Research Article

A Fast in-Motion Alignment Algorithm for DVL Aided SINS

Li Kang, Lingyun Ye, and Kaichen Song

College of Biomedical Engineering & Instrument Science, Zhejiang University, Hangzhou, Zhejiang 310027, China

Correspondence should be addressed to Lingyun Ye; lyye@zju.edu.cn

Received 3 March 2014; Revised 12 May 2014; Accepted 12 May 2014; Published 4 June 2014

Academic Editor: Chia-Cheng Tsai

Copyright © 2014 Li Kang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Doppler velocity log (DVL) aided strapdown inertial navigation system (SINS) is a common navigation method for underwater applications. Owing to the in-motion condition and the lack of the GPS, it is a challenge to align a SINS under water. This paper proposed a complete in-motion alignment solution for both attitude and position. The velocity update equation and its integral form in the body frame are studied, and the attitude coarse alignment becomes an optimization-based attitude determination problem between the body frame velocity and the integral form of gravity. The body frame velocity and the Earth frame position are separately treated, and the position alignment problem turns into an equation solving problem. Simulation and on-lake tests are carried out to examine the algorithm. The heading could reach around 10 deg accuracy and the pitch and roll could be aligned up to 0.05 deg in 60 s. With attitude error of this level, the heading could reach 1 deg accuracy in 240 s using unscented Kalman filter (UKF) based fine alignment. The final position error could achieve 1.5% of the voyage distance. This scheme can also be applied to other body frame velocity aided SINS alignments.

1. Introduction

DVL aided SINS is now widely used in the autonomous underwater vehicles (AUVs) [1–3]. It is a challenge to align a SINS under water for two reasons. The first is that the alignment needs to finish in-motion since the vehicle cannot keep stationary due to the water flow. The second is that the GPS [4, 5], the most useful aided sensor, which could provide geodetic frame velocity and the Earth frame position, is not available under water. So, it is necessary to develop DVL aided in-motion alignment. Due to the lack of position information, the alignment should include both attitude and position.

The attitude alignment often includes two stages: coarse alignment and fine alignment. The attitude fine alignment is mainly based on the Earth frame position or geodetic velocity matching with Kalman filter (KF) [6]. Currently, the body frame velocity matching alignment could already partly solve the problem if a roughly known attitude could pass to the filter. But if the error of the roughly known attitude is too large, the alignment would take much time and the final results would be worse [7]. So the key process is to get this

roughly known attitude. This process is called the attitude coarse alignment.

Various algorithms for the coarse alignment have been developed. One kind of algorithms determined the attitude from gyroscope and accelerometer measurements using the gravity and the Earth's rotation as a reference [6, 8, 9]. These algorithms are fit for the condition that the SINS is stationary or quasi-stationary. While in-motion condition the angular rate is much larger than the Earth's rotation, the attitude error is large in most situations. Another kind of algorithms applied the velocity of the GPS as an input and optimization approach is used between the velocities of the navigation frame and the body frame [10, 11]. This method could offer a good result to the fine alignment in in-motion condition. But there is no GPS signal under water. Also it is not a good idea to get the initial attitude using a magnetic compass, since there are many electromagnetic interference sources such as the motors and the solenoid for the emergency jettison. The magnetic compass is not robust in such harsh environment.

The task of the position alignment is to provide the position during the attitude alignment process. This is mainly provided by the GPS for on-land assignments [11], while

the GPS is not available for underwater assignments. The main idea is to use the DVL to reckon the coordinate from the start point. The problem is that the body frame velocity cannot be integrated into position before the attitude is known.

The contents are organized as follows. Section 2 explains the problem in mathematics. Section 3 is the attitude alignment section. The attitude coarse alignment is devised by further study of the velocity update equation in the navigation frame and its transformation in the body frame. Optimization approach is applied between the body frame velocity and the integral form of the gravity. Also the common large misalignment angles model based UKF is reviewed in Section 3. Section 4 proposed the position alignment by separately treating the body frame velocity and the Earth frame position. And when the initial attitude is gathered, the position could be straight obtained through an equation. The performance of the algorithm is evaluated with simulation in Section 5 and on-lake tests in Section 6. Conclusions are drawn in Section 7.

2. Problem Statement

The coordinate frames used for the in-motion alignment are defined as follows.

- (i) The n frame is the ideal local level navigation coordinate frame with east-north-up geodetic axes.
- (ii) The n' frame is the real local level navigation coordinate frame; there is some error between n and n' frame owing to the alignment error and the sensor error.
- (iii) The b frame is the strapdown inertial sensor's body coordinate frame.
- (iv) The e frame is the Earth coordinate frame.
- (v) The i frame is the nonrotating inertial coordinate frame.

The common navigation (attitude, velocity, and position) update equations are written in the n frame as [12–15]

$$\dot{C}_b^n = C_b^n \omega_{nb}^b \times, \quad (1)$$

$$\dot{\mathbf{v}}^n = C_b^n \mathbf{f}^b - (2\omega_{ie}^n + \omega_{en}^n) \times \mathbf{v}^n + \mathbf{g}^n, \quad (2)$$

$$\dot{\mathbf{p}} = \mathbf{R}_c \mathbf{v}^n, \quad (3)$$

where C_b^n is the attitude direction cosine matrix (DCM) from the b frame to the n frame, $\mathbf{v}^n$ is the velocity in the n frame, $\mathbf{f}^b$ is the special force measured by the accelerometers fixed in the b frame, ω_{ie}^n is the angular rate of the Earth's rotation in the n frame, ω_{en}^n is the angular rate caused by the velocity on the Earth (since the Earth is nearly round) in the n frame, $\mathbf{g}^n$ is the gravity in the n frame, $(\cdot) \times$ is the matrix form of a cross-product which satisfies $\mathbf{a} \times \mathbf{b} = (\mathbf{a} \times) \mathbf{b}$, $\mathbf{p}$ is the position described in the e frame as $\mathbf{p} = [\lambda, \varphi, h]^T$, λ is the longitude, φ is the latitude, h is the height above the mean sea level, R is the radius of the Earth, ω_{ib}^b is the angular rate measured by

the gyroscopes in the b frame, ω_{nb}^b is the angular rate of the SINS in the b frame with respect to the n frame described as

$$\omega_{nb}^b = \omega_{ib}^b - C_n^b (\omega_{ie}^n + \omega_{en}^n), \quad (4)$$

and $\mathbf{R}_c$ is the velocity transformation matrix between the n frame and the e frame defined as

$$\mathbf{R}_c = \begin{bmatrix} \frac{\sec \varphi}{R+h} & 0 & 0 \\ 0 & \frac{1}{R+h} & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (5)$$

The velocity $\mathbf{v}^n$ can be measured by the DVL if C_b^n is known. The purpose of the alignment is to determine the attitude DCM C_b^n and the position $\mathbf{p}$ based on the data collected from the accelerometers, the gyroscopes, and the DVL while in motion.

3. Attitude Alignment

3.1. Attitude Coarse Alignment. Since the velocity of the DVL is in the b frame, the aim here is to convert (2) into the b frame too. Equation (2) can be written as

$$\begin{aligned} \dot{\mathbf{v}}^n &= \frac{d(C_b^n \mathbf{v}^b)}{dt} = \dot{C}_b^n \mathbf{v}^b + C_b^n \dot{\mathbf{v}}^b \\ &= C_b^n \mathbf{f}^b - (2\omega_{ie}^n + \omega_{en}^n) \times \mathbf{v}^n + \mathbf{g}^n. \end{aligned} \quad (6)$$

Substituting (1) into (6) yields

$$\dot{\mathbf{v}}^b = \mathbf{f}^b - C_n^b (2\omega_{ie}^n + \omega_{en}^n) \times (C_b^n \mathbf{v}^b) - \omega_{nb}^b \times \mathbf{v}^b + C_n^b \mathbf{g}^n. \quad (7)$$

This is the complete form of the velocity update equation in the b frame. In fact, $C_n^b (2\omega_{ie}^n + \omega_{en}^n) \times (C_b^n \mathbf{v}^b)$ could be omitted for the reason that the AUVs would not sail in a very high speed (commonly several meters per second) and it is the coarse alignment stage. The simplified form of (7) for the coarse alignment is

$$\dot{\mathbf{v}}^b = \mathbf{f}^b - \omega_{nb}^b \times \mathbf{v}^b + C_n^b \mathbf{g}^n. \quad (8)$$

Equation (1) is the traditional attitude update equation. Nowadays, the attitude update is separated using the DCM product chain rule as [12]

$$C_{b(t+\Delta t)}^{n(t+\Delta t)} = C_{n(t)}^{n(t+\Delta t)} C_{b(t)}^{n(t)} C_{b(t+\Delta t)}^{b(t)}. \quad (9)$$

The update method for $C_{n(t)}^{n(t+\Delta t)}$ and $C_{b(t+\Delta t)}^{b(t)}$ is also proposed in [12]. If this rule is applied in each update cycle, the attitude update equation will be

$$C_{b(t)}^{n(t)} = C_{n(0)}^{n(t)} C_{b(0)}^{n(0)} C_{b(t)}^{b(0)}. \quad (10)$$

The update equations for $C_{n(t)}^{n(0)}$ and $C_{b(t)}^{b(0)}$ are [11]

$$\dot{C}_{n(t)}^{n(0)} = C_{n(t)}^{n(0)} \omega_{in}^n \times, \quad (11)$$

$$\dot{C}_{b(t)}^{b(0)} = C_{b(t)}^{b(0)} \omega_{ib}^b \times, \quad (12)$$

where $\omega_{in}^n = \omega_{ie}^n + \omega_{en}^n \approx \omega_{ie}^n$ for the reason that ω_{en}^n is unknown but minor in alignment stage. If $C_{b(0)}^{n(0)}$ is identified, $C_{b(t)}^{n(t)}$ on every time step is known. The procedure identifying $C_{b(0)}^{n(0)}$ is as follows.

Substitute (10) into (8) and multiply $C_{n(t)}^{n(0)}$ on each part

$$C_{b(t)}^{b(0)} \dot{\mathbf{v}}^b - C_{b(t)}^{b(0)} \mathbf{f}^b + C_{b(t)}^{b(0)} \omega_{nb}^b \times \mathbf{v}^b = C_{n(0)}^{b(0)} C_{n(t)}^{n(0)} \mathbf{g}^n. \quad (13)$$

Integrating (13) on both sides

$$\boldsymbol{\alpha}(t) = C_{n(0)}^{b(0)} \boldsymbol{\beta}(t), \quad (14)$$

where

$$\boldsymbol{\alpha}(t) = \int_0^t C_{b(t)}^{b(0)} \dot{\mathbf{v}}^b + C_{b(t)}^{b(0)} \omega_{nb}^b \times \mathbf{v}^b - C_{b(t)}^{b(0)} \mathbf{f}^b dt, \quad (15)$$

$$\boldsymbol{\beta}(t) = \int_0^t C_{n(t)}^{n(0)} \mathbf{g}^n dt. \quad (16)$$

The initial attitude matrix $C_{b(0)}^{n(0)}$ can be uniquely solved by the optimization-based method described in [16, 17] to minimize

$$L(C_{n(0)}^{b(0)}) = \sum (\boldsymbol{\alpha}(t_N) - C_{n(0)}^{b(0)} \boldsymbol{\beta}(t_N))^2. \quad (17)$$

So the key is how to calculate $\boldsymbol{\alpha}(t)$ and $\boldsymbol{\beta}(t)$ with different integral time.

3.2. Calculation Scheme for $\boldsymbol{\alpha}(t)$, $\boldsymbol{\beta}(t)$. Calculation scheme for $\boldsymbol{\alpha}(t)$, $\boldsymbol{\beta}(t)$ includes two parts: integration strategy and update scheme.

3.2.1. Integration Strategy. If $\dot{\mathbf{v}}^b$, ω_{ib}^b , and $\mathbf{f}^b$ were known, (13) could be the model for calculating $C_{b(0)}^{n(0)}$. This cannot be achieved by real sensors. The DVL's output is the velocity in the b frame, the gyroscope's output is the incremental angle over the sample time, and the accelerometer's output is the incremental velocity over the sample time. This is the reason that the integral forms are needed.

For (15), notice that

$$\begin{aligned} & \int_0^t C_{b(t)}^{b(0)} \dot{\mathbf{v}}^b + C_{b(t)}^{b(0)} \omega_{nb}^b \times \mathbf{v}^b dt \\ &= C_{b(t)}^{b(0)} \mathbf{v}^b \Big|_0^t - \int_0^t C_{b(t)}^{b(0)} \dot{\mathbf{v}}^b - C_{b(t)}^{b(0)} \omega_{nb}^b \times \mathbf{v}^b dt. \end{aligned} \quad (18)$$

Substitute (4) and (12) into (18) and omit the $C_{b(t)}^{b(0)} C_{n(t)}^{n(0)} (\omega_{ie}^n + \omega_{en}^n) \times \mathbf{v}^b$ part for the same reason as the AUVs would not voyage in a very high speed

$$\int_0^t C_{b(t)}^{b(0)} \dot{\mathbf{v}}^b + C_{b(t)}^{b(0)} \omega_{nb}^b \times \mathbf{v}^b dt \approx C_{b(t)}^{b(0)} \mathbf{v}^b \Big|_0^t. \quad (19)$$

The calculation methods for the other part of (15) and the whole equation (16) are already expressed in [11] as

$$\begin{aligned} \int_0^t C_{b(t)}^{b(0)} \mathbf{f}^b dt &= \sum_{k=0}^{N-1} C_{b(t_k)}^{b(0)} \int_{t_k}^{t_{k+1}} C_{b(t)}^{b(t_k)} \mathbf{f}^b dt \\ &\approx \sum_{k=0}^{N-1} C_{b(t_k)}^{b(0)} \int_{t_k}^{t_{k+1}} \left[\mathbf{I} + \left(\int_{t_k}^t \omega_{ib}^b \times d\tau \right) \right] \mathbf{f}^b dt \\ &= \sum_{k=0}^{N-1} C_{b(t_k)}^{b(0)} \left[\Delta \mathbf{v}_1 + \Delta \mathbf{v}_2 + \frac{1}{2} (\Delta \boldsymbol{\theta}_1 + \Delta \boldsymbol{\theta}_2) \right. \\ &\quad \times (\Delta \mathbf{v}_1 + \Delta \mathbf{v}_2) \\ &\quad \left. + \frac{2}{3} (\Delta \boldsymbol{\theta}_1 \times \Delta \mathbf{v}_2 + \Delta \mathbf{v}_1 \times \Delta \boldsymbol{\theta}_2) \right], \end{aligned} \quad (20)$$

$$\begin{aligned} \boldsymbol{\beta}(t_N) &= \int_0^t C_{n(t)}^{n(0)} \mathbf{g}^n dt = \sum_{k=0}^{N-1} C_{n(t_k)}^{n(0)} \int_{t_k}^{t_{k+1}} C_{n(t)}^{n(t_k)} \mathbf{g}^n dt \\ &\approx \sum_{k=0}^{N-1} C_{n(t_k)}^{n(0)} \int_{t_k}^{t_{k+1}} (\mathbf{I} + (t - t_k) \omega_{in}^n \times) \mathbf{g}^n dt \\ &= \sum_{k=0}^{N-1} C_{n(t_k)}^{n(0)} \left(T \mathbf{I} + \frac{T^2}{2} \omega_{in}^n \times \right) \mathbf{g}^n \\ &\approx \sum_{k=0}^{N-1} C_{n(t_k)}^{n(0)} \left(T \mathbf{I} + \frac{T^2}{2} \omega_{ie}^n \times \right) \mathbf{g}^n. \end{aligned} \quad (21)$$

So

$$\begin{aligned} \boldsymbol{\alpha}(t_N) &\approx C_{b(t_N)}^{b(0)} \mathbf{v}^b(t_N) - \mathbf{v}^b(0) \\ &\quad - \sum_{k=0}^{N-1} C_{b(t_k)}^{b(0)} \left[\Delta \mathbf{v}_1 + \Delta \mathbf{v}_2 + \frac{1}{2} (\Delta \boldsymbol{\theta}_1 + \Delta \boldsymbol{\theta}_2) \right. \\ &\quad \times (\Delta \mathbf{v}_1 + \Delta \mathbf{v}_2) \\ &\quad \left. + \frac{2}{3} (\Delta \boldsymbol{\theta}_1 \times \Delta \mathbf{v}_2 + \Delta \mathbf{v}_1 \times \Delta \boldsymbol{\theta}_2) \right], \end{aligned} \quad (22)$$

where the angular rate and the specific force are assumed in linear forms within a short time as

$$\begin{aligned} \omega_{ib}^b &= (t - t_k) \mathbf{c}_\omega + \mathbf{d}_\omega, \\ \mathbf{f}^b &= (t - t_k) \mathbf{c}_f + \mathbf{d}_f. \end{aligned} \quad (23)$$

The $\mathbf{c}_f$, $\mathbf{d}_f$, $\mathbf{c}_\omega$, and $\mathbf{d}_\omega$ are the coefficient vectors as

$$\begin{aligned} \mathbf{c}_f &= \frac{4(\Delta \mathbf{v}_2 - \Delta \mathbf{v}_1)}{T^2}, \\ \mathbf{d}_f &= \frac{3\Delta \mathbf{v}_1 - \Delta \mathbf{v}_2}{T}, \end{aligned} \quad (24)$$

$$\begin{aligned} \mathbf{c}_\omega &= \frac{4(\Delta\theta_2 - \Delta\theta_1)}{T^2}, \\ \mathbf{d}_\omega &= \frac{3\Delta\theta_1 - \Delta\theta_2}{T}, \end{aligned} \quad (25)$$

where T is the time period of one update interval $[t_k, t_{k+1}]$, $k = 0, 1, 2, \dots, N-1$, with $t_k = kT$, and $\Delta\mathbf{v}_1$, $\Delta\mathbf{v}_2$ and $\Delta\theta_1$, $\Delta\theta_2$ are the incremental velocity and angle collected from the accelerometers and gyroscopes.

3.2.2. Update Scheme. The update rate of the sensors is not the same. The update rate of the gyroscopes and accelerometers can go up from 1 to 2 kHz, while the update rate of the DVL is only several Hz and the update may be interrupted due to the depth of the water or some other reasons. The update scheme for $\alpha(t)$, $\beta(t)$ should be suitable for these conditions.

For $\alpha(t)$ expressed in (22), the lower part is related to the gyroscopes and accelerometers. In order to obtain an accurate result, this part should be updated at these sensors' update rate. A higher update rate could reduce the effect such as coning motion [18]. For the whole equation, the update rate is based on the DVL. It means that the lower part should be updated in high speed and when the DVL's data is available the whole equation is updated.

For $\beta(t)$ expressed in (21), the update rate is the same as the gyroscopes and accelerometers. And the output rate is kept the same as $\alpha(t)$ since the algorithm needs the data at the same time point. Consider

$$\begin{aligned} \alpha(t) &\rightarrow \alpha_{\text{DVL}}(t_M), \\ \beta(t) &\rightarrow \beta_{\text{DVL}}(t_M), \end{aligned} \quad (26)$$

where $M = 0, 1, 2, \dots, t_M$ means the time point that the DVL is available. The least number of M is decided by the sensor error level.

Owing to the low and unsteady output rate of the DVL, the $\mathbf{v}^b(0)$ is almost not obtainable. This part can be counteracted as

$$\alpha_{\text{DVL}}(t_M) - \alpha_{\text{DVL}}(t_0) = C_{n(0)}^{b(0)} (\beta_{\text{DVL}}(t_M) - \beta_{\text{DVL}}(t_0)). \quad (27)$$

This is the practical equation for the coarse alignment.

3.3. Attitude Fine Alignment. According to the sensor error level, about 200 samples ($M \approx 200$) of DVL data are needed for the coarse alignment. Since the update rate of the DVL is several Hz, the coarse alignment could cost about 60 s. After 60 s of the attitude coarse alignment, the pitch and roll would attain about 0.1 deg accuracy and the heading error would be around 10 deg. The fine alignment starts immediately after it. The goal is to improve the attitude accuracy especially the heading accuracy.

For the fine alignment, most methods use the SINS attitude and velocity error models as the state transition equation for the data fusion algorithms such as the Kalman

filter (KF), the extended KF (EKF), and the UKF. The UKF for its worthy performance in nonlinear estimation [19, 20] is especially used for nonlinear misalignment angles. Here is the review of the UKF based attitude fine alignment with nonlinear misalignment angles model [7, 21, 22].

The navigation algorithm is described in (1), (2), and (3). The state of the UKF is

$$\mathbf{x} = [\delta\mathbf{v}^n \quad \boldsymbol{\theta} \quad \delta\boldsymbol{\omega}_{ib}^b \quad \delta\mathbf{f}^b]^T. \quad (28)$$

And the large misalignment angles model is

$$\begin{aligned} \delta\dot{\mathbf{v}}^n &= (\mathbf{I} - C_{n'}^n) C_b^{n'} \hat{\mathbf{f}}^b - (2\delta\boldsymbol{\omega}_{ie}^n + \delta\boldsymbol{\omega}_{en}^n) \times \hat{\mathbf{v}}^n \\ &\quad - (2\hat{\boldsymbol{\omega}}_{ie}^n + \hat{\boldsymbol{\omega}}_{en}^n) \times \delta\mathbf{v}^n + C_{n'}^n C_b^{n'} \delta\mathbf{f}^b, \\ \dot{\boldsymbol{\theta}} &= (\mathbf{I} - C_n^{n'}) \hat{\boldsymbol{\omega}}_{in}^n + \delta\boldsymbol{\omega}_{in}^n - C_b^{n'} \delta\boldsymbol{\omega}_{ib}^b, \\ \delta\dot{\boldsymbol{\omega}}_{ib}^b &= \mathbf{0} + \boldsymbol{\varepsilon}_\omega, \quad \delta\dot{\mathbf{f}}^b = \mathbf{0} + \boldsymbol{\varepsilon}_f, \end{aligned} \quad (29)$$

where $\hat{y}$ is the real data equal to the ideal data y plus the error δy , $C_b^{n'}$ is the DCM from the b frame to the n' frame, $\boldsymbol{\varepsilon}_\omega$ and $\boldsymbol{\varepsilon}_f$ are the noise of the gyroscopes and accelerometers, and $C_n^{n'}$ is the DCM from the n frame to the n' frame. Assume that the n frame to the n' frame is completed through three noninterchangeable rotations θ_z , θ_x , and θ_y . Each rotation can be expressed as

$$\begin{aligned} C_{\theta_x} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & \sin\theta_x \\ 0 & -\sin\theta_x & \cos\theta_x \end{bmatrix}, \\ C_{\theta_y} &= \begin{bmatrix} \cos\theta_y & 0 & -\sin\theta_y \\ 0 & 1 & 0 \\ \sin\theta_y & 0 & \cos\theta_y \end{bmatrix}, \\ C_{\theta_z} &= \begin{bmatrix} \cos\theta_z & \sin\theta_z & 0 \\ -\sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (30)$$

The total DCM is

$$C_n^{n'} = C_{\theta_y} C_{\theta_x} C_{\theta_z}. \quad (31)$$

The measurement model is

$$\begin{aligned} \mathbf{z} &= \hat{\mathbf{v}}^n - C_{n'}^n C_b^{n'} \hat{\mathbf{v}}_{\text{DVL}}^b = \mathbf{v}^n + \delta\mathbf{v}^n \\ &\quad - C_{n'}^n C_b^{n'} (\mathbf{v}_{\text{DVL}}^b + \delta\mathbf{v}_{\text{DVL}}^b) \\ &= \mathbf{v}^n - \mathbf{v}_{\text{DVL}}^n + \delta\mathbf{v}^n - \delta\mathbf{v}_{\text{DVL}}^n = \delta\mathbf{v}^n - \delta\mathbf{v}_{\text{DVL}}^n, \end{aligned} \quad (32)$$

where $\delta\mathbf{v}_{\text{DVL}}^n$ is the noise of the DVL.

Then according to (29) and (32), the nonlinear state and measurement equations are

$$\begin{aligned} \dot{\mathbf{x}} &= \begin{bmatrix} \delta \dot{\mathbf{v}}^n \\ \dot{\boldsymbol{\theta}} \\ \delta \dot{\boldsymbol{\omega}}_{ib}^b \\ \delta \dot{\mathbf{f}}^b \end{bmatrix} = f(\mathbf{x}) + \mathbf{w} \\ &= \begin{bmatrix} (\mathbf{I} - C_{n'}^n) C_b^{n'} \hat{\mathbf{f}}^b - (2\delta \boldsymbol{\omega}_{ie}^n + \delta \boldsymbol{\omega}_{en}^n) \times \hat{\mathbf{v}}^n - (2\hat{\boldsymbol{\omega}}_{ie}^n + \hat{\boldsymbol{\omega}}_{en}^n) \times \delta \mathbf{v}^n + C_{n'}^n C_b^{n'} \delta \mathbf{f}^b \\ (\mathbf{I} - C_n^{n'}) \hat{\boldsymbol{\omega}}_{in}^n + \delta \boldsymbol{\omega}_{in}^n - C_b^{n'} \delta \boldsymbol{\omega}_{ib}^b \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \boldsymbol{\varepsilon}_\omega \\ \boldsymbol{\varepsilon}_f \end{bmatrix}, \\ \mathbf{z} &= h(\mathbf{x}) + \mathbf{u} = [\delta \mathbf{v}^n] + [-\delta \mathbf{v}_{\text{DVL}}^n], \end{aligned} \quad (33)$$

where $\mathbf{w} = [\mathbf{0}_{3 \times 1} \quad \mathbf{0}_{3 \times 1} \quad \boldsymbol{\varepsilon}_\omega \quad \boldsymbol{\varepsilon}_f]$ is the state noise and $\mathbf{u} = [-\delta \mathbf{v}_{\text{DVL}}^n]$ is the measurement noise.

The fine alignment's duty is to estimate $C_{n'}^n$ to improve the accuracy of C_b^n . Notice that $C_{b(0)}^n$, $C_{b(t)}^{b(0)}$, and $C_{n(t)}^{n(0)}$ can still be updated in fine alignment. It means that a more precise $C_{b(0)}^{n(0)}$ is realized too according to the inverse of (10) in the fine alignment stage as

$$C_{b(0)}^{n(0)} = C_{n(t)}^{n(0)} C_{b(t)}^{n(t)} C_{b(0)}^{b(t)}. \quad (34)$$

This result is of great importance in the position alignment.

4. Position Alignment

4.1. Position Alignment Algorithm. The original position update equation is described in (3). This position update equation can also be performed between the e frame and the b frame as

$$C_{n(t)}^{n(0)} \mathbf{R}_c^{-1} \dot{\mathbf{p}}(t) = C_{b(0)}^{n(0)} C_{b(t)}^{b(0)} \mathbf{v}^b(t). \quad (35)$$

In order to get the position, (35) should be integrated on each side first. The integral form of the left term is

$$\begin{aligned} \int_0^{t_{\text{end}}} C_{n(t)}^{n(0)} \mathbf{R}_c^{-1} \dot{\mathbf{p}}(t) dt \\ = C_{n(t)}^{n(0)} \mathbf{R}_c^{-1} \mathbf{p}(t) \Big|_0^{t_{\text{end}}} - \int_0^{t_{\text{end}}} \dot{C}_{n(t)}^{n(0)} \mathbf{R}_c^{-1} \mathbf{p}(t) dt, \end{aligned} \quad (36)$$

where t_{end} is the time point where the attitude alignment ends. Substitute (11) into (36):

$$\begin{aligned} \int_0^{t_{\text{end}}} C_{n(t)}^{n(0)} \mathbf{R}_c^{-1} \dot{\mathbf{p}}(t) dt \\ = C_{n(t)}^{n(0)} \mathbf{R}_c^{-1} \mathbf{p}(t) \Big|_0^{t_{\text{end}}} - \int_0^{t_{\text{end}}} C_{n(t)}^{n(0)} \boldsymbol{\omega}_{in}^n \times \mathbf{R}_c^{-1} \mathbf{p}(t) dt. \end{aligned} \quad (37)$$

The integral part of (37) can be converted as

$$\begin{aligned} \int_0^{t_{\text{end}}} C_{n(t)}^{n(0)} \boldsymbol{\omega}_{in}^n \times \mathbf{R}_c^{-1} \mathbf{p}(t) dt \\ = \sum_{k=0}^{N-1} C_{n(t_k)}^{n(0)} \int_{t_k}^{t_{k+1}} C_{n(t)}^{n(t_k)} \boldsymbol{\omega}_{in}^n \times \mathbf{R}_c^{-1} \mathbf{p}(t) dt, \end{aligned} \quad (38)$$

where $t_k = t_{\text{end}}$, when $k = N$. Practically, the effort of the position change is really small in the integration. It could be regarded as constant as

$$\mathbf{p}(t) = \mathbf{p}(t_{\text{end}}), \quad (39)$$

where $\mathbf{p}(t_{\text{end}})$ is the position which needs alignment. Substitute (39) into the integral part of (38):

$$\begin{aligned} \int_{t_k}^{t_{k+1}} C_{n(t)}^{n(t_k)} \boldsymbol{\omega}_{in}^n \times \mathbf{R}_c^{-1} \mathbf{p}(t) dt \\ \approx \int_{t_k}^{t_{k+1}} (\mathbf{I} + (t - t_k) \boldsymbol{\omega}_{in}^n \times) \boldsymbol{\omega}_{in}^n \times \mathbf{R}_c^{-1} \mathbf{p}(t_{\text{end}}) dt \\ = \boldsymbol{\omega}_{in}^n \times \mathbf{R}_c^{-1} \mathbf{p}(t_{\text{end}}) T + \boldsymbol{\omega}_{in}^n \times \boldsymbol{\omega}_{in}^n \times \mathbf{R}_c^{-1} \mathbf{p}(t_{\text{end}}) \frac{T^2}{2}. \end{aligned} \quad (40)$$

Therefore the computed form of (36) is

$$\begin{aligned} \int_0^{t_{\text{end}}} C_{n(t)}^{n(0)} \mathbf{R}_c^{-1} \dot{\mathbf{p}} dt \\ \approx C_{n(t_{\text{end}})}^{n(0)} \mathbf{R}_c^{-1} \mathbf{p}(t_{\text{end}}) - \mathbf{R}_c^{-1} \mathbf{p}(0) \\ - \sum_{k=0}^{N-1} C_{n(t_k)}^{n(0)} \left\{ \boldsymbol{\omega}_{in}^n \times \mathbf{R}_c^{-1} \mathbf{p}(t_{\text{end}}) T \right. \\ \left. + \boldsymbol{\omega}_{in}^n \times \boldsymbol{\omega}_{in}^n \times \mathbf{R}_c^{-1} \mathbf{p}(t_{\text{end}}) \frac{T^2}{2} \right\}. \end{aligned} \quad (41)$$

For (41), $\boldsymbol{\omega}_{in}^n \times \boldsymbol{\omega}_{in}^n \times \mathbf{R}_c^{-1} \mathbf{p}(t_{\text{end}}) T^2/2$, which is a second-order term, is much smaller than $\boldsymbol{\omega}_{in}^n \times \mathbf{R}_c^{-1} \mathbf{p}(t_{\text{end}}) T$ with $T < 1$. So

this part could be omitted. Then, notice that $T\omega_{in}^n \times \approx C_{n(t_{k+1})}^{n(t_k)} - \mathbf{I}$; (41) can be simplified as

$$\begin{aligned}
& \int_0^{t_{\text{end}}} C_{n(t)}^{n(0)} \mathbf{R}_c^{-1} \dot{\mathbf{p}} dt \\
& \approx C_{n(t_{\text{end}})}^{n(0)} \mathbf{R}_c^{-1} \mathbf{p}(t_{\text{end}}) - \mathbf{R}_c^{-1} \mathbf{p}(0) \\
& \quad - \sum_{k=0}^{N-1} C_{n(t_k)}^{n(0)} \left\{ \left(C_{n(t_{k+1})}^{n(t_k)} - \mathbf{I} \right) \mathbf{R}_c^{-1} \mathbf{p}(t_{\text{end}}) \right\} \\
& \approx C_{n(t_{\text{end}})}^{n(0)} \mathbf{R}_c^{-1} \mathbf{p}(t_{\text{end}}) - \mathbf{R}_c^{-1} \mathbf{p}(0) \\
& \quad - \left[C_{n(t_{\text{end}})}^{n(0)} \mathbf{R}_c^{-1} \mathbf{p}(t_{\text{end}}) - \mathbf{R}_c^{-1} \mathbf{p}(t_{\text{end}}) \right] \\
& = \mathbf{R}_c^{-1} \mathbf{p}(t_{\text{end}}) - \mathbf{R}_c^{-1} \mathbf{p}(0).
\end{aligned} \tag{42}$$

The integral form of the right term of (35) is

$$\begin{aligned}
& \int_0^{t_{\text{end}}} C_{b(0)}^{n(0)} C_{b(t)}^{b(0)} \mathbf{v}^b dt \\
& = C_{b(0)}^{n(0)} \sum_{j=0}^{M-1} C_{b(t_j)}^{b(0)} \int_{t_j}^{t_{j+1}} C_{b(t)}^{b(t_j)} \mathbf{v}^b dt \\
& \approx C_{b(0)}^{n(0)} \sum_{j=0}^{M-1} C_{b(t_j)}^{b(0)} \int_{t_j}^{t_{j+1}} \left(\mathbf{I} + \left(\int_{t_j}^t \omega_{ib}^b d\tau \right) \times \right) \mathbf{v}^b dt,
\end{aligned} \tag{43}$$

where $t_j = t_{\text{end}}$, when $j = M$. Similarly, $\mathbf{v}^b$ can be approximated linearly as

$$\mathbf{v}^b(t) = \mathbf{v}^b(t_j) + \frac{t - t_j}{T_{\text{DVL}}} (\mathbf{v}^b(t_{j+1}) - \mathbf{v}^b(t_j)), \tag{44}$$

where T_{DVL} is the duration of two DVL samples. Substitute (23), (25), and (44) into (43):

$$\begin{aligned}
& \int_0^{t_{\text{end}}} C_{b(0)}^{n(0)} C_{b(t)}^{b(0)} \mathbf{v}^b dt \\
& \approx C_{b(0)}^{n(0)} \sum_{j=0}^{M-1} C_{b(t_j)}^{b(0)} \left[\frac{T_{\text{DVL}}}{2} (\mathbf{v}^b(t_{k+1}) + \mathbf{v}^b(t_k)) + \frac{1}{3} \Delta \theta_1 \right. \\
& \quad \times T_{\text{DVL}} \mathbf{v}^b(t_k) + \left(\frac{1}{2} \Delta \theta_1 + \frac{1}{6} \Delta \theta_2 \right) \\
& \quad \times T_{\text{DVL}} \mathbf{v}^b(t_{k+1}) \left. \right].
\end{aligned} \tag{45}$$

According to (35), (42) equals (45). It follows that

$$\begin{aligned}
& \mathbf{p}(t_{\text{end}}) = \mathbf{p}(0) \\
& \quad + \mathbf{R}_c C_{b(0)}^{n(0)} \sum_{j=0}^{M-1} C_{b(t_j)}^{b(0)} \left[\frac{T_{\text{DVL}}}{2} (\mathbf{v}^b(t_{k+1}) + \mathbf{v}^b(t_k)) \right. \\
& \quad \quad + \frac{1}{3} \Delta \theta_1 \times T_{\text{DVL}} \mathbf{v}^b(t_k) \\
& \quad \quad + \left(\frac{1}{2} \Delta \theta_1 + \frac{1}{6} \Delta \theta_2 \right) \\
& \quad \quad \times T_{\text{DVL}} \mathbf{v}^b(t_{k+1}) \left. \right].
\end{aligned} \tag{46}$$

For (46), all the data can be calculated in real-time except $C_{b(0)}^{n(0)}$. It means that when the initial attitude matrix $C_{b(0)}^{n(0)}$ is obtained, the position could immediately be updated. As described in Section 3, $C_{b(0)}^{n(0)}$ can be gathered by (27) through the attitude coarse alignment, and then the accuracy can be improved by (34) through the whole attitude fine alignment.

Also, (39) can be treated as

$$\mathbf{p}(t) = \mathbf{p}(0). \tag{47}$$

And then, (41) goes to

$$\begin{aligned}
& \int_0^{t_{\text{end}}} C_{n(t)}^{n(0)} \mathbf{R}_c^{-1} \dot{\mathbf{p}} dt \\
& \approx C_{n(t_{\text{end}})}^{n(0)} \mathbf{R}_c^{-1} \mathbf{p}(t_{\text{end}}) - \mathbf{R}_c^{-1} \mathbf{p}(0) \\
& \quad - \sum_{k=0}^{N-1} C_{n(t_k)}^{n(0)} \left\{ \omega_{in}^n \times \mathbf{R}_c^{-1} \mathbf{p}(0) T + \omega_{in}^n \right. \\
& \quad \quad \times \omega_{in}^n \times \mathbf{R}_c^{-1} \mathbf{p}(0) \frac{T^2}{2} \left. \right\}.
\end{aligned} \tag{48}$$

And by making (48) be equal to (45), another solution comes out as

$$\begin{aligned}
& C_{n(t_{\text{end}})}^{n(0)} \mathbf{R}_c^{-1} \mathbf{p}(t_{\text{end}}) \\
& \approx C_{b(0)}^{n(0)} \sum_{k=0}^{N-1} C_{b(t_k)}^{b(0)} \left[\frac{T_{\text{DVL}}}{2} (\mathbf{v}^b(t_{k+1}) + \mathbf{v}^b(t_k)) + \frac{1}{3} \Delta \theta_1 \right. \\
& \quad \times T_{\text{DVL}} \mathbf{v}^b(t_k) + \left(\frac{1}{2} \Delta \theta_1 + \frac{1}{6} \Delta \theta_2 \right) \\
& \quad \times T_{\text{DVL}} \mathbf{v}^b(t_{k+1}) \left. \right] \\
& \quad + \mathbf{R}_c^{-1} \mathbf{p}(0) + \sum_{k=0}^{N-1} C_{n(t_k)}^{n(0)} \omega_{in}^n \times \mathbf{R}_c^{-1} \mathbf{p}(0) T \\
& \quad + \sum_{k=0}^{N-1} C_{n(t_k)}^{n(0)} \omega_{in}^n \times \omega_{in}^n \times \mathbf{R}_c^{-1} \mathbf{p}(0) \frac{T^2}{2}.
\end{aligned} \tag{49}$$

And we can also let (41) be equal to (45) to get another solution as

$$\begin{aligned}
& C_{n(t_{\text{end}})}^{n(0)} \mathbf{R}_c^{-1} \mathbf{p}(t_{\text{end}}) \\
& \approx C_{b(0)}^{n(0)} \sum_{k=0}^{N-1} C_{b(t_k)}^{b(0)} \left[\frac{T_{\text{DVL}}}{2} (\mathbf{v}^b(t_{k+1}) + \mathbf{v}^b(t_k)) + \frac{1}{3} \Delta \boldsymbol{\theta}_1 \right. \\
& \quad \times T_{\text{DVL}} \mathbf{v}^b(t_k) + \left(\frac{1}{2} \Delta \boldsymbol{\theta}_1 + \frac{1}{6} \Delta \boldsymbol{\theta}_2 \right) \\
& \quad \left. \times T_{\text{DVL}} \mathbf{v}^b(t_{k+1}) \right] \\
& + \mathbf{R}_c^{-1} \mathbf{p}(0) + \sum_{k=0}^{N-1} C_{n(t_k)}^{n(0)} \boldsymbol{\omega}_{in}^n \times \mathbf{R}_c^{-1} \mathbf{p}(t_{\text{end}}) T \\
& + \sum_{k=0}^{N-1} C_{n(t_k)}^{n(0)} \boldsymbol{\omega}_{in}^n \times \boldsymbol{\omega}_{in}^n \times \mathbf{R}_c^{-1} \mathbf{p}(t_{\text{end}}) \frac{T^2}{2}.
\end{aligned} \tag{50}$$

Although the expression of (49) and (50) is a little complicated, the alignment results are almost the same by the three different approximations. This will be approved in the simulation section.

4.2. In-Motion Alignment Summarization. A summarization of in-motion alignment for DVL aided SINS is proposed to give a better understanding of the whole algorithm.

Step 1. Set $C_{n(0)}^{n(0)} = C_{b(0)}^{b(0)} = \mathbf{I}$, $\mathbf{p}(0)$ as the beginning position coordinate.

Step 2. Update $C_{n(t)}^{n(0)}$, $C_{b(t)}^{b(0)}$, (21), and the lower part of (22) when the data of the inertial sensors is available.

Step 3. Update the whole equation of (22) when the DVL's data is available.

Step 4. Compute the initial attitude matrix $C_{b(0)}^{n(0)}$ based on (27). Compute the attitude matrix $C_{b(t)}^{n(t)}$ by (10).

Step 5. Update the position via (46).

Step 6. Go to Step 2 until the attitude coarse alignment is complete.

Step 7. Use the UKF based fine alignment algorithm. The navigation algorithm is described in (1), (2), and (3). The state and measurement equations are described in (33).

Step 8. Update $C_{n(t)}^{n(0)}$, $C_{b(t)}^{b(0)}$ when the data of inertial sensors is available.

Step 9. Compute a more accurate $C_{b(0)}^{n(0)}$ according to (34).

Step 10. Update the position via (46).

FIGURE 1: Average attitude errors of four types of trajectories.

Step 11. Go to Step 7 until the attitude fine alignment is complete.

It is clear from these steps that the alignment can work on a real-time SINS, and no backtrack is needed.

5. Simulation Results

The attitude coarse alignment and position alignment algorithms are examined in this section through simulation with four types of trajectories.

The simulation trajectories' base latitude and longitude are 30 deg and 120 deg. The pitch, roll, and heading are sinusoid waves with random amplitude and frequency within 10 deg and 1 Hz. Other parameters are listed below:

trajectory 1 (Tr1): zero velocity condition;

trajectory 2 (Tr2): random sinusoid wave condition with max velocity about 1 m/s;

trajectory 3 (Tr3): line condition with random velocity around 1.4 m/s;

trajectory 4 (Tr4): line condition with random velocity around 14 m/s.

The first simulation uses error-free gyroscopes, accelerometers, and DVL data. Each type runs 50 times. Figure 1 shows the average attitude errors. The pitch and roll errors reach 0.003 deg within 20 s for all four trajectories. And the heading errors reduce to 0.05 deg for Tr1 and Tr2 after 30 s

FIGURE 2: Average heading errors of Tr4 and Tr4 with the omitted parts kept (Tr4_no).

FIGURE 3: Average attitude errors of Tr3 and Tr3 with sensor errors (Tr3_SE). Dashed lines are the 3σ envelopes.

alignment. And the heading errors grow to 0.1 deg for Tr3 and 1 deg for Tr4. This is mainly because that, with the velocity increasing, the omitted parts are growing too. And these parts are affecting the heading alignment accuracy.

Figure 2 gives the results where the omitted parts are included. It is clear that the heading accuracy increased. We hope that the alignment algorithm is perfect with all parts being considered. But the omitted parts are unknown since these parts need velocity in the n frame. And for the real sensors, the sensor errors would affect the accuracy much more than the omitted parts.

Figure 3 illustrates the alignment errors with sensor errors (gyroscopes drift: 0.2 deg/h, gyroscopes noise:

FIGURE 4: Average position errors of four types of trajectories with three different position approximate methods.

0.9 deg/h/√Hz, accelerometers bias: 1×10^{-4} g, accelerometers noise: 2×10^{-4} g/√Hz, DVL bias: 1.5% of voyage velocity, and DVL noise: standard variance 0.02 m/s). The pitch and roll errors are almost the same compared with the error-free sensors, while the heading error's 3σ envelope is 10 times larger than the error-free sensors. And it is also larger than the heading error of Tr4. So, the largest attitude coarse alignment error source is the sensor errors. Meanwhile, owing to the difference of the sensor error level, the convergence speed of Tr3_SE is slower than Tr3. But the heading error of Tr3_SE can still be less than 20 deg after 60 s. It is enough for the fine alignment stage.

For the position alignment, three different position approximate methods are applied in Section 4.

Approximation 1 (Ap1): regard $\mathbf{p}(t)$ as $\mathbf{p}(t_{\text{end}})$, and $\mathbf{a}_{in}^n \times \mathbf{a}_{in}^n \times \mathbf{R}_c^{-1} \mathbf{p}(t_{\text{end}}) T^2 / 2$ part is omitted.

Approximation 2 (Ap 2): regard $\mathbf{p}(t)$ as $\mathbf{p}(0)$.

Approximation 3 (Ap 3): regard $\mathbf{p}(t)$ as $\mathbf{p}(t_{\text{end}})$.

Figure 4 displays the average position errors of these three different approximations. The simulation still runs 50 times with four types of trajectories.

As shown in Figure 4, the position errors of three different approximations are almost the same. It means that the approximation for (38) is reasonable. The position errors have

FIGURE 5: Average position errors of Tr3 with Ap1 and Ap1 with sensor errors (Ap1_SE). Dashed lines are the 3σ envelopes.

FIGURE 6: The towed body with an INS, a DVL, and a GPS.

a significant increase and decrease process in the first 20 s; this is mainly because the accuracy of the initial attitude DCM $C_{b(0)}^{n(0)}$ is increasing as described in Section 3.3. The max position errors of Tr4 are larger than the other trajectories for the reason that the velocity is at least 10 times larger and the attitude error is 10 times larger too. And the accuracy of Ap1 is especially a little bit higher than Ap2 and Ap3. So, Ap1 is applied in the next simulation and experiments.

Figure 5 demonstrates the average position errors with sensor errors. The position errors significantly increase. Still the largest error source is the sensor errors.

6. Experiment Results

6.1. Experiment Setup. The in-motion alignment experiment data was collected in the Thousand Island Lake in Zhejiang Province in China by using a towed body shown in Figure 6. The sensors used for the experiment are listed below.

- (1) An inertial navigation system developed by Zhejiang University with four fiber optics gyroscopes (bias instability: 0.2 deg/h) and four quartz accelerometers (bias: 1×10^{-4} g). The data output rate is 500 Hz.
- (2) A 600 kHz Phased Array ExplorerDVL produced by RD Instruments. Owing to the export restrictions to China, the accuracy is limited to about 1.5% of voyage. The data output rate is 4~5 Hz.

FIGURE 7: The route of the towed body.

- (3) An OEMV-1G GPS receiver produced by Novatel. The position accuracy is about 3 m. The velocity accuracy is about 0.1 m/s. The data output rate is 5 Hz.

The GPS is fixed on the top of the towed body, while the INS and DVL are installed under water for about 0.3 m. The time delay and the lever arm of the sensors are compensated.

6.2. In-Motion Alignment Results. The trajectory of the towed body is shown in Figure 7. Four alignments were done in the whole voyage. Each test includes 60 s coarse alignment, 240 s fine alignment, and 300 s navigation. The GPS aided INS was already aligned before the experiment and was working at navigation mode to produce standard position, velocity, and attitude. It was set as a standard system for the results comparison.

The attitude alignment results are presented in Figures 8, 9, and 10. The heading error reaches 20 deg within maximum 52 s (alignment 2) and minimum 24 s (alignment 3) and oscillates in this range during the coarse alignment. The heading error at 60 s is -5.2 deg, 13.7 deg, 4.8 deg, and 5.8 deg. This is consistent with the simulation results shown in Figure 3. The convergent speed of the coarse alignment is fast, but owing to the lack of the optimization approach method which cannot handle the noises of the sensors separately, the error just oscillates. This is why the fine alignment is required. With heading error of this level, it costs minimum 152 s (alignment 1) and maximum 225 s (alignment 3) to reach 1 deg of accuracy in the fine alignment stage. The errors of pitch and roll are less than 0.05 deg after the coarse alignment and keep in this range. So we believe that the total alignment could finish within 300 s.

The position alignment results are presented in Figures 11 and 12. The towed body voyaged at the speed of about 1 m/s. The total voyage distance for each test was about 600 m. The

FIGURE 8: The heading error of the alignment. (a) Overview. (b) Details of the coarse alignment. (c) Details of the fine alignment.

FIGURE 9: The pitch error of the alignment.

error of the four alignments is 3 m, 16 m, 7 m, and 5 m after the coarse alignment. This is consistent with the simulation results shown in Figure 5. The error reduced to 1 m, 4 m, 2 m, and 3 m after the fine alignment. It is less than 1.5% of the voyage distance. The position error grows slowly after the alignment. This is another evidence to prove that the alignment is effective.

Figures 13 and 14 choose alignment 4 to show a comparison between the algorithm proposed in this paper (IMA1) and the previous UKF based method reviewed in Section 3.3 (IMA2). As described in [7], the UKF based method could solve the attitude alignment problem if the initial heading

FIGURE 10: The roll error of the alignments.

FIGURE 11: Position error of the alignments.

FIGURE 12: Position error/voyage distance of the alignments.

error is within about 100 deg. But if without the coarse alignment process, the heading error is unknown. So, the initial heading errors larger (150 deg) and smaller (20 deg) than 100 deg are all considered for IMA2 in the comparison.

If the initial heading error is small (IMA2-20), the heading error after the alignment is at the same level as IMA1, while if the initial heading error is large (IMA2-150), the heading may appear to be larger than 1 deg after the alignment without the attitude coarse alignment process.

FIGURE 13: The comparison of the heading error of IMA1 and IMA2.

FIGURE 14: The comparison of the position error of IMA1 and IMA2.

Also the position error is larger than 6% of the voyage distance which is 3 times larger than IMA1.

7. Conclusions

The accuracy and applicability of SINS largely depended on the precision and swiftness of the alignment. It is a difficult problem to align an in-motion SINS with underwater applications due to the lack of the GPS which could provide velocity in the n frame and position in the e frame. The alignment problem for underwater applications is not only the attitude alignment but also the position alignment.

This paper proposed a complete alignment solution for DVL aided SINS in-motion by using the integration form of the velocity update equation in body frame to give a rough initial angle for the UKF based fine alignment and using the integration form of position update equation to alignment of the position. Simulation and on-lake tests were carried out to verify the alignment. The results show that the heading could reach around 10 deg accuracy and the pitch and roll could be aligned up to 0.05 deg during the coarse alignment.

The heading could reach 1 deg accuracy in 240 s using UKF based fine alignment with heading error of this level. And the position error could achieve 1.5% of the voyage distance.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported by The Principal Fund of Zhejiang University.

References

- [1] W. Alameda Jr., "Seadevil—a totally integrated inertial navigation system (INS) solution," in *Proceedings of the Underwater Intervention Symposium*, 2002.
- [2] B. Jalving, K. Gade, K. Svartveit, A. Willumsen, and R. Sørhagen, "DVL velocity aiding in the HUGIN 1000 integrated inertial navigation system," *Modeling, Identification and Control*, vol. 25, no. 4, pp. 223–235, 2004.
- [3] Ø. Hegrenæs and E. Berglund, "Doppler water-track aided inertial navigation for autonomous underwater vehicle," in *Proceedings of the IEEE Bremen: Balancing Technology with Future Needs (OCEANS '09)*, May 2009.
- [4] S. Hong, M. H. Lee, H. S. Kwon, and H. H. Chun, "A car test for the estimation of GPS/INS alignment errors," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 3, pp. 208–217, 2004.
- [5] S. Han and J. Wang, "A novel initial alignment scheme for low-cost INS aided by GPS for land vehicle applications," *Journal of Navigation*, vol. 63, no. 4, pp. 663–680, 2010.
- [6] D. Titterton and J. Weston, *Strapdown Inertial Navigation Technology*, The Institution of Electrical Engineers, Reston, Va, USA, 2nd edition, 2004.
- [7] W. Li, J. Wang, L. Lu, and W. Wu, "A novel scheme for DVL-aided SINS in-motion alignment using UKF techniques," *Sensors*, vol. 13, no. 1, pp. 1046–1063, 2013.
- [8] J. Cheng and F. J. de Wan, "A fast initial alignment method for strapdown inertial navigation system on stationary base," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, no. 4, pp. 1501–1505, 1996.
- [9] L. Schimelevich and R. Naor, "New approach to coarse alignment," in *Proceedings of the IEEE Position Location and Navigation Symposium (PLANS '96)*, pp. 324–327, April 1996.
- [10] P. M. G. Silson, "Coarse alignment of a ship's strapdown inertial attitude reference system using velocity loci," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 6, pp. 1930–1941, 2011.
- [11] Y. Wu and X. Pan, "Velocity/position integration formula part I: application to in-flight coarse alignment," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 2, pp. 1006–1023, 2013.
- [12] P. G. Savage, "Strapdown inertial navigation integration algorithm design part I: attitude algorithms," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 1, pp. 19–28, 1998.
- [13] P. G. Savage, "Strapdown inertial navigation integration algorithm design part 2: velocity and position algorithms," *Journal*

- of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 208–221, 1998.
- [14] P. G. Savage, *Strapdown Analytics*, Strapdown Associates, Maple Plain, Minn, USA, 2000.
  - [15] Y. Wu and X. Pan, “Velocity/position integration formula part II: application to strapdown inertial navigation computation,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 2, pp. 1024–1034, 2013.
  - [16] M. D. Shuster and S. D. Oh, “Three-axis attitude determination from vector observations,” *Journal of Guidance, Control, and Dynamics*, vol. 4, no. 1, pp. 70–77, 1981.
  - [17] I. Y. Bar-Itzhack, “REQUEST: a recursive QUEST algorithm for sequential attitude determination,” *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 5, pp. 1034–1038, 1996.
  - [18] P. G. Savage, “Coning algorithm design by explicit frequency shaping,” *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 4, pp. 1123–1132, 2010.
  - [19] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, “New approach for filtering nonlinear systems,” in *Proceedings of the American Control Conference*, pp. 1628–1632, June 1995.
  - [20] E. A. Wan and R. van der Merwe, “The unscented Kalman filter for nonlinear estimation,” in *Proceedings of the IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium (AS-SPCC '00)*, pp. 153–158, 2000.
  - [21] X. Kong, E. M. Nebot, and H. Durrant-Whyte, “Development of a non-linear psi-angle model for large misalignment errors and its application in INS alignment and calibration,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '99)*, pp. 1430–1435, May 1999.
  - [22] H. S. Hong, J. G. Lee, and C. G. Park, “Performance improvement of in-flight alignment for autonomous vehicle under large initial heading error,” *IEE Proceedings: Radar, Sonar and Navigation*, vol. 151, no. 1, pp. 57–62, 2004.