_{1}) \right]^{T} \\ \left[f^{i_{b0}}(t_{2}) \right]^{T} \\ \left[f^{i_{b0}}(t_{1}) \times f^{i_{b0}}(t_{2}) \right]^{T} \end{bmatrix}$$
(A19)
$$\begin{bmatrix} \left[\hat{f}^{i_{b0}}(t_{1}) \right]^{T} \\ \left[\hat{f}^{i_{b0}}(t_{1}) \right]^{T} \end{bmatrix}$$

$$\hat{F}^{i_{b0}} = \begin{bmatrix} \begin{bmatrix} \hat{f}^{i_{b0}}(t_2) \end{bmatrix}^T \\ \begin{bmatrix} \hat{f}^{i_{b0}}(t_1) \times \hat{f}^{i_{b0}}(t_2) \end{bmatrix}^T \end{bmatrix}$$
(A20)

$$\hat{F}^{i_{b0}} = F^{i_{b0}} + \delta F^{i_{b0}} \tag{A21}$$

Substitute Equation (A17) into Equation (A20) and ignore second-order small item, and we get

$$\delta F^{i_{b0}} = \hat{F}^{i_{b0}} - F^{i_{b0}} \approx \begin{bmatrix} \left[\nabla^{i_{b0}} - \phi^{i_{b0}}(t_{1}) \times f^{i_{b0}}(t_{1}) \right]^{T} \\ \left[\nabla^{i_{b0}} - \phi^{i_{b0}}(t_{2}) \times f^{i_{b0}}(t_{2}) \right]^{T} \\ \left\{ \left[\nabla^{i_{b0}} - \phi^{i_{b0}}(t_{1}) \times f^{i_{b0}}(t_{1}) \right] \times f^{i_{b0}}(t_{2}) + f^{i_{b0}}(t_{1}) \times \left[\nabla^{i_{b0}} - \phi^{i_{b0}}(t_{2}) \times f^{i_{b0}}(t_{2}) \right]^{T} \end{bmatrix}$$
(A22)

According to Equation (6), we know that $(G^i)^{-1}$ will not introduce any error during the calculation process. Therefore,

$$\hat{C}_{i_{b0}}^{i} = \left(G^{i}\right)^{-1} \hat{F}^{i_{b0}} = \left(G^{i}\right)^{-1} F^{i_{b0}} + \left(G^{i}\right)^{-1} \delta F^{i_{b0}} = \left(I + \left(G^{i}\right)^{-1} \delta F^{i_{b0}} C^{i_{b0}}_{i}\right) C^{i}_{i_{b0}}$$
(A23)

According to reference [1], we have

$$\hat{C}_{i_{b0}}^{i} = \hat{C}_{i_{b0}}^{i} \left[\left(\hat{C}_{i_{b0}}^{i} \right)^{T} \hat{C}_{i_{b0}}^{i} \right]^{-\frac{1}{2}} \approx \left[I - \frac{1}{2} \left(\left(\left(G^{i} \right)^{-1} \delta F^{i_{b0}} C_{i}^{i_{b0}} \right)^{T} - \left(G^{i} \right)^{-1} \delta F^{i_{b0}} C_{i}^{i_{b0}} \right) \right] C_{i_{b0}}^{i}$$
(A24)

Compare Equations (A4) and (A24), we know that

$$\left(\phi^{i}\times\right) = \frac{1}{2} \left(\left(\left(G^{i}\right)^{-1} \delta F^{i_{b0}} C_{i}^{i_{b0}}\right)^{T} - \left(G^{i}\right)^{-1} \delta F^{i_{b0}} C_{i}^{i_{b0}}\right)$$
(A25)

Since $C_i^{i_{b0}}$ is an orthonormal matrix, when Equation (A17) is substituted into Equation (5) we get

$$\delta F^{i_{b0}} C_{i}^{i_{b0}} = \begin{bmatrix} \begin{bmatrix} \nabla^{i} - \phi^{i}(t_{1}) \times f^{i}(t_{1}) \end{bmatrix}^{T} \\ \begin{bmatrix} \nabla^{i} - \phi^{i}(t_{2}) \times f^{i}(t_{2}) \end{bmatrix}^{T} \\ \left\{ \begin{bmatrix} \nabla^{i} - \phi^{i}(t_{1}) \times f^{i}(t_{1}) \end{bmatrix} \times f^{i}(t_{2}) + f^{i}(t_{1}) \times \begin{bmatrix} \nabla^{i} - \phi^{i}(t_{2}) \times f^{i}(t_{2}) \end{bmatrix} \right\}^{T} \end{bmatrix} = \delta F_{A}^{i} + \delta F_{G}^{i} \quad (A26)$$

$$\delta F_{A}^{i} = \begin{bmatrix} \left(\nabla^{i}\right)^{T} \\ \left(\nabla^{i}\right)^{T} \\ \left[\nabla^{i} \times f^{i}(t_{2}) + f^{i}(t_{1}) \times \nabla^{i}\right]^{T} \end{bmatrix}$$
(A27)
$$\delta F_{G}^{i} = -\begin{bmatrix} \left[\phi^{i}(t_{1}) \times f^{i}(t_{1})\right]^{T} \\ \left[\phi^{i}(t_{2}) \times f^{i}(t_{2})\right]^{T} \\ \left[\phi^{i}(t_{2}) \times f^{i}(t_{2}) + f^{i}(t_{1}) \times \left[\phi^{i}(t_{2}) \times f^{i}(t_{2})\right]\right]^{T} \end{bmatrix}$$
(A28)

According to the principle of coordinate transformation we have

$$\nabla^i = C_n^i \nabla^n \tag{A29}$$

$$\left[\phi^{i} \times f^{i}\right] = -\left(\int_{0}^{t} \left[\varepsilon^{i} \times\right] dt\right) C_{n}^{i} g^{n}$$
(A30)

$$\varepsilon^i = C_n^i \varepsilon^n \tag{A31}$$

Under the static conditions, $f^i = g^i$, if we introduce Equations (34)–(38) into Equation (32), and assume $\sin(\omega t) \approx 0$, $\cos(\omega t) \approx 1$, we finally obtain the errors ϕ_g^i and ϕ_a^i caused by gyros and accelerometers, respectively. Then, project them into the navigation frame, and we have

$$\phi_{3a}^{n} = \left[-\frac{\nabla_{n}}{g} \quad \frac{\sin L \cdot \omega_{ie} t \nabla_{n}}{g} + \frac{\nabla_{e}}{g} \quad -\frac{\cos L \cdot \omega_{ie} t \nabla_{n}}{g} + \frac{\nabla_{e}}{g} \tan L \right]^{T}$$
(A32)

$$\phi_{3g}^{n} = \left[\frac{\sin L \cdot \varepsilon_{n}}{\omega_{ie}} - \frac{\cos L \cdot \varepsilon_{u}}{\omega_{ie}} - \frac{\varepsilon_{e} \sin L}{\omega_{ie}} + \frac{\varepsilon_{n} t}{2} - \frac{\sin L^{2} \varepsilon_{e}}{\cos L \omega_{ie}} + \varepsilon_{u} t\right]^{T}$$
(A33)

According to Equations (A16), (A32) and (A33), we can get the following result: the misalignment angle caused by gyro error is

$$\phi_g^n = \phi_2^n + \phi_{3g}^n = \begin{bmatrix} 0 & (\frac{1}{2} - \cos^2 L)\varepsilon_n t + \sin L \cos L\varepsilon_u t & \cos^2 L \cdot \varepsilon_u t - \sin L \cos L\varepsilon_n t - \frac{\varepsilon_e}{\cos L \cdot \omega_{ie}} \end{bmatrix}^T$$
(A34)

The misalignment angle caused by accelerometers' error is expressed as Equation (A32).

For further analysis, assume that t = 360s, $\nabla_e = \nabla_n = \nabla_u = 1 \times 10^{-4} g$, $\varepsilon_e = \varepsilon_n = \varepsilon_u = 0.01^\circ / h$. It is also known that $\omega_{ie} \approx 7.3 \times 10^{-5} rad / s$ and $L \in [0, \frac{\pi}{2}]$. Therefore, it can be calculated approximately and we find that the error terms consisting of t are very small and can be ignored. Therefore,

$$\phi^{n} = \phi_{3a}^{n} + \phi_{g}^{n} \approx \begin{bmatrix} -\frac{\nabla_{n}}{g} \\ \frac{\nabla_{e}}{g} \\ -\frac{\varepsilon_{e}}{\cos L \cdot \omega_{ie}} + \frac{\nabla_{e}}{g} \tan L \end{bmatrix}$$
(A35)

Through analysis, we know that the pitch angle error and the roll angle error depend on the accelerometer bias. The yaw angle error is determined by equivalent east gyro drift, equivalent east accelerometer bias and latitude.

B. Gaussian Approximation Filters

Consider the following discrete-time nonlinear state-space model:

$$x_{k+1} = f(x_k) + w_k \tag{B1}$$

$$y_k = h(x_k) + v_k \tag{B2}$$

where, $k \in \{1, 2, ...\}$ is discrete time, x_k and y_k are the state vector and the measurement vector at time k, respectively; w_k and v_k are the noise vectors from two independent zero-mean Gaussian processes with their covariance matrices Q_k and R_k , respectively.

Under the assumption of Gaussian distributions, the general form of Gaussian approximation filtering algorithm can be summarized as follows:

$$\hat{x}_{k+1} = \hat{x}_{k+1/k} + K_{k+1}(y_{k+1} - \hat{y}_{k+1})$$
(B3)

$$K_{k+1} = P_{k+1/k}^{xy} (P_{k+1}^{y})^{-1}$$
(B4)

$$P_{k+1} = P_{k+1/k} - K_{k+1} P_{k+1}^{y} K_{k+1}^{T}$$
(B5)

where, \hat{x}_{k+1} is the estimation of the state; \hat{y}_{k+1} is the estimation of the measurement with the covariance matrix P_{k+1}^{y} ; $\hat{x}_{k+1/k}$ is the state prediction; K_{k+1} is the gain matrix; $P_{k+1/k}^{xy}$ is the cross covariance matrix; $N(\mathbf{x}_{k}; \hat{\mathbf{x}}_{k}, \mathbf{P}_{k})$ denotes the multivariate normal distribution with mean \hat{x}_{k} and covariance P_{k}

$$P_{k+1} = P_{k+1/k} - K_{k+1} P_{k+1}^{y} K_{k+1}^{T}$$
(B6)

$$P_{k+1/k} = \int_{\mathbb{R}^n} f(x_k) f^T(x_k) N(x_k; \hat{x}_k, P_k) dx_k - \hat{x}_{k+1/k} \hat{x}_{k+1/k}^T + Q_k$$
(B7)

$$\hat{y}_{k+1/k} = \int_{\mathbb{R}^n} h(x_{k+1}) N(x_{k+1}; \hat{x}_{k+1/k}, P_{k+1/k}) dx_{k+1}$$
(B8)

$$P_{k+1}^{y} = \int_{\mathbb{R}^{n}} h(x_{k+1}) h^{T}(x_{k+1}) N(x_{k+1}; \hat{x}_{k+1/k}, P_{k+1/k}) dx_{k+1} - \hat{y}_{k+1/k} \hat{y}_{k+1/k}^{T} + R_{k+1}$$
(B9)

$$P_{k+1/k}^{xy} = \int_{\mathbb{R}^n} x_{k+1} h^T(x_{k+1}) N(x_{k+1}; \hat{x}_{k+1/k}, P_{k+1/k}) dx_{k+1} - \hat{x}_{k+1/k} \hat{y}_{k+1/k}^T$$
(B10)

The integrals in can be approximated by the GHQ, UT, or cubature rule [20,22,23]. The generic quadrature-based Gaussian approximation filter is given as [22,23].

Prediction:

$$\hat{x}_{k+1/k} = \sum_{i=1}^{N_p} w_i f\left(\varepsilon_i\right) \tag{B11}$$

$$P_{k+1/k} = \sum_{i=1}^{N_p} w_i \left[f\left(\varepsilon_i\right) - \hat{x}_{k+1/k} \right] \left[f\left(\varepsilon_i\right) - \hat{x}_{k+1/k} \right]^T + Q_k$$
(B12)

where, N_p is the total number of points; ε_i is the transformed point obtained from the covariance decomposition such as Cholesky decomposition; that is

$$P_k = SS^T \tag{B13}$$

$$\varepsilon_i = S\gamma_i + \hat{x}_k \tag{B14}$$

 γ_i is the point corresponding to N(x;0,I); and w_i is the associated weight. Update:

$$\hat{x}_{k+1} = \hat{x}_{k+1/k} + K_{k+1}(y_{k+1} - \hat{y}_{k+1})$$
(B15)

$$P_{k+1} = P_{k+1|k} - K_k P_{k+1/k}^{xy}$$
(B16)

where

$$\hat{y}_{k+1/k} = \sum_{i=1}^{N_p} w_i h(\tilde{\varepsilon}_i)$$
(B17)

$$P_{k+1/k}^{xy} = \sum_{i=1}^{N_p} w_i \left(\tilde{\varepsilon}_i - \hat{x}_{k+1/k}\right) \left[h(\tilde{\varepsilon}_i) - \hat{y}_{k+1/k}\right]^T$$
(B18)

$$P_{k+1}^{y} = \sum_{i=1}^{N_{p}} w_{i} \left[h(\tilde{\varepsilon}_{i}) - \hat{y}_{k+1/k} \right] \left[h(\tilde{\varepsilon}_{i}) - \hat{y}_{k+1/k} \right]^{T} + R_{k+1}$$
(B19)

where $\tilde{\varepsilon}_i$ is the transformed point obtained from the predicted covariance decomposition; that is

$$P_{k+1|k} = \tilde{S}\tilde{S}^{T} \tag{B20}$$

$$\tilde{\varepsilon}_i = \tilde{S}\gamma_i + \hat{x}_{k+1/k} \tag{B21}$$

C. Gauss-Hermite Quadrature Point Selection

For the univariate standard Gaussian distribution with m quadrature points (γ_i , w_i) can be calculated as follows [23,25].

If m = 1, then $(\gamma_i, w_i) = (0, 1)$.

If m > 1, firstly, a symmetric tridiagonal matrix J needs to be constructed, whose diagonal elements are zero and $J_{i,i+1} = \sqrt{\frac{i}{2}} (1 \le i \le m-1)$. Then the quadrature point $(\gamma_i, w_i) = (\sqrt{2}\xi_i, (v_i)_1^2)$, where ξ_i is the i-th eigenvalue of J, $(v_i)_1$ is the first element of the i-th normalized eigenvector of J.

For example,
$$m = 3$$
, we have $(\gamma_i, w_i) = \left\{ (-\sqrt{3}, \frac{1}{6}), (0, \frac{2}{3}), (\sqrt{3}, \frac{1}{6}) \right\}$.

The multivariate GHQ rule extends the univariate m-point set to the n-dimensional point set by the tensor product rule. However, the total number of points is $N_p = m^n$, which increases exponentially with dimension n. That will result in the so-called" curse of dimensionality".

D. RMS Error

RMS error means the square root of the mean/average of the square of all of the error. It is commonly used and makes an excellent general purpose error metric. The calculation formula is as follows:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(E_i - \overline{E} \right)^2}$$
(D1)

where N is the number of experiments, in this paper N = 100, E_i is the *i*-th experimental result, and \overline{E} denotes arithmetic mean value of all the experimental results.

References

- 1. Britting, K.R. *Inertial Navigation Systems Analysis*; Artech House Publishers: Norwood, MA, USA, 1971.
- Titterton, D.; Weston, J.L. Strapdown Inertial Navigation Technology; IET: London, UK, 2004; Volume 17.
- 3. Qin, Y.; Yan, G.; Gu, D.; Zheng, J. Clever way of sins coarse alignment despite rocking ship. *Xibei Gongye Daxue Xuebao/J. Northwest. Polytech. Univ.* **2005**, *23*, 681–684.
- 4. Sun, F.; Lan, H.Y.; Yu, C.Y.; El-Sheimy, N.; Zhou, G.T.; Cao, T.; Liu, H. A robust self-alignment method for ship's strapdown INS under mooring conditions. *Sensors* **2013**, *13*, 8103–8139.
- Gu, D.Q.; El-Sheimy, N.; Hassan, T.; Syed, Z.; Coarse alignment for marine sins using gravity in the inertial frame as a reference. In Proceedings of the 2008 IEEE/ION Position, Location and Navigation Symposium, Monterey, CA, USA, 5–8 May 2008; pp. 961–965.
- 6. Silson, P.M.G. Coarse alignment of a ship's strapdown inertial attitude reference system using velocity loci. *IEEE Trans. Instrum. Meas.* **2011**, *60*, 1930–1941.
- Sun, F.; Sun, W. Mooring alignment for marine SINS using the digital filter. *Measurement* 2010, 43, 1489–1494.
- 8. Gao, W.; Ben, Y.Y.; Zhang, X.; Li, Q.; Yu, F. Rapid fine strapdown ins alignment method under marine mooring condition. *IEEE Trans. Aerosp. Electron. Syst.* **2011**, *47*, 2887–2896.
- Gao, W.; Che, Y.; Zhang, X.; Feng, J.; Zhang, B. A fast alignment algorithm based on inertial frame for marine SINS. In Proceedings of the 2012 9th IEEE International Conference on Mechatronics and Automation (ICMA 2012), Chengdu, China, 5–8 August 2012; pp. 1756–1760.
- 10. Hong, H.S.; Lee, J.G.; Park, C.G. In-flight alignment of SDINS under large initial heading error. *Trans. Soc. Instrum. Control Eng.* **2002**, *38*, 1107–1113.
- Wei, C.; Zhang, H. Sins in-flight alignment using quaternion error models. *Chinese J. Aeronaut.* 2001, 14, 166–170.
- Yu, M.-J.; Park, H.-W.; Jeon, C.-B. Equivalent nonlinear error models of strapdown inertial navigation system. In Proceedings of the 1997 AIAA Guidance, Navigation, and Control Conference, New Orleans, LA, USA, 11–13 August 1997; pp. 581–587.

- 13. Dmitriyev, S.P.; Stepanov, O.A.; Shepel, S.V. Nonlinear filtering methods application in ins alignment. *IEEE Trans. Aerosp. Electron. Syst.* **1997**, *33*, 260–272.
- 14. Wang, B.; Xiao, X.; Xia, Y.Q.; Fu, M.Y. Unscented particle filtering for estimation of shipboard deformation based on inertial measurement units. *Sensors* **2013**, *13*, 15656–15672.
- Guo, L.; Cao, S.Y.; Qi, C.T.; Gao, X.Y. Initial alignment for nonlinear inertial navigation systems with multiple disturbances based on enhanced anti-disturbance filtering. *Int. J. Control.* 2012, *85*, 491–501.
- 16. Kubo, Y.; Fujioka, S.; Nishiyama, M.; Sugimoto, S. Nonlinear filtering methods for the INS/GPS in-motion alignment and navigation. *Int J. Innov. Comput. Inf. Control* **2006**, *2*, 1137–1151.
- 17. Gao, W.; Che, Y.; Yu, F.; Liu, Y. A fast inertial frame alignment algorithm based on horizontal alignment information for marine SINS. In Proceedings of the 2014 IEEE/ION Position, Location and Navigation Symposium-PLANS 2014, Monterey, CA, USA, 5–8 May 2014; pp. 1415–1419.
- 18. Anderson, B.D.; Moore, J.B. Optimal Filtering; Dover Publications: Englewood Cliffs, NJ, USA, 1979.
- 19. Smith, G.L.; Schmidt, S.F.; McGee, L.A. *Application of Statistical Filter Theory to the Optimal Estimation of Position and Velocity on Board a Circumlunar Vehicle*; National Aeronautics and Space Administration: Washington, DC, USA, 1962.
- 20. Julier, S.J.; Uhlmann, J.K. Unscented filtering and nonlinear estimation. Proc. IEEE 2004, 92, 401-422.
- 21. Arasaratnam, I.; Haykin, S.; Hurd, T.R. Cubature kalman filtering for continuous-discrete systems: Theory and simulations. *IEEE Trans. Signal. Process.* **2010**, *58*, 4977–4993.
- 22. Arasaratnam, I.; Haykin, S. Cubature kalman filters. IEEE Trans. Autom. Control. 2009, 54, 1254–1269.
- Ito, K.; Xiong, K. Gaussian filters for nonlinear filtering problems. *IEEE Trans. Autom. Control.* 2000, 45, 910–927.
- 24. Gao, W.; Zhang, Y.; Wang, J.G. A strapdown interial navigation system/beidou/doppler velocity log integrated navigation algorithm based on a cubature kalman filter. *Sensors* **2014**, *14*, 1511–1527.
- 25. Arasaratnam, I.; Haykin, S.; Elliott, R.J. Discrete-time nonlinear filtering algorithms using gauss-hermite quadrature. *Proc. IEEE* 2007, *95*, 953–977.
- Jia, B.; Xin, M.; Cheng, Y.; Sparse gauss-hermite quadrature filter for spacecraft attitude estimation. In Proceedings of the 2010 American Control Conference, Baltimore, MD, USA, 30 June–2 July 2010; pp. 2873–2878.
- 27. Jia, B.; Xin, M.; Cheng, Y. High-degree cubature kalman filter. Automatica 2013, 49, 510–518.

 \odot 2015 by the authors; license MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/4.0/).