IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 27, 2019, accepted September 16, 2019, date of publication September 20, 2019, date of current version October 8, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2942638

A Method Converting Cone Into Sculling Algorithm for Strapdown Inertial Navigation System

PAN JIANG, YA ZHANG, QIANG HAO, SHIWEI FAN, AND DINGJIE XU
Institute of Navigation Instrument, Harbin Institute of Technology, Harbin 150001, China

Corresponding author: Ya Zhang (yazhang@hit.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51709068 and Grant 61573117, and in part by the Postdoctoral Foundation of Heilongjiang Province Government under Grant LBH-Z17094 and Grant LBH-Z17091.

ABSTRACT The cone error compensation algorithm and the sculling error compensation algorithm are two important components of the inertial navigation algorithm. At present, scholars have done a lot of research on cone error compensation, and have proposed many cone error compensation algorithms, but there are few studies on the compensation of the sculling error. So for this problem, this paper proposes a new operation method, which can convert the cone error compensation algorithm into the corresponding sculling error compensation algorithm with simple calculation. Taking four classical cone error algorithms as examples, we give the conversion process and conversion results. The conversion results are exactly the same as the derivation error compensation algorithms given by the derivation, which proves the effectiveness of the proposed method. The method avoids the complicated derivation process of the traditional sculling error compensation algorithm, significantly reduces the amount of calculation, and has strong engineering practical significance.

INDEX TERMS Cone algorithm, sculling algorithm, transformation method, strapdown inertial navigation system.

I. INTRODUCTION

Attitude solving and velocity solving are two important processes of inertial navigation algorithm. However, due to the noncommuttativity of rigid body limited rotation, cone effect and sculling effect will be caused, and the corresponding attitude and velocity are introduced. Reducing navigation errors is an important issue in the field of strapdown inertial navigation.

The accuracy of the attitude algorithm will directly affect the accuracy of the force integral transformation, which will affect the velocity and position resolution accuracy. Since the concept of strapdown inertial navigation has been proposed, researchers in the navigation field have focused more on the research of attitude algorithms [1]–[5]. Miller first proposed the concept of optimizing the coefficient of the cone algorithm in a pure conical environment [6]. Ignagni then gave nine conic correction algorithms and derived two important properties about cone integration and angular increment under pure cone conditions [7]. Jiang and Lin introduced the sum of the angular increments of the previous cone compensation period as a correction term into the cone compensation algorithm in the current period to improve the performance of the cone algorithm [8], [9]. Musoff and Murphy proposed a method to improve the performance of the algorithm by using data interpolation techniques [10]. Panov derived the optimized attitude solving algorithm according to the Miller method under certain generalized angular precession [11]. Inspired by Miller, Gusinsky *et al.* gave a new derivation method for the attitude solving algorithm, which only needs to know the analytical expression of the angular velocity of the carrier [12], [13]. Savage gave a classic two-speed attitude update algorithm, the medium speed algorithm uses the direction cosine matrix update method, and the high speed algorithm is a simplified equivalent rotation vector algorithm [5], [14]. Song proposed an extended cone error compensation algorithm, which can effectively suppress the algorithm error in high dynamic environment [15]. On the basis of Song, Wang first considered the third order of Picard in cone error compensation [16]. Theoretical analysis shows that the algorithm has higher precision in high dynamic environment.

The associate editor coordinating the review of this manuscript and approving it for publication was Donghyun Kim.

In contrast, there are very few scholars who study the sculling error compensation algorithm. Only a few people such as Ignagni, Savage, Song, and WANG have studied the sculling error compensation algorithm [15]–[19]. Roscoe gave a simple mathematical formula in 2000 to convert the cone algorithm into a corresponding sculling algorithm [20]. This idea is great and can significantly simplify the design of the sculling algorithm. And it is no problem to verify with the existing algorithm at that time. However, with the deepening of research, more accurate cone compensation algorithms have been proposed. Theoretical studies have shown that the mathematical formula proposed by Roscoe is not suitable for the cone error compensation algorithm considering the third-order term of the Picard series.

Therefore, in view of the above problems, this paper proposes a new operation method, which can convert the cone error compensation algorithm into the corresponding sculling error compensation algorithm with simple operation. We prove the general equivalence between the sculling algorithm and the cone algorithm, and theoretically prove the effectiveness of the conversion method. In addition, we take the existing cone error compensation algorithms as examples, and give its conversion process and conversion results. The conversion results are exactly the same as the derivation error compensation algorithms given by the derivation, which proves the effectiveness of the method. This manuscript is organized as follows: Section 2 demonstrates the general equivalence between the sculling algorithm and the cone algorithm; Section 3 proposes a new method for converting the cone error compensation algorithm into sculling error compensation algorithm; Section 4 uses the existing cone compensation algorithm and the sculling compensation algorithm to verify the transformation method proposed in this paper; Section 5 carries out simulation experiment verification; Section 6 summarizes some conclusions and major contributions.

II. CONING AND SCULLING ERROR COMPENSATION ALGORITHMS

A. CONING ALGORITHMS

The differential equation of the equivalent rotation vector can be expressed as:

$$\dot{\boldsymbol{\phi}}=\boldsymbol{\omega}+\frac{1}{2}\boldsymbol{\phi}\times\boldsymbol{\omega}+\frac{1}{\phi^2}\left[1-\frac{\phi\sin\phi}{2\left(1-\cos\phi\right)}\right]\boldsymbol{\phi}\times(\boldsymbol{\phi}\times\boldsymbol{\omega}) \quad (1)$$

where, $\boldsymbol{\phi}$ denotes the equivalent rotation vector and its amplitude is $\phi=(\boldsymbol{\phi}\cdot\boldsymbol{\phi})^{1/2}$; $\boldsymbol{\omega}$ is the angular rate vector obtained by the gyro measurement; operator $\times$ represents the cross product between the vectors.

Performing series expansion on the trigonometric function in the equation, there are:

$$\begin{aligned}&\frac{\phi\sin\phi}{2\left(1-\cos\phi\right)}\\&=\frac{\phi\cdot 2\sin\frac{\phi}{2}\cos\frac{\phi}{2}}{2\cdot 2\sin^2\frac{\phi}{2}}=\frac{\phi}{2}\cot\frac{\phi}{2}\\&=\frac{\phi}{2}\left[\frac{2}{\phi}-\frac{1}{3}\frac{\phi}{2}-\frac{1}{45}\left(\frac{\phi}{2}\right)^3-\frac{2}{945}\left(\frac{\phi}{2}\right)^5-\frac{1}{4725}\left(\frac{\phi}{2}\right)^7-\cdots\right]\\&=1-\frac{\phi^2}{12}-\frac{\phi^4}{720}-\frac{\phi^6}{30240}-\frac{\phi^8}{1209600}-\cdots\end{aligned} \quad (2)$$

Since that the noncommutativity error is $\frac{1}{2}\boldsymbol{\phi}\times\boldsymbol{\omega}+\frac{1}{\phi^2}\left[1-\frac{\phi\sin\phi}{2(1-\cos\phi)}\right]\boldsymbol{\phi}\times(\boldsymbol{\phi}\times\boldsymbol{\omega})$, the noncommutativity error is at least 5 orders, if we take the 4th order approximation $\frac{\phi\sin\phi}{2(1-\cos\phi)}=1-\frac{\phi^2}{12}-\frac{\phi^4}{720}$. For the three sample attitude compensation algorithm, theoretical derivation shows that 5th order error has little effect. Therefore, this article only take the approximation $\frac{\phi\sin\phi}{2(1-\cos\phi)}=1-\frac{\phi^2}{12}$. And the approximate equation commonly used in engineering is obtained:

$$\dot{\boldsymbol{\phi}}=\boldsymbol{\omega}+\frac{1}{2}\boldsymbol{\phi}\times\boldsymbol{\omega}+\frac{1}{12}\boldsymbol{\phi}\times(\boldsymbol{\phi}\times\boldsymbol{\omega}) \quad (3)$$

$\frac{1}{2}\boldsymbol{\phi}\times\boldsymbol{\omega}+\frac{1}{12}\boldsymbol{\phi}\times(\boldsymbol{\phi}\times\boldsymbol{\omega})$ represents the noncommutativity error. The core of the cone error compensation algorithm is to improve the accuracy of the cone compensation algorithm in various environments by designing a numerical integration algorithm that realizes the noncommutativity error.

If only the third order term of the Picard series is considered, the above formula can be simplified to:

$$\begin{aligned}\dot{\boldsymbol{\phi}}=\boldsymbol{\omega}+\frac{1}{2}\left[\boldsymbol{\Delta\theta}+\frac{1}{2}\int_0^{\tau}(\boldsymbol{\Delta\theta}\times\boldsymbol{\omega})\,dt\right]\times\boldsymbol{\omega}\\+\frac{1}{12}\left[\boldsymbol{\Delta\theta}\times\right](\boldsymbol{\Delta\theta}\times\boldsymbol{\omega})\end{aligned} \quad (4)$$

where, $\boldsymbol{\Delta\theta}$ is the angular increment that can be expressed as $\boldsymbol{\Delta\theta}=\int_0^{\tau}\boldsymbol{\omega}d\tau$.

B. SCULLING ALGORITHMS

The velocity differential equation in the body coordinate system is:

$$\boldsymbol{\Delta}\dot{\boldsymbol{v}}_{\boldsymbol{SF}}^{\boldsymbol{B_{m-1}}}(t)=\boldsymbol{C}_{B(t)}^{B_{m-1}}\boldsymbol{a_{SF}} \quad (5)$$

where, $B_{m-1}$is the body coordinate system at the end of the $m-1$ attitude update period; $B_{(t)}$ is the body coordinate system at time t in the attitude update period from t_{m-1} to t_m; a_{SF} is specific force acceleration vector.

$\boldsymbol{C}_{B(t)}^{B_{m-1}}$ represents the direction cosine matrix from B_t to B_{m-1} transformation, which can be expressed as:

$$\boldsymbol{C}_{B(t)}^{B_{m-1}}=\boldsymbol{I}+\frac{\sin\phi}{\phi}\left[\boldsymbol{\phi}\times\right]+\frac{(1-\cos\phi)}{\phi^2}[\boldsymbol{\phi}\times]^2 \quad (6)$$

The velocity increment introduced by substituting Eq.6 into Eq.5 and introducing the specific force acceleration over time t_{m-1} to t_m interval is:

$$\begin{aligned}\boldsymbol{\Delta v}_{\boldsymbol{SF}}^{\boldsymbol{B_{m-1}}}(t)=&\int_{t_{m-1}}^{t}\boldsymbol{C}_{B(t)}^{B_{m-1}}\boldsymbol{a_{SF}}d\tau=\boldsymbol{v}(t)\\&+\int_{t_{m-1}}^{t}\frac{\sin\phi}{\phi}\left[\boldsymbol{\phi}\times\right]\boldsymbol{a_{SF}}d\tau\\&+\int_{t_{m-1}}^{t}\frac{(1-\cos\phi)}{\phi^2}[\boldsymbol{\phi}\times]^2\boldsymbol{a_{SF}}d\tau\end{aligned} \quad (7)$$

The algorithm input is replaced by the angular rate integral and the specific force acceleration integral into the equivalent rotation vector and the velocity translation vector, the following is obtained:

$$\Delta \boldsymbol{v}_{SF}^{B_{m-1}}(t) = \boldsymbol{\eta}(t) + \frac{(1-\cos\phi)}{\phi^2}[\boldsymbol{\phi}\times]\boldsymbol{\eta}(t) + \frac{1}{\phi^2}\left(1-\frac{\sin\phi}{\phi}\right)[\boldsymbol{\phi}\times]^2\boldsymbol{\eta}(t) = \boldsymbol{F}\boldsymbol{\eta}(t) \quad (8)$$

wherein, $\boldsymbol{F} = \boldsymbol{I} + \frac{(1-\cos\phi)}{\phi^2}[\boldsymbol{\phi}\times] + \frac{1}{\phi^2}\left(1-\frac{\sin\phi}{\phi}\right)[\boldsymbol{\phi}\times]^2$.

For Eq.8, the derivative of time t is:

$$\Delta \dot{\boldsymbol{v}}_{\boldsymbol{SF}}^{\boldsymbol{B_{m-1}}}(t) = \dot{\boldsymbol{F}}\boldsymbol{\eta}(t) + \boldsymbol{F}\dot{\boldsymbol{\eta}}(t) = \boldsymbol{C}_{B(t)}^{B_{m-1}}\boldsymbol{a}_{SF} \quad (9)$$

Then we can get:

$$\dot{\boldsymbol{\eta}}(t) = \boldsymbol{F}^{-1}\left[\boldsymbol{C}_{B(t)}^{B_{m-1}}\boldsymbol{a}_{SF} - \dot{\boldsymbol{F}}\boldsymbol{\eta}(t)\right] \quad (10)$$

Expanded and organized, the differential equation of velocity translation vector is:

$$\begin{aligned}\dot{\boldsymbol{\eta}}(t) = {} & \boldsymbol{a}_{SF} + \frac{1}{2}\left(\boldsymbol{\phi}\times\boldsymbol{a}_{SF} - \dot{\boldsymbol{\phi}}\times\boldsymbol{\eta}\right) \\ & + f_1\boldsymbol{\phi}\times\left(\boldsymbol{\phi}\times\boldsymbol{a}_{SF} - \dot{\boldsymbol{\phi}}\times\boldsymbol{\eta}\right) + f_2\left(\boldsymbol{\phi}\times\dot{\boldsymbol{\phi}}\right)\times\boldsymbol{\eta} \\ & - \frac{1}{2}f_2\boldsymbol{\phi}\times\left[\left(\boldsymbol{\phi}\times\dot{\boldsymbol{\phi}}\right)\times\boldsymbol{\eta}\right] + f_3\phi^2\dot{\boldsymbol{\phi}}\times\boldsymbol{\eta} \\ & + f_4\boldsymbol{\phi}\times\left[\boldsymbol{\phi}\times\left(\dot{\boldsymbol{\phi}}\times\boldsymbol{\eta}\right)\right] + \frac{1}{2}f_2\boldsymbol{\phi}\cdot\boldsymbol{\omega}\boldsymbol{\phi}\times\boldsymbol{\eta} \\ & + f_5\boldsymbol{\phi}\times\left\{\boldsymbol{\phi}\times\left[\left(\boldsymbol{\phi}\times\dot{\boldsymbol{\phi}}\right)\times\boldsymbol{\eta}\right]\right\} \\ & + f_6\phi^2\boldsymbol{\phi}\times\left(\dot{\boldsymbol{\phi}}\times\boldsymbol{\eta}\right) - f_6\boldsymbol{\phi}\cdot\boldsymbol{\omega}\boldsymbol{\phi}\times\left(\boldsymbol{\phi}\times\boldsymbol{\eta}\right) \quad (11)\end{aligned}$$

wherein, $f_1 = \frac{1}{\phi^2}\left[1-\frac{\phi\sin\phi}{2(1-\cos\phi)}\right]$, $f_2 = \frac{1}{\phi^2}\left(1-\frac{\sin\phi}{\phi}\right)$, $f_3 = \frac{1}{\phi^2}\left[\frac{1}{2}-\frac{1-\cos\phi}{\phi^2}\right]$, $f_4 = \frac{1}{\phi^2}\left(1-\frac{1}{2}\frac{\sin\phi}{\phi}-\frac{1-\cos\phi}{\phi^2}\right)$, $f_5 = f_1f_2$, and $f_6 = \frac{1}{2\phi^4}\left[5+\cos\phi-\frac{3\phi\sin\phi}{1-\cos\phi}\right]$.

Similar to the approximation principle in Eq.3, take the first-order Taylor expansion approximation of f, that is $f_1 \approx \frac{1}{12}, f_2 \approx \frac{1}{6}, f_3 \approx \frac{1}{24}, f_4 \approx \frac{1}{8}$. Then the fourth-order Picard solution components of the velocity translation vector can be expressed as:

$$\begin{aligned}\dot{\boldsymbol{\eta}}(t) &= \Delta\dot{\boldsymbol{\eta}}_{\boldsymbol{1}} + \Delta\dot{\boldsymbol{\eta}}_{\boldsymbol{2}} + \Delta\dot{\boldsymbol{\eta}}_{\boldsymbol{3}} + \Delta\dot{\boldsymbol{\eta}}_{\boldsymbol{4}}\cdots \\ \Delta\dot{\boldsymbol{\eta}}_{\boldsymbol{1}} &= \boldsymbol{a}_{\boldsymbol{SF}}, \quad \Delta\boldsymbol{\eta}_{\boldsymbol{1}} = \Delta\boldsymbol{v} \\ \Delta\dot{\boldsymbol{\eta}}_{\boldsymbol{2}} &= \frac{1}{2}\left(\Delta\boldsymbol{\theta}\times\boldsymbol{a}_{\boldsymbol{SF}} - \boldsymbol{\omega}\times\Delta\boldsymbol{v}\right) \\ \Delta\dot{\boldsymbol{\eta}}_{\boldsymbol{3}} &= \frac{1}{4}\left(\int_{t_{m-1}}^{t_m}\Delta\boldsymbol{\theta}\times\boldsymbol{\omega}dt\right)\times\boldsymbol{a}_{\boldsymbol{SF}} \\ &\quad - \frac{1}{4}\boldsymbol{\omega}\times\left(\int_{t_{m-1}}^{t_m}\left(\Delta\boldsymbol{\theta}\times\boldsymbol{a}_{\boldsymbol{SF}} - \boldsymbol{\omega}\times\Delta\boldsymbol{v}\right)dt\right) \\ &\quad + \frac{1}{12}\Delta\boldsymbol{\theta}\times\left(\Delta\boldsymbol{\theta}\times\boldsymbol{a}_{\boldsymbol{SF}} - \boldsymbol{\omega}\times\Delta\boldsymbol{v}\right) - \frac{1}{12}\left(\Delta\boldsymbol{\theta}\times\boldsymbol{\omega}\right)\times\Delta\boldsymbol{v}\end{aligned}$$

$$\Delta\dot{\boldsymbol{\eta}}_{\boldsymbol{4}} = \frac{1}{2}\left\{\begin{array}{l}\int_{t_{m-1}}^{t_m}\left[\begin{array}{l}\frac{1}{4}\int_{t_{m-1}}^{t_m}\Delta\boldsymbol{\theta}\times\boldsymbol{\omega}dt\times\boldsymbol{\omega} \\ +\frac{1}{12}\Delta\boldsymbol{\theta}\times\left(\Delta\boldsymbol{\theta}\times\boldsymbol{\omega}\right)\end{array}\right]dt\times\boldsymbol{a}_{SF} \\ -\boldsymbol{\omega}\times\Delta\boldsymbol{\eta}_3 + \frac{1}{3}\Delta\boldsymbol{\theta}\times\Delta\dot{\boldsymbol{\eta}}_3 \\ -\frac{1}{3}\left(\begin{array}{l}\frac{1}{4}\int_{t_{m-1}}^{t_m}\Delta\boldsymbol{\theta}\times\boldsymbol{\omega}dt\times\boldsymbol{\omega} \\ +\frac{1}{12}\Delta\boldsymbol{\theta}\times\left(\Delta\boldsymbol{\theta}\times\boldsymbol{\omega}\right)\end{array}\right)\times\boldsymbol{v}\end{array}\right\} + \frac{1}{6}\left[\frac{1}{2}\int_{t_{m-1}}^{t_m}\Delta\boldsymbol{\theta}\times\boldsymbol{\omega}dt\times\Delta\dot{\boldsymbol{\eta}}_2 - \frac{1}{2}\left(\Delta\boldsymbol{\theta}\times\boldsymbol{\omega}\right)\times\Delta\boldsymbol{\eta}_{\boldsymbol{2}}\right] \quad (12)$$

where, $\boldsymbol{v}$ is the velocity increment that can be expressed as $\boldsymbol{v} = \int_0^\tau \boldsymbol{a}_{\boldsymbol{SF}}d\tau$. $\Delta\dot{\boldsymbol{\eta}}_{\boldsymbol{2}}$ is to consider the sculling error of the second order term of the Picard series, $\Delta\dot{\boldsymbol{\eta}}_3$ is the sculling error considering the third order term of the Picard series, and the corresponding $\Delta\dot{\boldsymbol{\eta}}_4$ is the sculling error considering the fourth order term of the Picard series.

III. A METHOD FOR TRANSFORMING CONING ERROR COMPENSATION INTO SCULLING ERROR COMPENSATION

Through theoretical research and analysis, we propose a new operation method ξ, which means to convert the cone error compensation algorithm into the sculling error compensation algorithm. Its operational theorems are as follows:

$$\begin{aligned}\xi(\boldsymbol{\omega}) &= \boldsymbol{a}_{\boldsymbol{SF}}, \quad \xi(\Delta\boldsymbol{\theta}) = \Delta\boldsymbol{v} \\ \xi(a\boldsymbol{x}) &= a\xi(\boldsymbol{x}) \\ \xi(\boldsymbol{x}\pm\boldsymbol{y}) &= \xi(\boldsymbol{x})\pm\xi(\boldsymbol{y}) \\ \xi(\boldsymbol{x}\times\boldsymbol{y}) &= \xi(\boldsymbol{x})\times\boldsymbol{y} + \boldsymbol{x}\times\xi(\boldsymbol{y})\end{aligned} \quad (13)$$

where, a is constant, $\boldsymbol{x}$, $\boldsymbol{y}$ are polynomials associated with $\boldsymbol{\omega}$ and $\Delta\boldsymbol{\theta}$.

Next we theoretically prove the effectiveness of the conversion algorithm. Take the equivalent differential equation based on the Picard series expansion proposed by Savage as an example. The Picard series solution of Eq.3 can be expressed as follows:

$$\begin{aligned}\Delta\dot{\boldsymbol{\phi}}_{\boldsymbol{1}} &= \boldsymbol{\omega}, \ \Delta\boldsymbol{\phi}_{\boldsymbol{1}} = \Delta\boldsymbol{\theta} \\ \Delta\dot{\boldsymbol{\phi}}_{\boldsymbol{2}} &= \frac{1}{2}\Delta\boldsymbol{\theta}\times\boldsymbol{\omega} \\ \Delta\dot{\boldsymbol{\phi}}_{\boldsymbol{3}} &= \frac{1}{2}\Delta\boldsymbol{\phi}_{\boldsymbol{2}}\times\boldsymbol{\omega} + \frac{1}{6}\Delta\boldsymbol{\theta}\times\Delta\dot{\boldsymbol{\phi}}_{\boldsymbol{2}} \\ \Delta\dot{\boldsymbol{\phi}}_{\boldsymbol{4}} &= \frac{1}{2}\left(\Delta\boldsymbol{\phi}_{\boldsymbol{3}}\times\boldsymbol{\omega} + \frac{1}{3}\Delta\boldsymbol{\theta}\times\Delta\dot{\boldsymbol{\phi}}_3\right) + \frac{1}{6}\Delta\boldsymbol{\phi}_{\boldsymbol{2}}\times\Delta\dot{\boldsymbol{\phi}}_{\boldsymbol{2}}\end{aligned} \quad (14)$$

wherein, $\Delta\boldsymbol{\phi}_{\boldsymbol{i}} = \int_0^\tau \Delta\dot{\boldsymbol{\phi}}_i d\tau$.

Applying the transformation method proposed in this paper to Eq.14, we can get

$$\begin{aligned}\xi\left(\Delta\dot{\boldsymbol{\phi}}_{\boldsymbol{1}}\right) &= \xi(\boldsymbol{\omega}) = \boldsymbol{a}_{\boldsymbol{SF}} = \Delta\dot{\boldsymbol{\eta}}_{\boldsymbol{1}} \\ \xi\left(\Delta\dot{\boldsymbol{\phi}}_{\boldsymbol{2}}\right) &= \frac{1}{2}\xi(\Delta\boldsymbol{\theta})\times\boldsymbol{\omega} + \frac{1}{2}\Delta\boldsymbol{\theta}\times\xi(\boldsymbol{\omega})\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2}\left(\Delta\boldsymbol{v}\times\boldsymbol{\omega}+\Delta\boldsymbol{\theta}\times\boldsymbol{a}_{SF}\right)=\Delta\dot{\boldsymbol{\eta}}_{\mathbf{2}}\\
\xi\left(\Delta\dot{\boldsymbol{\phi}}_{\mathbf{3}}\right)&=\frac{1}{2}\xi\left(\Delta\boldsymbol{\phi}_{\mathbf{2}}\right)\times\boldsymbol{\omega}+\frac{1}{2}\Delta\boldsymbol{\phi}_{\mathbf{2}}\times\xi\left(\boldsymbol{\omega}\right)+\frac{1}{6}\xi\left(\Delta\boldsymbol{\theta}\right)\times\Delta\dot{\boldsymbol{\phi}}_{2}\\
&\quad+\frac{1}{6}\Delta\boldsymbol{\theta}\times\xi\left(\Delta\dot{\boldsymbol{\phi}}_{\mathbf{2}}\right)=\frac{1}{2}\left(\Delta\boldsymbol{\eta}_{\mathbf{2}}\times\boldsymbol{\omega}+\Delta\boldsymbol{\phi}_{\mathbf{2}}\times\boldsymbol{a}_{SF}\right)\\
&\quad+\frac{1}{6}\left(\Delta\boldsymbol{v}\times\Delta\dot{\boldsymbol{\phi}}_{2}+\Delta\boldsymbol{\theta}\times\Delta\dot{\boldsymbol{\eta}}_{2}\right)=\Delta\dot{\boldsymbol{\eta}}_{3}\\
\xi\left(\Delta\dot{\boldsymbol{\phi}}_{\mathbf{4}}\right)&=\xi\left(\frac{1}{2}\Delta\boldsymbol{\phi}_{\mathbf{3}}\times\boldsymbol{\omega}\right)+\xi\left(\frac{1}{6}\Delta\boldsymbol{\theta}\times\Delta\dot{\boldsymbol{\phi}}_{3}\right)\\
&\quad+\xi\left(\frac{1}{6}\Delta\boldsymbol{\phi}_{\mathbf{2}}\times\Delta\dot{\boldsymbol{\phi}}_{2}\right)=\frac{1}{2}\xi\left(\Delta\boldsymbol{\phi}_{\mathbf{3}}\right)\times\boldsymbol{\omega}\\
&\quad+\frac{1}{2}\Delta\boldsymbol{\phi}_{\mathbf{3}}\times\xi\left(\boldsymbol{\omega}\right)\\
&\quad+\frac{1}{6}\xi\left(\Delta\boldsymbol{\theta}\right)\times\Delta\dot{\boldsymbol{\phi}}_{3}+\frac{1}{6}\Delta\boldsymbol{\theta}\times\xi\left(\Delta\dot{\boldsymbol{\phi}}_{\mathbf{3}}\right)\\
&\quad+\frac{1}{6}\xi\left(\Delta\boldsymbol{\phi}_{\mathbf{2}}\right)\times\Delta\dot{\boldsymbol{\phi}}_{2}\\
&\quad+\frac{1}{6}\Delta\boldsymbol{\phi}_{\mathbf{2}}\times\xi\left(\Delta\dot{\boldsymbol{\phi}}_{\mathbf{2}}\right)=\frac{1}{2}\left(\Delta\boldsymbol{\eta}_{\mathbf{3}}\times\boldsymbol{\omega}+\Delta\boldsymbol{\phi}_{\mathbf{3}}\times\boldsymbol{a}_{SF}\right)\\
&\quad+\frac{1}{6}\left(\Delta\boldsymbol{v}\times\Delta\dot{\boldsymbol{\phi}}_{3}+\Delta\boldsymbol{\theta}\times\Delta\dot{\boldsymbol{\eta}}_{3}\right)\\
&\quad+\frac{1}{6}\left(\Delta\boldsymbol{\eta}_{\mathbf{2}}\times\Delta\dot{\boldsymbol{\phi}}_{2}+\Delta\boldsymbol{\phi}_{\mathbf{2}}\times\Delta\dot{\boldsymbol{\eta}}_{2}\right)=\Delta\dot{\boldsymbol{\eta}}_{4}\qquad(15)
\end{aligned}
$$

where, $\Delta\boldsymbol{\eta}_{\boldsymbol{i}}=\int_0^{\tau}\Delta\dot{\boldsymbol{\eta}}_{i}d\tau$.

The Picard series expansion of the velocity translation vector obtained by the transformation method proposed in this paper is exactly the same as that derived by the derivation [21], which proves the general equivalence between the sculling algorithm and the cone algorithm. That is to say, as long as we know a cone error compensation algorithm, we can get the corresponding sculling error compensation algorithm by utilizing the transformation method proposed in this paper. This greatly simplifies the design of the sculling algorithm.

IV. EFFECTIVENESS VERIFICATION OF THE CONVERSION METHOD

In the previous section, the general equivalence between the sculling algorithm and the cone algorithm is proved, and a new conversion method is given. The validity of the conversion method is theoretically proved. In this section, we use the four most representative cone error compensation algorithms as examples to give the conversion process and the conversion results and compare them with the theoretically derived sculling error compensation algorithm to further verify the conversion method effectiveness.

A. SAVAGE'S ALGORITHMS

The coning algorithm derived in Savage is [14]:

$$
\boldsymbol{\phi}_{\boldsymbol{c}}=\sum_{l=1}^{L}\frac{1}{2}\left(\boldsymbol{\theta}_{\boldsymbol{l-1}}+\frac{1}{6}\Delta\boldsymbol{\theta}_{\boldsymbol{l-1}}\right)\times\Delta\boldsymbol{\theta}_{\boldsymbol{l}}\qquad(16)
$$

where, l is the computer period index within a computer m period, defined to be zero at time t_{m-1}; L is the number of l periods within an m period; $\Delta\boldsymbol{\theta}_{\boldsymbol{l}}$ is $\Delta\boldsymbol{\theta}$ in Eq.4 with integral limits from t_{l-1} to t_l that can be expressed as $\Delta\boldsymbol{\theta}_{\boldsymbol{l}}=\int_{t_{l-1}}^{l}\boldsymbol{\omega}d\tau$; and $\boldsymbol{\theta}_{\boldsymbol{l}}$ can be expressed as $\boldsymbol{\theta}_{\boldsymbol{l}}=\boldsymbol{\theta}_{\boldsymbol{l-1}}+\Delta\boldsymbol{\theta}_{\boldsymbol{l}}$.

We transformed this algorithm by the method ξ proposed in this paper, and the conversion process is as follows:

$$
\begin{aligned}
\xi\left(\boldsymbol{\phi}_{\boldsymbol{c}}\right)&=\sum_{l=1}^{L}\left[\begin{array}{l}\frac{1}{2}\xi\left(\boldsymbol{\theta}_{\boldsymbol{l-1}}+\frac{1}{6}\Delta\boldsymbol{\theta}_{\boldsymbol{l-1}}\right)\times\Delta\boldsymbol{\theta}_{\boldsymbol{l}}\\+\frac{1}{2}\left(\boldsymbol{\theta}_{\boldsymbol{l-1}}+\frac{1}{6}\Delta\boldsymbol{\theta}_{\boldsymbol{l-1}}\right)\times\xi\left(\Delta\boldsymbol{\theta}_{\boldsymbol{l}}\right)\end{array}\right]\\
&=\sum_{l=1}^{L}\left[\begin{array}{l}\frac{1}{2}\left(\boldsymbol{v}_{\boldsymbol{l-1}}+\frac{1}{6}\Delta\boldsymbol{v}_{\boldsymbol{l-1}}\right)\times\Delta\boldsymbol{\theta}_{\boldsymbol{l}}\\+\frac{1}{2}\left(\boldsymbol{\theta}_{\boldsymbol{l-1}}+\frac{1}{6}\Delta\boldsymbol{\theta}_{\boldsymbol{l-1}}\right)\times\Delta\boldsymbol{v}_{\boldsymbol{l}}\end{array}\right]\qquad(17)
\end{aligned}
$$

where,$\Delta\boldsymbol{v}_{\boldsymbol{l}}$ is $\Delta\boldsymbol{v}$ in Eq.12 with integrallimits from t_{l-1} to t_l that can be expressed as $\Delta\boldsymbol{v}_{\boldsymbol{l}}=\int_{t_{l-1}}^{l}\boldsymbol{a}_{\boldsymbol{SF}}d\tau$; and $\boldsymbol{v}_{\boldsymbol{l}}$ can be expressed as $\boldsymbol{v}_{\boldsymbol{l}}=\boldsymbol{v}_{\boldsymbol{l-1}}+\Delta\boldsymbol{v}_{\boldsymbol{l}}$.

The sculling error compensation algorithm given in [18] is exactly the same as that obtained by Eq.17.

B. IGNAGNI'S ALGORITHMS

Ignagni's coning algorithm is [7]:

$$
\begin{aligned}
\boldsymbol{\phi}_{\boldsymbol{c}}&=\frac{1}{2}\sum_{l=2}^{L}\boldsymbol{\theta}_{\boldsymbol{l-1}}\times\Delta\boldsymbol{\theta}_{\boldsymbol{l}}\\
&\quad+\sum_{l=1}^{L}\left[\frac{9}{20}\Delta\boldsymbol{\theta}_{\boldsymbol{l}}\left(1\right)+\frac{27}{20}\Delta\boldsymbol{\theta}_{\boldsymbol{l}}\left(2\right)\right]\times\Delta\boldsymbol{\theta}_{\boldsymbol{l}}\left(3\right)\qquad(18)
\end{aligned}
$$

where, $\Delta\boldsymbol{\theta}_{\boldsymbol{l}}\left(i\right)$ is the angular increments vector over the *ith* interval within the l period; what's more, the sum of $\Delta\boldsymbol{\theta}_{\boldsymbol{l}}\left(1\right)$, $\Delta\boldsymbol{\theta}_{\boldsymbol{l}}\left(2\right)$ and $\Delta\boldsymbol{\theta}_{\boldsymbol{l}}\left(3\right)$ equals $\Delta\boldsymbol{\theta}_{\boldsymbol{l}}$ in Eq.4 integral limits from t_{l-1} to t_l; L, l, m, θ_l are the same defined in Eq.16

This algorithm can be transformed by the method ξ proposed in this paper, and the conversion process is as follows:

$$
\begin{aligned}
\xi\left(\boldsymbol{\phi}_{\boldsymbol{c}}\right)&=\frac{1}{2}\sum_{l=2}^{L}\xi\left(\boldsymbol{\theta}_{\boldsymbol{l-1}}\times\Delta\boldsymbol{\theta}_{\boldsymbol{l}}\right)\\
&\quad+\sum_{l=1}^{L}\xi\left[\frac{9}{20}\Delta\boldsymbol{\theta}_{\boldsymbol{l}}\left(1\right)+\frac{27}{20}\Delta\boldsymbol{\theta}_{\boldsymbol{l}}\left(2\right)\right]\times\Delta\boldsymbol{\theta}_{\boldsymbol{l}}\left(3\right)\\
&\quad+\sum_{l=1}^{L}\left[\frac{9}{20}\Delta\boldsymbol{\theta}_{\boldsymbol{l}}\left(1\right)+\frac{27}{20}\Delta\boldsymbol{\theta}_{\boldsymbol{l}}\left(2\right)\right]\times\xi\left(\Delta\boldsymbol{\theta}_{\boldsymbol{l}}\left(3\right)\right)\\
&=\frac{1}{2}\sum_{l=2}^{L}\left(\boldsymbol{v}_{\boldsymbol{l-1}}\times\Delta\boldsymbol{\theta}_{\boldsymbol{l}}\,\boldsymbol{\mathcal{C}}\,\boldsymbol{\theta}_{\boldsymbol{l-1}}\times\Delta\boldsymbol{v}_{\boldsymbol{l}}\right)\\
&\quad+\sum_{l=1}^{L}\left[\frac{9}{20}\Delta\boldsymbol{v}_{\boldsymbol{l}}\left(1\right)+\frac{27}{20}\Delta\boldsymbol{v}_{\boldsymbol{l}}\left(2\right)\right]\times\Delta\boldsymbol{\theta}_{\boldsymbol{l}}\left(3\right)\\
&\quad+\sum_{l=1}^{L}\left[\frac{9}{20}\Delta\boldsymbol{\theta}_{\boldsymbol{l}}\left(1\right)+\frac{27}{20}\Delta\boldsymbol{\theta}_{\boldsymbol{l}}\left(2\right)\right]\times\Delta\boldsymbol{v}_{\boldsymbol{l}}\left(3\right)\qquad(19)
\end{aligned}
$$

where, $\Delta\boldsymbol{v}_{\boldsymbol{l}}\left(i\right)$ is the angular increments vector over the *ith* interval within the l period; what's more, the sum of $\Delta\boldsymbol{v}_{\boldsymbol{l}}\left(1\right)$,

$\Delta\boldsymbol{v}_l$ (2) and $\Delta\boldsymbol{v}_l$ (3) equals $\Delta\boldsymbol{v}_l$ in Eq.12 integral limits from t_{l-1} to t_l; $\boldsymbol{v}_l$ are the same defined in Eq.17.

The sculling error compensation algorithm given in [17] is exactly the same as that obtained by Eq.19.

C. MILLER AND SONG'S ALGORITHMS

We use the three sub-sample algorithm as an example to verify, and the three sub-sample algorithm is expressed as follows [6]:

$$\boldsymbol{\phi}_{c3} = \Delta\boldsymbol{\theta}_1 + \Delta\boldsymbol{\theta}_2 + \Delta\boldsymbol{\theta}_3 + \frac{27}{40}\Delta\boldsymbol{\theta}_1 \times \Delta\boldsymbol{\theta}_2 + \frac{9}{20}\Delta\boldsymbol{\theta}_1 \times \Delta\boldsymbol{\theta}_3 + \frac{27}{40}\Delta\boldsymbol{\theta}_2 \times \Delta\boldsymbol{\theta}_3 \quad (20)$$

where, $\Delta\boldsymbol{\theta}_1$, $\Delta\boldsymbol{\theta}_2$, $\Delta\boldsymbol{\theta}_3$ are the angular increments in the $\left[t_k, t_k + \frac{h}{3}\right]$, $\left[t_k + \frac{h}{3}, t_k + \frac{2h}{3}\right]$, $\left[t_k + \frac{2h}{3}, t_{k+1}\right]$, respectively; the sum of $\Delta\boldsymbol{\theta}_1$, $\Delta\boldsymbol{\theta}_2$ and $\Delta\boldsymbol{\theta}_3$ is equal to $\Delta\boldsymbol{\theta}$; h is the attitude update period.

This algorithm is transformed by the method ξ proposed in this paper. The conversion process is as follows:

$$\begin{aligned}\xi(\boldsymbol{\phi}_{c3}) &= \xi(\Delta\boldsymbol{\theta}_1) + \xi(\Delta\boldsymbol{\theta}_2) + \xi(\Delta\boldsymbol{\theta}_3) + \frac{27}{40}\xi(\Delta\boldsymbol{\theta}_1 \times \Delta\boldsymbol{\theta}_2) \\ &\quad + \frac{9}{20}\xi(\Delta\boldsymbol{\theta}_1 \times \Delta\boldsymbol{\theta}_3) + \frac{27}{40}\xi(\Delta\boldsymbol{\theta}_2 \times \Delta\boldsymbol{\theta}_3) \\ &= \Delta\boldsymbol{v}_1 + \Delta\boldsymbol{v}_2 + \Delta\boldsymbol{v}_3 + \frac{27}{40}(\Delta\boldsymbol{v}_1 \times \Delta\boldsymbol{\theta}_2 + \Delta\boldsymbol{\theta}_1 \times \Delta\boldsymbol{v}_2) \\ &\quad + \frac{9}{20}(\Delta\boldsymbol{v}_1 \times \Delta\boldsymbol{\theta}_3 + \Delta\boldsymbol{\theta}_1 \times \Delta\boldsymbol{v}_3) \\ &\quad + \frac{27}{40}(\Delta\boldsymbol{v}_2 \times \Delta\boldsymbol{\theta}_3 + \Delta\boldsymbol{\theta}_2 \times \Delta\boldsymbol{v}_3) \end{aligned} \quad (21)$$

where, $\Delta\boldsymbol{v}_1$, $\Delta\boldsymbol{v}_2$, $\Delta\boldsymbol{v}_3$ are the velocity increments in the $\left[t_k, t_k + \frac{h}{3}\right]$, $\left[t_k + \frac{h}{3}, t_k + \frac{2h}{3}\right]$, $\left[t_k + \frac{2h}{3}, t_{k+1}\right]$, respectively; the sum of $\Delta\boldsymbol{v}_1$, $\Delta\boldsymbol{v}_2$ and $\Delta\boldsymbol{v}_3$ is equal to $\Delta\boldsymbol{v}$; h is the attitude update period.

We can find that the sculling error compensation algorithm given in [15] is exactly the same as the result of the conversion of Eq.21.

D. WANG'S ALGORITHMS

WANG first considered the third order of Picard in cone error compensation. Theoretical analysis shows that the algorithm has higher precision in high dynamic environment. We use the three-subsample third-order cone algorithm proposed as an example to verify it. The algorithm can be expressed as follows [16], [19], (22), as shown at the bottom of this page, where, $\Delta\boldsymbol{\theta}_1$, $\Delta\boldsymbol{\theta}_2$ and $\Delta\boldsymbol{\theta}_3$ are the same defined in Eq.20.

It is transformed by the transformation method ξ proposed in this paper. The conversion process is as (23), as shown at the top of the next page.

The above formula can be used to obtain the three-subsampled error compensation algorithm considering the Picard third-order term (24), as shown at the top of the next page, where, $\Delta\boldsymbol{v}_1$, $\Delta\boldsymbol{v}_2$, $\Delta\boldsymbol{v}_3$ are the same defined in Eq.21.

We can find that the sculling error compensation algorithm obtained by the conversion method ξ is exactly the same as the derived result.

V. EXPERIMENT

In order to further verify the effectiveness of the proposed transformation method, we carry out simulation experiments under pure sculling conditions and high dynamic environment. In the simulation experiment, we select three sculling error compensation algorithms obtained in the previous section through the transformation method proposed in this paper. They are Eq.19, Eq.21, and Eq.24, respectively. The corresponding algorithms are Ignagni's algorithm, Song's extension algorithm, and Wang's algorithm.

A. SIMULATION IN PURE SCULLING ENVIRONMENT

The pure sculling simulation experimental conditions used in this paper are as follows: angular amplitude 5°; linear amplitude $0.3m$; cone and scull period $2s$; sampling period $0.01s$; simulation time $6s$.

In Fig.1, Fig.1(a)-(e) depict the typical sculling motion state, Fig.1(f) is the comparison of sculling drift error. In Fig.1(f) the red line is the Ignagni's sculling error compensation algorithm; the blue line is the extended sculling error compensation algorithm proposed by Song; the black line is the high-precision sculling error compensation algorithm obtained by transforming the cone algorithm of Wang.

From Figure 1, we can see that the Ignagni's sculling error compensation algorithm has the same accuracy as the extended sculling error compensation algorithm proposed by SONG, and its Z-axis velocity error at 6s is $1.8 \times 10^{-3} m/s$. Simulation results are consistent with theoretical conclusions. This is because it only considers the second-order term of the velocity translation vector, and the cross product of the angular increment and the velocity increment depends only on the spacing of the two increments in the purely swept state regardless of its absolute time position. The high-precision

$$\begin{aligned}\boldsymbol{\phi}_{c3} &= \Delta\boldsymbol{\theta}_1 + \Delta\boldsymbol{\theta}_2 + \Delta\boldsymbol{\theta}_3 + \frac{57}{80}\Delta\boldsymbol{\theta}_1 \times \Delta\boldsymbol{\theta}_2 + \frac{33}{80}\Delta\boldsymbol{\theta}_1 \times \Delta\boldsymbol{\theta}_3 + \frac{57}{80}\Delta\boldsymbol{\theta}_2 \times \Delta\boldsymbol{\theta}_3 \\ &\quad - \frac{261}{2240}(\Delta\boldsymbol{\theta}_1 \times \Delta\boldsymbol{\theta}_2) \times \Delta\boldsymbol{\theta}_1 + \frac{207}{1120}(\Delta\boldsymbol{\theta}_1 \times \Delta\boldsymbol{\theta}_2) \times \Delta\boldsymbol{\theta}_2 + \frac{981}{2240}(\Delta\boldsymbol{\theta}_1 \times \Delta\boldsymbol{\theta}_2) \times \Delta\boldsymbol{\theta}_3 \\ &\quad - \frac{27}{448}(\Delta\boldsymbol{\theta}_1 \times \Delta\boldsymbol{\theta}_3) \times \Delta\boldsymbol{\theta}_1 - \frac{27}{160}(\Delta\boldsymbol{\theta}_1 \times \Delta\boldsymbol{\theta}_3) \times \Delta\boldsymbol{\theta}_2 + \frac{27}{448}(\Delta\boldsymbol{\theta}_1 \times \Delta\boldsymbol{\theta}_3) \times \Delta\boldsymbol{\theta}_3 \\ &\quad - \frac{45}{448}(\Delta\boldsymbol{\theta}_2 \times \Delta\boldsymbol{\theta}_3) \times \Delta\boldsymbol{\theta}_1 - \frac{207}{1120}(\Delta\boldsymbol{\theta}_2 \times \Delta\boldsymbol{\theta}_3) \times \Delta\boldsymbol{\theta}_2 + \frac{261}{2240}(\Delta\boldsymbol{\theta}_2 \times \Delta\boldsymbol{\theta}_3) \times \Delta\boldsymbol{\theta}_3 \end{aligned} \quad (22)$$

$$
\begin{aligned}
\xi(\boldsymbol{\phi}_{c3}) &= \xi(\Delta\boldsymbol{\theta}_1)+\xi(\Delta\boldsymbol{\theta}_2)+\xi(\Delta\boldsymbol{\theta}_3)+\frac{57}{80}\xi(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_2)+\frac{33}{80}\xi(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_3)+\frac{57}{80}\xi(\Delta\boldsymbol{\theta}_2\times\Delta\boldsymbol{\theta}_3)\\
&\quad-\frac{261}{2240}\xi[(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_2)\times\Delta\boldsymbol{\theta}_1]+\frac{207}{1120}\xi[(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_2)\times\Delta\boldsymbol{\theta}_2]+\frac{981}{2240}\xi[(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_2)\times\Delta\boldsymbol{\theta}_3]\\
&\quad-\frac{27}{448}\xi[(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_3)\times\Delta\boldsymbol{\theta}_1]-\frac{27}{160}\xi[(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_3)\times\Delta\boldsymbol{\theta}_2]+\frac{27}{448}\xi[(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_3)\times\Delta\boldsymbol{\theta}_3]\\
&\quad-\frac{45}{448}\xi[(\Delta\boldsymbol{\theta}_2\times\Delta\boldsymbol{\theta}_3)\times\Delta\boldsymbol{\theta}_1]-\frac{207}{1120}\xi[(\Delta\boldsymbol{\theta}_2\times\Delta\boldsymbol{\theta}_3)\times\Delta\boldsymbol{\theta}_2]+\frac{261}{2240}\xi[(\Delta\boldsymbol{\theta}_2\times\Delta\boldsymbol{\theta}_3)\times\Delta\boldsymbol{\theta}_3]\\
&= \Delta\boldsymbol{v}_1+\Delta\boldsymbol{v}_2+\Delta\boldsymbol{v}_3+\frac{57}{80}(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{v}_2+\Delta\boldsymbol{v}_1\times\Delta\boldsymbol{\theta}_2)+\frac{33}{80}(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{v}_3+\Delta\boldsymbol{v}_1\times\Delta\boldsymbol{\theta}_3)\\
&\quad+\frac{57}{80}(\Delta\boldsymbol{\theta}_2\times\Delta\boldsymbol{v}_3+\Delta\boldsymbol{v}_2\times\Delta\boldsymbol{\theta}_3)-\frac{261}{2240}[\xi(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_2)\times\Delta\boldsymbol{\theta}_1+(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_2)\times\xi(\Delta\boldsymbol{\theta}_1)]\\
&\quad+\frac{207}{1120}[\xi(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_2)\times\Delta\boldsymbol{\theta}_2+(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_2)\times\xi(\Delta\boldsymbol{\theta}_2)]+\frac{981}{2240}[\xi(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_2)\times\Delta\boldsymbol{\theta}_3+(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_2)\times\xi(\Delta\boldsymbol{\theta}_3)]\\
&\quad-\frac{27}{448}[\xi(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_3)\times\Delta\boldsymbol{\theta}_1+(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_3)\times\xi(\Delta\boldsymbol{\theta}_1)]-\frac{27}{160}[\xi(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_3)\times\Delta\boldsymbol{\theta}_2+(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_3)\times\xi(\Delta\boldsymbol{\theta}_2)]\\
&\quad+\frac{27}{448}[\xi(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_3)\times\Delta\boldsymbol{\theta}_3+(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_3)\times\xi(\Delta\boldsymbol{\theta}_3)]-\frac{45}{448}[\xi(\Delta\boldsymbol{\theta}_2\times\Delta\boldsymbol{\theta}_3)\times\Delta\boldsymbol{\theta}_1+(\Delta\boldsymbol{\theta}_2\times\Delta\boldsymbol{\theta}_3)\times\xi(\Delta\boldsymbol{\theta}_1)]\\
&\quad-\frac{207}{1120}[\xi(\Delta\boldsymbol{\theta}_2\times\Delta\boldsymbol{\theta}_3)\times\Delta\boldsymbol{\theta}_2+(\Delta\boldsymbol{\theta}_2\times\Delta\boldsymbol{\theta}_3)\times\xi(\Delta\theta_2)]+\frac{261}{2240}[\xi(\Delta\boldsymbol{\theta}_2\times\Delta\boldsymbol{\theta}_3)\times\Delta\boldsymbol{\theta}_3+(\Delta\boldsymbol{\theta}_2\times\Delta\boldsymbol{\theta}_3)\times\xi(\Delta\boldsymbol{\theta}_3)]
\end{aligned}
\tag{23}
$$

$$
\begin{aligned}
\xi(\boldsymbol{\phi}_{c3}) &= \Delta\boldsymbol{v}_1+\Delta\boldsymbol{v}_2+\Delta\boldsymbol{v}_3+\frac{57}{80}(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{v}_2+\Delta\boldsymbol{v}_1\times\Delta\boldsymbol{\theta}_2)+\frac{33}{80}(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{v}_3+\Delta\boldsymbol{v}_1\times\Delta\boldsymbol{\theta}_3)\\
&\quad+\frac{57}{80}(\Delta\boldsymbol{\theta}_2\times\Delta\boldsymbol{v}_3+\Delta\boldsymbol{v}_2\times\Delta\boldsymbol{\theta}_3)-\frac{261}{2240}(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_2)\times\Delta\boldsymbol{v}_1+\frac{207}{1120}(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_2)\times\Delta\boldsymbol{v}_2\\
&\quad+\frac{981}{2240}(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_2)\times\Delta\boldsymbol{v}_3\\
&\quad-\frac{27}{448}(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_3)\times\Delta\boldsymbol{v}_1-\frac{27}{160}(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_3)\times\Delta\boldsymbol{v}_2+\frac{27}{448}(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{\theta}_3)\times\Delta\boldsymbol{v}_3-\frac{45}{448}(\Delta\boldsymbol{\theta}_2\times\Delta\boldsymbol{\theta}_3)\times\Delta\boldsymbol{v}_1\\
&\quad-\frac{207}{1120}(\Delta\boldsymbol{\theta}_2\times\Delta\boldsymbol{\theta}_3)\times\Delta\boldsymbol{v}_2+\frac{261}{2240}(\Delta\boldsymbol{\theta}_2\times\Delta\boldsymbol{\theta}_3)\times\Delta\boldsymbol{v}_3+\frac{261}{2240}\Delta\boldsymbol{\theta}_1\times(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{v}_2-\Delta\boldsymbol{\theta}_2\times\Delta\boldsymbol{v}_1)\\
&\quad+\frac{27}{448}\Delta\boldsymbol{\theta}_1\times(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{v}_3-\Delta\boldsymbol{\theta}_3\times\Delta\boldsymbol{v}_1)+\frac{45}{448}\Delta\boldsymbol{\theta}_1\times(\Delta\boldsymbol{\theta}_2\times\Delta\boldsymbol{v}_3-\Delta\boldsymbol{\theta}_3\times\Delta\boldsymbol{v}_2)\\
&\quad-\frac{207}{1120}\Delta\boldsymbol{\theta}_2\times(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{v}_2-\Delta\boldsymbol{\theta}_2\times\Delta\boldsymbol{v}_1)+\frac{27}{160}\Delta\boldsymbol{\theta}_2\times(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{v}_3-\Delta\boldsymbol{\theta}_3\times\Delta\boldsymbol{v}_1)\\
&\quad+\frac{207}{1120}\Delta\boldsymbol{\theta}_2\times(\Delta\boldsymbol{\theta}_2\times\Delta\boldsymbol{v}_3-\Delta\boldsymbol{\theta}_3\times\Delta\boldsymbol{v}_2)-\frac{981}{2240}\Delta\boldsymbol{\theta}_3\times(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{v}_2-\Delta\boldsymbol{\theta}_2\times\Delta\boldsymbol{v}_1)\\
&\quad-\frac{27}{448}\Delta\boldsymbol{\theta}_3\times(\Delta\boldsymbol{\theta}_1\times\Delta\boldsymbol{v}_3-\Delta\boldsymbol{\theta}_3\times\Delta\boldsymbol{v}_1)-\frac{261}{2240}\Delta\boldsymbol{\theta}_3\times(\Delta\boldsymbol{\theta}_2\times\Delta\boldsymbol{v}_3-\Delta\boldsymbol{\theta}_3\times\Delta\boldsymbol{v}_2)
\end{aligned}
\tag{24}
$$

sculling error compensation algorithm obtained by transforming the cone algorithm of Wang has a Z-axis velocity of 6 s error of $1.1\times10^{-3}m/s$. It has higher precision in purely sculling state.

The simulation results show that the sculling error compensation algorithm obtained by the transformation method proposed in this paper is completely consistent with the theoretical precision and the accuracy of the proposed algorithm in the literature, which proves the effectiveness of the proposed transformation method.

B. SIMULATION IN HIGH DYNAMIC ENVIRONMENT

In order to more fully verify the effectiveness of the proposed transformation method, we performed a simulation experiment under high dynamic conditions. The angular rate and acceleration under high dynamic conditions used in the simulation are [21]

$$
\begin{aligned}
\omega_x &= 1.1+0.9(t-t_{m-1})-0.6\left[(t-t_{m-1})^2/2!\right]\\
&\quad+1.1\left[(t-t_{m-1})^3/3!\right]-0.1\left[(t-t_{m-1})^4/4!\right]rad/s\\
\omega_y &= -0.5+1.0(t-t_{m-1})+0.3\left[(t-t_{m-1})^2/2!\right]\\
&\quad+0.7\left[(t-t_{m-1})^3/3!\right]+0.2\left[(t-t_{m-1})^4/4!\right]rad/s\\
\omega_z &= 0.3-1.2(t-t_{m-1})+2\left[(t-t_{m-1})^2/2!\right]\\
&\quad-0.9\left[(t-t_{m-1})^3/3!\right]+0.4\left[(t-t_{m-1})^4/4!\right]rad/s\\
a_{SF_x} &= 3.5-2.3(t-t_{m-1})+1.5\left[(t-t_{m-1})^2/2!\right]
\end{aligned}
$$

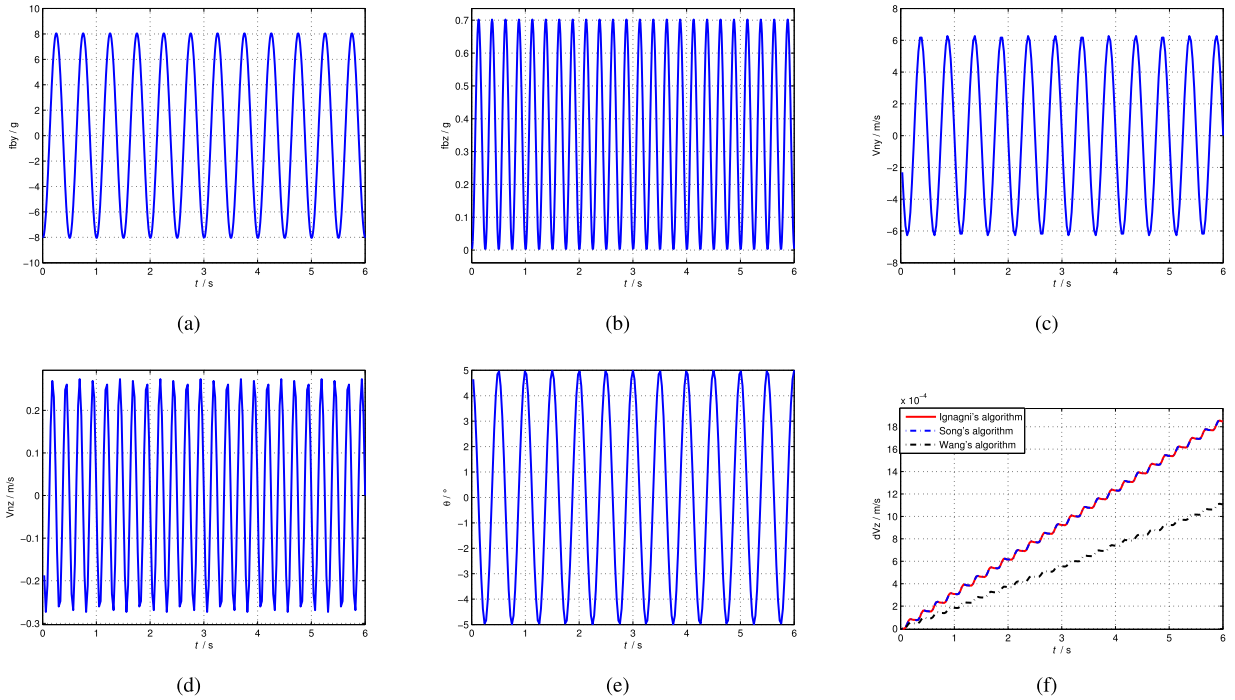


FIGURE 1. Simulation in pure sculling environment (a) Y axial acceleration (b) Z axial acceleration (c) Y axial velocity (d) Z axial velocity(e) X axial angular motion(f) Comparison of sculling drift error.

TABLE 1. Velocity error simulation in high dynamic environment.

Converted algorithm	Simulation results(m/s)	Exact results(m/s)	Velocity error(m/s)
Ignagni's algorithm	0.263149562827659	0.263148946228663	6.16599E-07
	0.585790206754473	0.585793264218951	-3.05746E-06
	-1.379255751623962	-1.379256062459716	3.10832E-07
Song's algorithm	0.263149372571892		4.26347E-07
	0.585790428066367		-2.83615E-06
	-1.379255800587521		2.61872E-07
Wang's algorithm	0.263149241366383		2.95138E-07
	0.585790598282977		-2.66593E-06
	-1.379255822508652		2.39951E-07

$$
\begin{aligned}
&+6.1\left[(t-t_{m-1})^3/3!\right]-2.7\left[(t-t_{m-1})^4/4!\right]m/s^2\\
a_{SF_y} &= 7.3+1.5\,(t-t_{m-1})-2.7\left[(t-t_{m-1})^2/2!\right]\\
&-3.6\left[(t-t_{m-1})^3/3!\right]+1.9\left[(t-t_{m-1})^4/4!\right]m/s^2\\
a_{SF_z} &= -9-5.6\,(t-t_{m-1})+4.6\left[(t-t_{m-1})^2/2!\right]\\
&+4.3\left[(t-t_{m-1})^3/3!\right]-3.5\left[(t-t_{m-1})^4/4!\right]m/s^2
\end{aligned}
\tag{25}
$$

In order to verify the validity of the proposed transformation algorithm,the navigation result of the 0.1s data of sampling frequency 10MHZ is taken as the exact velocity solution. The three sculling error compensation algorithms obtained by the conversion method proposed in this paper are used for verification. The Ignagni's algorithm, Song's algorithm and Wang's algorithm have the velocity error of 0.1s as shown in Table 1.

The simulation results show that the accuracy of the extended algorithm of Song is better than that of the traditional algorithm; the high-precision sculling error compensation algorithm obtained by transforming the cone algorithm of Wang is better than the Song extended algorithm. The results show that the sculling algorithm obtained by the conversion method is exactly the same as the algorithm derived from the corresponding literature, which proves the feasibility of the proposed transformation method.

VI. CONCLUSION

In view of the research that the scholars mainly spend their energy on the design of the cone error compensation algorithm, and the research on the sculling error compensation algorithm is less. This paper proposes a new conversion method, which can convert the cone algorithm into the corresponding sculling algorithm accurately by simple operation. We theoretically prove the general equivalence between the sculling algorithm and the cone algorithm, and prove the effectiveness of the conversion method. In addition, we also take the existing cone error compensation algorithm as examples, and give its conversion process and conversion results. The conversion results are exactly the same as the derivation

error compensation algorithms given by the derivation, proving the effectiveness of the proposed method. Since most of the previous research work focused on the design of attitude solving, a large number of cone algorithms have been designed. The conversion method can directly obtain the corresponding sculling error compensation algorithm, which undoubtedly simplifies the design of the sculling algorithm.

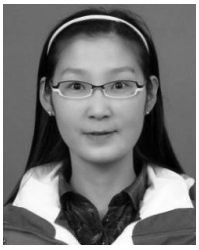
ACKNOWLEDGMENT

The authors want to express their sincere thanks to the reviewers and the associate editor of this paper for their constructive comments and suggestions, which improved the quality of this paper.

REFERENCES

[1] Y. Wu, "RodFIter: Attitude reconstruction from inertial measurement by functional iteration," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 5, pp. 2131–2142, Feb. 2018.
[2] Y. Wu, Q. Cai, and T.-K. Truong, "Fast RodFIter for attitude reconstruction from inertial measurements," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 1, pp. 419–428, Aug. 2019.
[3] G. M. Yan, J. Weng, X. Yang, and Y. Qin, "An accurate numerical solution for strapdown attitude algorithm based on Picard iteration," *J. Astronaut.*, vol. 38, no. 12, pp. 1308–1313, 2017.
[4] G. M. Yan, W. S. Yan, and D. M. Xu, "Limitations of error estimation for classic coning compensation algorithm," *J. Chin. Inertial Technol.*, vol. 16, no. 4, pp. 379–385, Dec. 2008.
[5] P. G. Savage, "Coning algorithm design by explicit frequency shaping," *J. Guid. Control Dyn.*, vol. 33, no. 4, pp. 1123–1132, 2010.
[6] R. B. Miller, "A new strapdown attitude algorithm," *J. Guid., Control, Dyn.*, vol. 6, no. 4, pp. 287–291, 1983.
[7] M. B. Ignagni, "Efficient class of optimized coning compensation algorithms," *J. Guid., Control, Dyn.*, vol. 19, no. 2, pp. 424–429, 1996.
[8] Y. F. Jiang and Y. P. Lin, "On the rotation vector differential equation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 27, no. 1, pp. 181–183, Jan. 1991.
[9] Y. FuhJiang and Y. P. Lin, "Improved strapdown coning algorithms," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 28, no. 2, pp. 484–490, Apr. 1992.
[10] H. Musoff and J. H. Murphy, "Study of strapdown navigation attitude algorithms," *J. Guid. Control Dyn.*, vol. 18, no. 2, pp. 287–290, 1995.
[11] A. Panov, *Mathematical Fundamentals of Inertial Orientation Theory*. Kyiv, Ukraine, Naukova Dumka, 1995.
[12] V. Z. Gusinsky, V. M. Lesyuchevsky, Y. A. Litmanovich, H. Musoff, and G. T. Schmidt, "New procedure for deriving optimized strapdown attitude algorithms," *J. Guid., Control, Dyn.*, vol. 20, no. 4, pp. 673–680, 1997.
[13] V. Z. Gusinsky, V. M. Lesyuchevsky, Y. A. Litmanovich, H. Musoff, and G. Schmidt, "Optimization of a strapdown attitude algorithm for a stochastic motion," *Navigation*, vol. 44, no. 2, pp. 163–170, 1997.
[14] P. Savage, "Strapdown inertial navigation integration algorithm design Part 1: Attitude algorithms," *J. Guid., Control Dyn.*, vol. 21, no. 1, pp. 19–28, 1998.
[15] M. Song, "Research on error analysis and optimization methods for strapdown inertial navigation algorithm under highly dynamic environment," Ph.D. dissertation, Dept. Automat. Control, National Univ. Defense Technol., Changsha, China, 2012.
[16] M. Wang, W. Wu, and X. He, "Design and evaluation of high-order non-commutativity error compensation algorithm in dynamics," in *Proc. IEEE/ION Position, Location Navigat. Symp. (PLANS)*, Apr. 2018, pp. 34–41.
[17] M. B. Ignagni, "Duality of optimal strapdown sculling and coning compensation algorithms," *Navigation*, vol. 45, no. 2, pp. 85–95, 1998.
[18] P. G. Savage, "Strapdown inertial navigation integration algorithm design Part 2: Velocity and position algorithms," *J. Guid., Control, Dyn.*, vol. 21, no. 2, pp. 208–221, 1998.
[19] M. Wang, W. Wu, X. He, G. Yang, and H. Yu, "Higher-order rotation vector attitude updating algorithm," *J. Navigat.*, vol. 72, no. 3, pp. 721–740, 2019.
[20] K. M. Roscoe, "Access equivalency between strapdown inertial navigation coning and sculling integrals/algorithms," *J. Guid., Control, Dyn.*, vol. 24, no. 2, pp. 201–205, 2001.
[21] P. G. Savage, "A unified mathematical framework for strapdown algorithm design," *J. Guid., Control, Dyn.*, vol. 29, no. 2, pp. 237–249, 2006.

PAN JIANG received the bachelor's degree from Harbin Engineering University, China, in 2015. He is currently pursuing the Ph.D. degree with the Harbin Institute of Technology. His main research interest includes high precision inertial navigation algorithm.



YA ZHANG received the B.Eng. and Ph.D. degrees from Harbin Engineering University, Harbin, Heilongjiang, China, in 2010 and 2015, respectively. She holds a postdoctoral position with the School of Electrical Engineering and Automation, Harbin Institute of Technology, Harbin. Her current research interests include vision-based mobile robot navigation systems and multi-sensor information fusion.

QIANG HAO was born in 1992. He received the B.S. degree from the Department of Automation, Harbin Engineering University, Harbin, China, in 2015. He is currently pursuing the Ph.D. degree in instruments science and technology with the Harbin Institute of Technology. His current research interests include inertial navigation systems and indoor positioning systems.

SHIWEI FAN was born in 1993. He received the B.S. degree from the Department of Automation, Harbin Engineering University, Harbin, China, in 2015. He is currently pursuing the Ph.D. degree in instruments science and technology with the Harbin Institute of Technology. His current research interests include inertial navigation systems and cooperative navigation systems.

DINGJIE XU born in April 1966. He is currently a Professor and a Doctoral Tutor with the Institute of Navigation Instrument, Harbin Institute of Technology. He is mainly engaged in teaching and research in navigation, guidance, and control.