Strapdown Inertial Navigation System Algorithms Based on Dual Quaternions

YUANXIN WU

XIAOPING HU

DEWEN HU, Member, IEEE

TAO LI

JUNXIANG LIAN
National University of Defense Technology
P. R. China

The design of strapdown inertial navigation system (INS) algorithms based on dual quaternions is addressed. Dual quaternion is a most concise and efficient mathematical tool to represent rotation and translation simultaneously, i.e., the general displacement of a rigid body. The principle of strapdown inertial navigation is represented using the tool of dual quaternion. It is shown that the principle can be expressed by three continuous kinematic equations in dual quaternion. These equations take the same form as the attitude quaternion rate equation. Subsequently, one new numerical integration algorithm is structured to solve the three kinematic equations, utilizing the traditional two-speed approach originally developed in attitude integration. The duality between the coning and sculling corrections, raised in the recent literature, can be essentially explained by splitting the new algorithm into the corresponding rotational and translational parts. The superiority of the new algorithm over conventional ones in accuracy is analytically derived. A variety of simulations are carried out to support the analytic results. The numerical results agree well with the analyses. The new algorithm turns out to be a better choice than any conventional algorithm for high-precision navigation systems and high-maneuver applications. Several guidelines in choosing a suitable navigation algorithm are also provided.

Manuscript received March 3, 2003; revised February 3 and August 12, 2004; released for publication October 12, 2004.

IEEE Log No. T-AES/41/1/844814.

Refereeing of this contribution was handled by R. C. Michelson.

This work was supported in part by National Natural Science Foundation of China (60374006, 60234030 and 60171003), Distinguished Young Scholars Fund of China (60225015), and Ministry of Education of China (TRAPOYT Project).

Authors' address: Laboratory of Inertial Technology, Dept. of Automatic Control, College of Mechatronics and Automation, National University of Defense Technology, Changsha, Hunan, P. R. China, 410073, E-mail: (yuanx_wu@hotmail.com).

0018-9251/05/$17.00 © 2005 IEEE

I. NOMENCLATURE

$\mathbf{g}$	Gravitational acceleration
$\mathbf{g}_l$	Local gravity vector
$\mathbf{s}$	Specific force, defined as the acceleration relative to nonrotating inertial space produced by applied nongravitational forces, measured by the accelerometer triad
C	Arbitrary coordinate frame
$\mathbf{r}^C$	Vector expressed in frame C
$\boldsymbol{\omega}_{C_1C_2}$	Angular rate vector of frame C_2 relative to frame C_1
$\mathbf{v}_t$	Thrust velocity, integration of specific force
$\mathbf{v}_g$	Gravitational velocity, sum of the vehicle's initial velocity and integration of the gravitational acceleration
E	Earth frame, with its origin at center of Earth, one axis parallel to Earth polar axis, the other axes fixed to Earth and parallel to equatorial plane
I	Inertial frame, Earth frame at time instant $t = 0$
B	Body frame, strapdown inertial sensor coordinate frame with its origin at center of mass of accelerometer triad and axes parallel to nominal right-handed orthogonal sensor input axes
T	Thrust velocity frame, with its axes aligned with those of body frame. Vector from Earth's center to its origin is exactly thrust velocity.
G	Gravitational velocity frame, with its axes aligned with those of the Earth frame. Vector from Earth's center to its origin is exactly gravitational velocity.
U	Position frame, with its axes aligned with those of Earth frame. Vector from Earth's center to its origin is exactly position vector of vehicle.
q	Quaternion
$\hat{q}$	Dual quaternion
$\hat{\omega}$	Twist
$\hat{\sigma}$	Screw vector
$*$	Conjugate operator
$\circ$	Quaternion multiplication
$\cdot$	Scalar product
$\times$	Vector product.

II. INTRODUCTION

The inertial navigation system (INS) is a system of calculating velocity by integration of the total acceleration and computing position by integration of the resultant velocity. The strapdown INS does away with most of the mechanical complexity of the platform INS by having the sensors attached rigidly to the body of the vehicle, for the benefits of lower cost, reduced size, greater reliability, etc. The computing task of a strapdown INS computer is usually composed of attitude computation by integration of the angular rate and acceleration vector transformation that is twice integrated into velocity and position. For high-performance systems, the latter part of the task is further divided into velocity integration and position integration, in which position integration also requires being delicately treated [37].

Since the emergence of the strapdown INS concept in the late 1950s, nearly half a century has seen the great strides in the development of strapdown INS theory and applications. Among the numerous treatises on strapdown INS algorithms, most have been focused on the design of two-speed algorithms for attitude computation, in which an analytically exact equation is used at moderate speed to update the integration parameter, direction cosine matrix (DCM) or quaternion, with input provided from a high-speed algorithm (the coning algorithm) [7, 18–21, 23, 26, 29, 31–33, 36, 43]. The coning algorithm utilizes an intermediate parameter such as the rotation vector or some kind of generalized kinematic vectors [10, 43], etc., to measure rectified dynamic motion within the parameter update interval. In contrast, there are merely a few literatures dwelling upon the development of the companion algorithms for the specific force transformation, velocity/position integration [22, 27, 35, 37]. In the two fundamental and summary papers [36, 37], Savage discussed the position updating for high-resolution applications for the first time and called it the scrolling algorithm, which is of similar mechanism with the coning algorithm and the sculling algorithm (for velocity updating). As Litmanovich stated in [27], "these papers properly marked the end of a long period of algorithm development" for the strapdown INS.

So far in most, if not all, fields, researches have been made within the framework of vector algebra. When it comes to the general displacement of a rigid body, however, there is no such a mathematical tool in vector algebra as to treat rotation and translation simultaneously. Fortunately, from a viewpoint of kinematics the general displacement can be taken apart into two separate motions, i.e., fixed-point rotation and translation, in which rotation is completely independent of translation [30]. This means that we can otherwise treat rotation and translation in a different manner. In the navigation community, DCM/quaternion and vector are chosen to represent rotation and translation, respectively. So are the strapdown INS algorithms. Individual algorithm as mentioned above has to be structured for attitude integration and velocity/position integration, respectively. The characteristics of duality/equivalence between the coning and sculling algorithms were revealed recently in [22, 35], but the algorithm design and implementation are still rather involved.

Is there any mathematical language that represents rotation and translation in a consolidating and compact manner? Is it possible to reduce the perplexing and error-prone strapdown algorithms to some extent? The answers to these questions are both affirmative.

According to Chasles theorem [11], one of the most fundamental results in spatial kinematics, the general displacement of a rigid body in space consists of a rotation about an axis (called the screw axis) and a translation parallel to that axis. Since this theorem can be visually demonstrated through the motion performed by a physical screw in the hardware store, the general displacement has a visual name, the screw motion. As shown in Chasles' theorem, the screw motion is characterized by two essential elements: the screw axis and a pitch, which relates rotation about the screw axis with translation along the screw axis. In 1900, Ball [3] founded screw theory to express velocities in three-dimensional space, combining both rotational and translational parts together. It provides us the possibility of representing the screw motion in a compact mathematical language. Among so many mathematical approaches as homogeneous transformation [14], quaternion/vector pairs [17], Lie algebra [6] and alike, dual quaternion has been found so far to be the most compact and efficient way to express the screw motion [1, 16, 17]. Up to now, dual quaternion has turned out to be an elegant and useful tool for kinematic (and dynamic) analysis in many research areas such as in mechanics [44] and robotics [5, 13].

In the research done by Branets and Shmyglevsky in 1992 [10], the principle of strapdown inertial navigation was first restated in terms of dual quaternion in the literature as far as the authors know. They devised a Picard-type recursive integration based algorithm to solve the differential kinematic equations, which, however, has already been abandoned in the modern algorithms [36]. Unfortunately, for the sake of language few researchers in the navigation community have been aware of the benefits of dual quaternion during the last decade. Thus we regard it as one of the purposes of this work to propose that dual quaternion is an elegant and efficient mathematical tool for investigating strapdown INS algorithms.

Motivated by [10], this paper follows Branets and Shmyglevsky's footprints investigating the application of dual quaternion in designing the strapdown INS algorithm. The contents are organized as follows. Section III presents some background materials for the convenience of further discussions. Section IV reinterprets the principle of strapdown inertial navigation using the tool of dual quaternion. The resultant kinematic equations in dual quaternion serve as the foundation of subsequent algorithm design. Section V develops one new numerical integration algorithm to solve all of these kinematic equations, utilizing the traditional two-speed approach originally developed in attitude integration. Section VI makes analytic comparisons between the new algorithm and the conventional ones in various aspects. To justify the new algorithm based on dual quaternions, a variety of simulations are carried out in Section VII. Finally, the concluding remarks are made in Section VIII.

III. DUAL QUATERNION

In this section, a brief summary of quaternion, dual number, and dual quaternion is presented for ease of reference and providing the foundation for mathematical formulations to be developed. More detailed treatment on these subjects can be found in the previous research works [9, 10, 15, 24, 44].

A. Quaternions

Quaternion was invented by Hamilton as an extension of the complex number to $\Re^4$. A definition of quaternion is given as $q = [s, \mathbf{v}]$, where s is a scalar (called the scalar part), and $\mathbf{v}$ is a three-dimensional vector (called the vector part). By definition, quaternions satisfy the following operations

$$\begin{aligned} q_1 + q_2 &= [s_1 + s_2, \mathbf{v}_1 + \mathbf{v}_2] \\ \lambda q &= [\lambda s, \lambda \mathbf{v}] \end{aligned} \tag{1}$$

where λ is a scalar. The multiplication between quaternions is defined as

$$q_1 \circ q_2 = [s_1 s_2 - \mathbf{v}_1 \cdot \mathbf{v}_2, s_1 \mathbf{v}_2 + s_2 \mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2]. \tag{2}$$

The norm of a quaternion is defined as $\|q\| = q \circ q^*$ where $q^* = [s, -\mathbf{v}]$ is the conjugate quaternion.

Quaternion can be used to facilitate the representation of rotation[1] instead of other parameters such as DCM, etc. For the frame rotation about a unit axis $\mathbf{n}$ with an angle θ, there is a unit quaternion[2] $q = [\cos(\theta/2), \sin(\theta/2)\mathbf{n}]$ relating a fixed vector expressed in the original frame $\mathbf{r}^O$ with the same vector expressed in the new frame $\mathbf{r}^N$ by

$$r^N = q^* \circ r^O \circ q. \tag{3}$$

r^O and r^N are two quaternions with vanishing scalar parts formed by $\mathbf{r}^O$ and $\mathbf{r}^N$, respectively. For example, $r^O = [0, \mathbf{r}^O]$. If not specifically stated, a vector is treated equivalently as its corresponding quaternion with a vanishing scalar part.

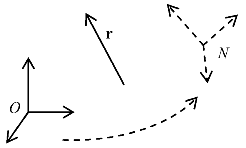


Fig. 1. Frame rotation depicts fixed vector $\mathbf{r}$ in rotating frame O. Frame N can be acquired from frame O by rotation succeeded by translation or translation succeeded by rotation.

The kinematic equations of quaternion are

$$2\dot{q} = \boldsymbol{\omega}_{ON}^O \circ q = q \circ \boldsymbol{\omega}_{ON}^N \tag{4}$$

where $\boldsymbol{\omega}_{ON}^N$ is the angular rate vector of the frame N relative to the frame O, expressed in the frame N.

B. Dual Numbers

Dual numbers were invented by Clifford [12] and further developed by Study [39]. A dual number is defined as

$$\hat{z} = a + \varepsilon b \qquad \text{with} \quad \varepsilon^2 = 0 \quad \text{but} \quad \varepsilon \neq 0 \tag{5}$$

where a is called the real part and b the dual part. In matrix algebra for example,

$$\varepsilon = \begin{bmatrix} 0 & 0 \\ x & 0 \end{bmatrix}$$

with a real x satisfies the above condition. By definition, dual numbers satisfy

$$\begin{aligned} \hat{z}_1 + \hat{z}_2 &= a_1 + a_2 + \varepsilon(b_1 + b_2) \\ \lambda \hat{z} &= \lambda a + \varepsilon \lambda b \\ \hat{z}_1 \hat{z}_2 &= a_1 a_2 + \varepsilon(a_1 b_2 + a_2 b_1) \end{aligned} \tag{6}$$

where λ is a scalar. A dual number with zero real part cannot be inverted, because the inverse is not well defined, e.g., $1/(\varepsilon b) = \varepsilon b/(\varepsilon b)^2 = \varepsilon b/0$. The function in dual number arguments has an important property. Since all powers of ε equal to or greater than two are zero, a Taylor expansion yields

$$f(a + \varepsilon b) = f(a) + \varepsilon b f'(a). \tag{7}$$

Specifically we have

$$\begin{aligned} \sin(a + \varepsilon b) &= \sin a + \varepsilon b \cos a \\ \cos(a + \varepsilon b) &= \cos a - \varepsilon b \sin a. \end{aligned} \tag{8}$$

A dual vector is a special class of dual number whose real and dual parts are both vectors, or in other words, a dual vector is analogous to a usual vector with the exception of its components being dual numbers.

A unit dual vector is one of alternative representations of lines in space. The real part is the unit direction vector of a line and the dual part is the line moment with respect to the origin of the

[1] In this paper, we discriminate the vector rotation from the frame rotation, namely, a rotating vector in a fixed frame versus a fixed vector in a rotating frame. These two kinds of rotation are actually equivalent in that a vector rotation is a frame rotation in the opposite direction, and vice versa. It depends on where the observer stands, on the frame or on the vector. The frame rotation is illustrated in Fig. 1 and will be frequently used in the sequel.

[2] The Euler's theorem tells us that any displacement of a rigid body such that a point, say $\mathbf{p}$, on the rigid body remains fixed, is equivalent to a rotation by a fixed axis through the point $\mathbf{p}$.

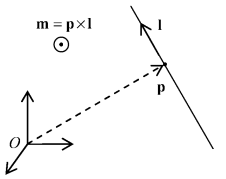


Fig. 2. Plücker line. (Line moment directs outwards.)

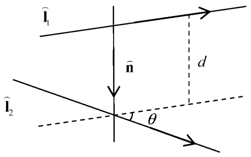


Fig. 3. Relationship between lines.

coordinate frame.[3] In this case, a unit dual vector is known as the Plücker coordinate or the Plücker line. Accordingly a line with unit direction vector $\mathbf{l}$ (boldfaced l) and passing through a point $\mathbf{p}$, as shown in Fig. 2, can be represented as a Plücker line $\hat{\mathbf{l}} = \mathbf{l} + \varepsilon\mathbf{m}$, where the line moment $\mathbf{m} = \mathbf{p} \times \mathbf{l}$. The line moment is normal to the plane passing through the line and the frame origin.

Plücker lines can be manipulated to derive the relationship between lines. For example, the scalar product between two Plücker lines is equal to the cosine of a dual angle $\hat{\theta} = \theta + \varepsilon d$ that has a nice geometric interpretation: θ is their crossing angle and d their common perpendicular distance, as shown in Fig. 3. It can be formulated as

$$\hat{\mathbf{l}}_1 \cdot \hat{\mathbf{l}}_2 = \cos\hat{\theta} \tag{9}$$

where $\hat{\mathbf{l}}_1 = \mathbf{l}_1 + \varepsilon\mathbf{m}_1$ and $\hat{\mathbf{l}}_2 = \mathbf{l}_2 + \varepsilon\mathbf{m}_2$, respectively. As for the vector product, we have in analogy to usual vectors

$$\hat{\mathbf{l}}_1 \times \hat{\mathbf{l}}_2 = \sin\hat{\theta}\hat{\mathbf{n}} \tag{10}$$

where the new Plücker line $\hat{\mathbf{n}}$ represents the common perpendicularly intersecting line in the direction of $\mathbf{l}_1 \times \mathbf{l}_2$, as shown in Fig. 3.

C. Dual Quaternions

A dual quaternion is in fact a usual quaternion with dual number components, i.e., $\hat{q} = [\hat{s}, \hat{\mathbf{v}}]$ where $\hat{s}$ is a dual number and $\hat{\mathbf{v}}$ a dual vector. It can also be reformulated as a dual number with quaternion components

$$\hat{q} = q + \varepsilon q' \tag{11}$$

where q and q' are both usual quaternions. Obviously, a dual vector can be regarded as a dual quaternion with zero scalar part. The operations of dual quaternions have the same characteristics as the usual quaternions

$$\begin{aligned} \hat{q}_1 + \hat{q}_2 &= [\hat{s}_1 + \hat{s}_2, \hat{\mathbf{v}}_1 + \hat{\mathbf{v}}_2] \\ \lambda\hat{q} &= [\lambda\hat{s}, \lambda\hat{\mathbf{v}}] \\ \hat{q}_1 \circ \hat{q}_2 &= [\hat{s}_1\hat{s}_2 - \hat{\mathbf{v}}_1 \cdot \hat{\mathbf{v}}_2, \hat{s}_1\hat{\mathbf{v}}_2 + \hat{s}_2\hat{\mathbf{v}}_1 + \hat{\mathbf{v}}_1 \times \hat{\mathbf{v}}_2]. \end{aligned} \tag{12}$$

The norm of a dual quaternion is defined as $\|\hat{q}\| = \hat{q}\hat{q}^*$ and is a dual number. If the norm of a dual quaternion has a nonvanishing real part, its inverse can be obtained using $\hat{q}^{-1} = \|\hat{q}\|^{-1}\hat{q}^*$, where $\hat{q}^* = [\hat{s}, -\hat{\mathbf{v}}]$ is the conjugate dual quaternion. Therefore the inverse of a dual quaternion with identity norm exists and is exactly its conjugate.

Suppose that the relationship between the frame O and the frame N in Fig. 1 is described by a rotation q succeeded by a translation $\mathbf{t}^N$ (or a translation $\mathbf{t}^O$ succeeded by a rotation q). According to (3) $\mathbf{t}^N = q^* \circ \mathbf{t}^O \circ q$. Then it can be shown that a Plücker line $\hat{\mathbf{l}}$ satisfies $\hat{\mathbf{l}}^N = \hat{q}^* \circ \hat{\mathbf{l}}^O \circ \hat{q}$, where the unit dual quaternion $\hat{q}$ is a function of q and $\mathbf{t}^O$ (or q and $\mathbf{t}^N$), i.e. (see Appendix A for details)

$$\hat{q} = q + \varepsilon q' = q + \varepsilon\tfrac{1}{2}\mathbf{t}^O \circ q = q + \varepsilon\tfrac{1}{2}q \circ \mathbf{t}^N. \tag{13}$$

We have from

$$\begin{aligned} \mathbf{t}^O &= 2q' \circ q^* \\ \mathbf{t}^N &= 2q^* \circ q'. \end{aligned} \tag{14}$$

The unit dual quaternion can also be explicitly written as a function of elements of the screw motion

$$\hat{q} = \left[\cos\left(\frac{\hat{\theta}}{2}\right), \sin\left(\frac{\hat{\theta}}{2}\right)\hat{\mathbf{n}}\right] \tag{15}$$

where $\hat{\mathbf{n}}$ is the screw axis and $\hat{\theta}$ is the dual angle of the screw motion (see Appendix B for details).

The kinematic equations of dual quaternion are (see Appendix C for details)

$$2\dot{\hat{q}} = \hat{\boldsymbol{\omega}}_{ON}^O \circ \hat{q} = \hat{q} \circ \hat{\boldsymbol{\omega}}_{ON}^N \tag{16}$$

where the dual vectors

$$\hat{\boldsymbol{\omega}}_{ON}^O = \boldsymbol{\omega}_{ON}^O + \varepsilon(\dot{\mathbf{t}}^O + \mathbf{t}^O \times \boldsymbol{\omega}_{ON}^O) \tag{17}$$

and

$$\hat{\boldsymbol{\omega}}_{ON}^N = \boldsymbol{\omega}_{ON}^N + \varepsilon(\dot{\mathbf{t}}^N + \boldsymbol{\omega}_{ON}^N \times \mathbf{t}^N) \tag{18}$$

are called twists [3].

It appears that dual quaternion has many properties in common with quaternion. In fact, according to the principle of transference by Kotelnikov [25], the characteristics of quaternion are completely inherited by dual quaternion.

IV. STRAPDOWN INERTIAL NAVIGATION EQUATIONS IN CONTINUOUS FORM

The INS is a system of calculating velocity by integration of the total acceleration and computing

[3] The moment of a line is defined as the cross product of a point vector on it and its unit direction vector.

position by integration of the resultant velocity. The total acceleration is calculated as the sum of the gravitational acceleration and the acceleration measured by accelerometers, known as the specific force. In the inertial frame I, the total acceleration is expressed as

$$\ddot{\mathbf{r}}^I = \mathbf{g}^I + \mathbf{s}^I \tag{19}$$

where $\mathbf{r}$ is the position vector from the origin to the center of mass of the INS accelerometer triad.

Integrating twice with respect to time gives

$$\begin{aligned}\dot{\mathbf{r}}^I &= \dot{\mathbf{r}}^I(0) + \int_0^t (\mathbf{g}^I + \mathbf{s}^I) d\tau \\ \mathbf{r}^I &= \mathbf{r}^I(0) + \dot{\mathbf{r}}^I(0)t + \int_0^t \left[\int_0^{\tau_1} (\mathbf{g}^I + \mathbf{s}^I) d\tau_2\right] d\tau_1.\end{aligned} \tag{20}$$

In a typical strapdown INS the specific force is measured in the body frame, but the gravitational acceleration is commonly calculated in the Earth or the local level frame. Thus, it is reasonable to treat them separately. This separation was first carried out by Bar-Itzhack [4] in navigation computation for the terrestrial strapdown INS. Considering the guideline that "computations performed in a fast rotating coordinate system have to be performed at a fast rate even if the norm of the computed vector changes slowly," a split-coordinate computational scheme was proposed, in which only the necessary differential equations were solved in the body frame and the rest of the equations solved in a slowly rotating system. A comparison between [4] and this section is interesting and may help to better understand the development in the sequel.

The thrust velocity is defined as integration of the specific force [4, 34], i.e.,

$$\mathbf{v}_t^I = \int_0^t \mathbf{s}^I d\tau. \tag{21}$$

It is the velocity that the vehicle would have experienced if the gravitational acceleration were absent and the initial velocity were zero. Further define the gravitational velocity as the sum of the initial velocity of the vehicle and integration of the gravitational acceleration, i.e.,

$$\mathbf{v}_g^I = \dot{\mathbf{r}}^I(0) + \int_0^t \mathbf{g}^I d\tau. \tag{22}$$

The gravitational velocity is what the vehicle would have experienced if only the gravitational acceleration were present. Note that these two definitions are both given in the inertial frame I. As pointed out in [4], one has the freedom to distribute the initial velocity of the vehicle between the thrust velocity and the gravitational velocity as one wishes. However, the definition in (22) has the benefit of eliminating the otherwise necessary transformation (see [4] for explanation). Using (20), (21), and (22), the velocity of the vehicle with respect to the inertial frame is

$$\dot{\mathbf{r}}^I = \mathbf{v}_t^I + \mathbf{v}_g^I. \tag{23}$$

In the sequel, the thrust velocity and the gravitational velocity will be treated in different manners instead of the vehicle's velocity as a whole. The specific manner of separation depends on the convenience and efficiency in handling raw measurements. Take the thrust velocity for example. Because the specific force is originally measured by accelerometers in the body frame, we have good reasons to treat the thrust velocity in a so-called thrust velocity frame. By doing so, transformations of the specific force between frames are spared.

Firstly, consider the inertial frame I and the thrust velocity frame T. Suppose a unit dual quaternion $\hat{q}_{IT} = q_{IT} + \varepsilon q'_{IT}$ characterizes the general displacement of the frame T relative to the frame I. Since the angular rate vector and the specific force are both provided in the body frame, the twist is expressed in the frame T. Noting that the frame T is aligned with the frame B in axes by definition, we have $\boldsymbol{\omega}_{IT}^T = \boldsymbol{\omega}_{IB}^B$ and $q_{IT} = q_{IB}$. According to (18) and using (85), the twist expressed in the frame T is[4]

$$\begin{aligned}\hat{\boldsymbol{\omega}}_{IT}^T &= \boldsymbol{\omega}_{IT}^T + \varepsilon(\dot{\mathbf{v}}_t^T + \boldsymbol{\omega}_{IT}^T \times \mathbf{v}_t^T) = \boldsymbol{\omega}_{IT}^T + \varepsilon(q_{IT}^* \circ \dot{\mathbf{v}}_t^I \circ q_{IT}) \\ &= \boldsymbol{\omega}_{IT}^T + \varepsilon(q_{IT}^* \circ \mathbf{s}^I \circ q_{IT}) = \boldsymbol{\omega}_{IB}^B + \varepsilon \mathbf{s}^B.\end{aligned} \tag{24}$$

Using (16), the kinematic equation of $\hat{q}_{IT}$ is

$$2\dot{\hat{q}}_{IT} = \hat{q}_{IT} \circ \hat{\boldsymbol{\omega}}_{IT}^T. \tag{25}$$

The thrust velocity can be derived from the solution of the above differential equation according to (14)

$$\mathbf{v}_t^I = 2q'_{IT} \circ q_{IT}^*. \tag{26}$$

Secondly, investigate the relationship between the inertial frame I and the gravitational velocity frame G. Let the general displacement of the frame G relative to the frame I be represented by the other unit dual quaternion $\hat{q}_{IG} = q_{IG} + \varepsilon q'_{IG}$. In this case the gravitational acceleration is calculated in the Earth frame. Noting that the frame G is aligned with the frame E in axes by definition, we have $\boldsymbol{\omega}_{IG}^G = \boldsymbol{\omega}_{IE}^E$ and $q_{IG} = q_{IE}$. According to (18) and using (85), the twist

[4] Note that the "translation" is a local concept between two considered frames. True, the trust velocity is a velocity rather than a translation according to (21), when considering the relationship between the inertial frame and the body frame. However, the trust velocity frame is defined in the Nomenclature section as a new frame that is aligned with the body frame in axes. The vector from the Earth's center to its origin is exactly the thrust velocity. Regarding the relationship between the inertial frame and the thrust velocity frame, the trust velocity becomes otherwise a translation by definition. Consequently, we can describe (24) according to (18). The same story goes for the gravitational velocity in (27).

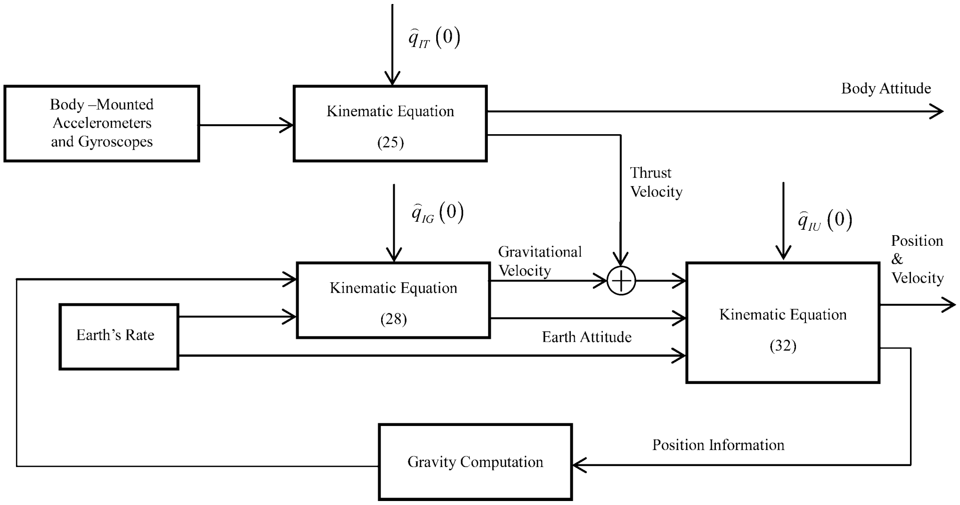


Fig. 4. Mechanism of strapdown INS algorithm based on dual quaternions.

expressed in the frame G is

$$\begin{aligned}\hat{\boldsymbol{\omega}}_{IG}^{G} &= \boldsymbol{\omega}_{IG}^{G} + \varepsilon(\dot{\mathbf{v}}_{g}^{G} + \boldsymbol{\omega}_{IG}^{G} \times \mathbf{v}_{g}^{G}) = \boldsymbol{\omega}_{IG}^{G} + \varepsilon(q_{IG}^{*} \circ \dot{\mathbf{v}}_{g}^{I} \circ q_{IG})\\ &= \boldsymbol{\omega}_{IG}^{G} + \varepsilon(q_{IG}^{*} \circ \mathbf{g}^{I} \circ q_{IG}) = \boldsymbol{\omega}_{IE}^{E} + \varepsilon \mathbf{g}^{E}. \end{aligned} \tag{27}$$

The kinematic equation of $\hat{q}_{IG}$ is

$$2\dot{\hat{q}}_{IG} = \hat{q}_{IG} \circ \hat{\boldsymbol{\omega}}_{IG}^{G}. \tag{28}$$

The gravitational velocity can be derived from the solution of differential equation (28) as

$$\mathbf{v}_{g}^{I} = 2q_{IG}' \circ q_{IG}^{*}. \tag{29}$$

Then using (23), (26), and (29), the vehicle's velocity in any form can be derived. For example, the ground speed, i.e., the vehicle's velocity relative to the Earth, is

$$\begin{aligned}\dot{\mathbf{r}}^{E} = \dot{\mathbf{r}}^{G} &= q_{IG}^{*} \circ \dot{\mathbf{r}}^{I} \circ q_{IG} + \mathbf{r}^{G} \times \boldsymbol{\omega}_{IG}^{G}\\ &= q_{IG}^{*} \circ (\mathbf{v}_{t}^{I} + \mathbf{v}_{g}^{I}) \circ q_{IG} + \mathbf{r}^{G} \times \boldsymbol{\omega}_{IG}^{G}\\ &= q_{IE}^{*} \circ (\mathbf{v}_{t}^{I} + \mathbf{v}_{g}^{I}) \circ q_{IE} + \mathbf{r}^{E} \times \boldsymbol{\omega}_{IE}^{E}. \end{aligned} \tag{30}$$

Finally, consider the inertial frame I and the position frame U. Let the general displacement of the frame U relative to the frame I be described by another unit dual quaternion $\hat{q}_{IU} = q_{IU} + \varepsilon q_{IU}'$. Noting that the frame U is also aligned with the frame E in axes by definition. According to (18) and using (85), the twist expressed in the frame U is

$$\begin{aligned}\hat{\boldsymbol{\omega}}_{IU}^{U} &= \boldsymbol{\omega}_{IU}^{U} + \varepsilon(\dot{\mathbf{r}}^{U} + \boldsymbol{\omega}_{IU}^{U} \times \mathbf{r}^{U}) = \boldsymbol{\omega}_{IU}^{U} + \varepsilon(q_{IU}^{*} \circ \dot{\mathbf{r}}^{I} \circ q_{IU})\\ &= \boldsymbol{\omega}_{IE}^{E} + \varepsilon(q_{IU}^{*} \circ (\mathbf{v}_{t}^{I} + \mathbf{v}_{g}^{I}) \circ q_{IU}). \end{aligned} \tag{31}$$

The kinematic equation of $\hat{q}_{IU}$ is

$$2\dot{\hat{q}}_{IU} = \hat{q}_{IU} \circ \hat{\boldsymbol{\omega}}_{IU}^{U}. \tag{32}$$

The vehicle's position vector in the Earth frame can be calculated from the solution of the above differential equation as

$$\mathbf{r}^{E} = \mathbf{r}^{U} = 2q_{IU}^{*} \circ q_{IU}'. \tag{33}$$

Thus far, the principle of strapdown inertial navigation has been reinterpreted and represented as three continuous kinematic equations in dual quaternion. In order to implement the functions of the strapdown INS, we have to find suitable numerical integration algorithms for a navigation computer to solve these three differential equations (25), (28), and (32).

We schematically present a block diagram in Fig. 4 to outline the fundamental mechanism of the strapdown INS algorithm based on dual quaternions. Note that the vehicle's position and velocity vectors are both generated in the Earth frame. For other applications such as the aircraft INS and tactical/strategic missile guidance, the navigation parameters usually need to be transformed to their convenient coordinate frames and the block diagrams will be slightly different from that shown in Fig. 4.

V. STRAPDOWN INERTIAL NAVIGATION ALGORITHM

In this section, a new numerical strapdown INS integration algorithm is developed based on dual quaternions that is suitable for implementation in a navigation computer.

Inspection reveals that all of those three equations (25), (28), and (32) take the same form with each other

$$2\dot{\hat{q}} = \hat{q} \circ \hat{\boldsymbol{\omega}} \tag{34}$$

where superscripts and subscripts are neglected for brevity. It is noted that discussions in this section are made with (34) assumed. Interestingly, (34) has exactly the same form as the attitude quaternion rate equation in the conventional strapdown INS

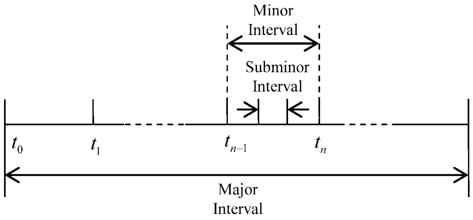


Fig. 5. Time intervals in screw algorithm.

algorithms, which means that we can reasonably introduce the mature techniques applied in attitude integration into the new algorithm. Specifically speaking, the new algorithm can be structured using the traditional two-speed approach originally developed in conventional attitude integration, in which an analytically exact equation is used at moderate speed to update dual quaternions during the major interval with input provided from a high-speed updating algorithm, which is termed the screw algorithm. In contrast to the rotation vector in the coning algorithm, the screw algorithm utilizes the screw vector [15, p. 91] to measure rectified dynamic motion during the minor interval. The relationship among the major, minor, and subminor intervals is illustrated in Fig. 5, in which a major interval is divided into a number of minor intervals, each further being divided into a number of subminor intervals.

Without loss of generality, (34) is selected to exemplify the new numerical integration algorithm suitable for all three differential equations, i.e. (25), (28), and (32).

A. Calculation of Screw Vector

Using the same method in deriving the rotation vector differential equation [43], the differential equation of the screw vector can be written as

$$\dot{\hat{\boldsymbol{\sigma}}} = \hat{\boldsymbol{\omega}} + \frac{1}{2}\hat{\boldsymbol{\sigma}} \times \hat{\boldsymbol{\omega}} + \frac{1}{\hat{\sigma}^2}\left(1 - \frac{\hat{\sigma}\sin\hat{\sigma}}{2(1-\cos\hat{\sigma})}\right)\hat{\boldsymbol{\sigma}} \times (\hat{\boldsymbol{\sigma}} \times \hat{\boldsymbol{\omega}}) \tag{35}$$

where $\hat{\boldsymbol{\sigma}}$ represents the screw vector lying on the screw axis of magnitude $\hat{\sigma} = (\hat{\boldsymbol{\sigma}} \cdot \hat{\boldsymbol{\sigma}})^{1/2}$. The last two terms in (35) are referred to as the noncommutativity components. Let $\hat{\sigma}$ be small enough and second-order accuracy (35) can be reduced to

$$\dot{\hat{\boldsymbol{\sigma}}} = \hat{\boldsymbol{\omega}} + \tfrac{1}{2}\hat{\boldsymbol{\alpha}}(t) \times \hat{\boldsymbol{\omega}}, \qquad \hat{\boldsymbol{\alpha}}(t) = \int_{t_{n-1}}^{t} \hat{\boldsymbol{\omega}}\, d\tau \tag{36}$$

which can be used to approximate the screw vector during a minor interval.

Integrating (36) during a minor interval, the screw vector $\Delta\hat{\boldsymbol{\sigma}}$ is

$$\Delta\hat{\boldsymbol{\sigma}} = \hat{\boldsymbol{\alpha}} + \frac{1}{2}\int_{t_{n-1}}^{t_n} \hat{\boldsymbol{\alpha}}(\tau) \times \hat{\boldsymbol{\omega}}\, d\tau \tag{37}$$

where $\hat{\boldsymbol{\alpha}} = \hat{\boldsymbol{\alpha}}(t_n)$ is the accumulated incremental twist during the current minor interval. The integral term in (37) is named the screw correction, denoted by $\Delta\hat{\boldsymbol{\sigma}}_s$ in the sequel.

A general algorithmic approximation to the screw correction consists of the sum of all possible cross products formed from the incremental twists for the N subminor sensor-data intervals, making up a minor computational interval. An approximation can be formulated as

$$\Delta\hat{\boldsymbol{\sigma}}_s = \sum_{i=1}^{N}\sum_{j>i}^{N} \hat{K}_{ij}(\hat{\boldsymbol{\alpha}}_i \times \hat{\boldsymbol{\alpha}}_j) \tag{38}$$

where $\hat{\boldsymbol{\alpha}}_i$ is the incremental twist over the ith minor interval; $\hat{K}_{ij}$ are constant coefficients and can be optimally determined by minimizing an error criterion under the classical screw motion, just the same as what has been done in the coning correction.[5] After a considerable development referencing the procedure for the coning correction [18–21, 23, 32, 33, 36, 43], the optimal coefficients $\hat{K}_{ij}$ for the screw correction are derived. Interestingly, the derived coefficients are exactly the same as those for the coning correction. For example when $N = 2$, the optimal coefficient $\hat{K}_{12} = 2/3$. Since the optimal coefficients have no dual parts, they are replaced by K_{ij} in the sequel for notational brevity.

The classical screw motion is characterized by the screw vector

$$\hat{\boldsymbol{\sigma}} = \hat{\sigma}\begin{bmatrix} 0 \\ \cos(\hat{\gamma}t) \\ \sin(\hat{\gamma}t) \end{bmatrix} \tag{39}$$

where the dual number $\hat{\gamma}$ is the screw frequency, analogous to the coning frequency. Then the twist describing the classical screw motion can be written as

$$\hat{\boldsymbol{\omega}} = \hat{\gamma}\begin{bmatrix} -2\sin^2\left(\dfrac{\hat{\sigma}}{2}\right) \\ -\sin(\hat{\sigma})\sin(\hat{\gamma}t) \\ \sin(\hat{\sigma})\cos(\hat{\gamma}t) \end{bmatrix}. \tag{40}$$

Considering the composition rule of the screw vector analogous to that of the rotation vector [43], the screw vector during a major interval could be estimated by the following iterative formula

$$\hat{\boldsymbol{\sigma}}_n = \hat{\boldsymbol{\sigma}}_{n-1} + \Delta\hat{\boldsymbol{\sigma}} + \tfrac{1}{2}\hat{\boldsymbol{\sigma}}_{n-1} \times \hat{\boldsymbol{\alpha}} \tag{41}$$

where $\hat{\boldsymbol{\alpha}} = \sum_{i=1}^{N}\hat{\boldsymbol{\alpha}}_i$ and the iterative result of (41) will be used to update dual quaternion as can be seen later.

[5] For the sake of length constraint, details are omitted. Interested readers can easily repeat the derivation procedure by analogy. The following quantities such as the classical screw motion and the screw frequency are both generated during this procedure.

B. Dual Quaternion Update

A general algorithm for updating dual quaternion is

$$\hat{q}(t+\Delta T) = \hat{q}(t) \circ \hat{q}(\Delta T) \tag{42}$$

where $\hat{q}(t)$ and $\hat{q}(t+\Delta T)$ are the previous and current dual quaternions, respectively, and $\hat{q}(\Delta T)$ is the updating dual quaternion during the current major interval. With (15), the updating dual quaternion $\hat{q}(\Delta T)$ can be expressed as a function of the screw vector as

$$\hat{q}(\Delta T) = \left[\cos\left(\frac{\hat{\sigma}}{2}\right), \frac{\hat{\boldsymbol{\sigma}}}{\hat{\sigma}} \sin\left(\frac{\hat{\sigma}}{2}\right)\right] \tag{43}$$

where the screw vector $\hat{\boldsymbol{\sigma}}$ during the current major interval is the resultant twist by iterating (41). Since the exact formula of (43) is too complicated to compute in real time, we need to approximate it with a simplified form. Trigonometric functions are usually approximated by truncation of trigonometric series. For example, a fourth-order approximation is utilized as

$$\begin{aligned} &\cos\left(\frac{\hat{\sigma}}{2}\right) = 1 - \frac{\hat{\sigma}^2}{8} + \frac{\hat{\sigma}^4}{384} \qquad \text{and} \\ &\frac{1}{\hat{\sigma}} \sin\left(\frac{\hat{\sigma}}{2}\right) = \frac{1}{2} - \frac{\hat{\sigma}^2}{48}. \end{aligned} \tag{44}$$

VI. ANALYTIC COMPARISONS WITH CONVENTIONAL ALGORITHMS

This section makes analytic comparisons between the new algorithm and the conventional algorithms from both theoretical and algorithmic aspects.

A. Theoretical Consistency in Continuous Form

In the sequel, it is shown that the strapdown navigation equations in dual quaternion are theoretically consistent with the conventional ones by proving that the conventional navigation mechanization in the Earth frame can be derived from three dual quaternion kinematic equations in Section IV.

Denote by $\hat{q}_{GT} = q_{GT} + \varepsilon q'_{GT}$ the relationship between the gravitational velocity frame G and the thrust velocity frame T. We have

$$\hat{q}_{GT} = \hat{q}^*_{IG} \circ \hat{q}_{IT}. \tag{45}$$

The derivative of $\hat{q}_{GT}$ with respective to time is

$$\dot{\hat{q}}_{GT} = \frac{d(\hat{q}^*_{IG})}{dt} \circ \hat{q}_{IT} + \hat{q}^*_{IG} \circ \dot{\hat{q}}_{IT}. \tag{46}$$

Considering the norm of $\hat{q}_{GT}$ is unity, we have

$$\frac{d(\hat{q}^*_{IG})}{dt} = -\hat{q}^*_{IG} \circ \dot{\hat{q}}_{IG} \circ \hat{q}^*_{IG}. \tag{47}$$

Substituting (47) into (46) yields

$$\dot{\hat{q}}_{GT} = -\hat{q}^*_{IG} \circ \dot{\hat{q}}_{IG} \circ \hat{q}^*_{IG} \circ \hat{q}_{IT} + \hat{q}^*_{IG} \circ \dot{\hat{q}}_{IT}. \tag{48}$$

With (25) and (28), (48) is written as

$$\begin{aligned} \dot{\hat{q}}_{GT} &= -\hat{q}^*_{IG} \circ (\tfrac{1}{2}\hat{\boldsymbol{\omega}}^I_{IG} \circ \hat{q}_{IG}) \circ \hat{q}^*_{IG} \circ \hat{q}_{IT} + \hat{q}^*_{IG} \circ (\tfrac{1}{2}\hat{q}_{IT} \circ \hat{\boldsymbol{\omega}}^T_{IT}) \\ &= \tfrac{1}{2}\hat{q}_{GT} \circ (\hat{\boldsymbol{\omega}}^T_{IT} - \hat{q}^*_{IT} \circ \hat{\boldsymbol{\omega}}^I_{IG} \circ \hat{q}_{IT}) \\ &= \tfrac{1}{2}\hat{q}_{GT} \circ (\hat{\boldsymbol{\omega}}^T_{IT} - \hat{\boldsymbol{\omega}}^T_{IG}) = \tfrac{1}{2}\hat{q}_{GT} \circ \hat{\boldsymbol{\omega}}^T_{GT}. \end{aligned} \tag{49}$$

Equalizing real parts on both sides, we obtain

$$2\dot{q}_{EB} = 2\dot{q}_{GT} = q_{GT} \circ \boldsymbol{\omega}^T_{GT} = q_{EB} \circ \boldsymbol{\omega}^B_{EB} \tag{50}$$

where the angular rate vector of the body frame relative to the Earth frame $\boldsymbol{\omega}^B_{EB} = \boldsymbol{\omega}^B_{IB} - \boldsymbol{\omega}^B_{IE} = \boldsymbol{\omega}^B_{IB} - q^*_{EB} \circ \boldsymbol{\omega}^E_{IE} \circ q_{EB}$. It is obvious that (50) is the right attitude quaternion rate equation in the Earth frame mechanism [40].

On the other hand, differentiating the ground speed in (30) with respect to time yields

$$\begin{aligned} \ddot{\mathbf{r}}^E = \frac{d(q^*_{IE})}{dt} \circ \dot{\mathbf{r}}^I \circ q_{IE} + q^*_{IE} \circ (\mathbf{g}^I + \mathbf{s}^I) \circ q_{IE} \\ + q^*_{IE} \circ \dot{\mathbf{r}}^I \circ \dot{q}_{IE} + \dot{\mathbf{r}}^E \times \boldsymbol{\omega}^E_{IE}. \end{aligned} \tag{51}$$

With (28) and (47) and replacing $\hat{q}_{IG}$ by $\hat{q}_{IE}$, (51) yields

$$\begin{aligned} \ddot{\mathbf{r}}^E &= -q^*_{IE} \circ \dot{q}_{IE} \circ q^*_{IE} \circ \dot{\mathbf{r}}^I \circ q_{IE} \\ &\quad + q^*_{IE} \circ (\mathbf{g}^I + \mathbf{s}^I) \circ q_{IE} + q^*_{IE} \circ \dot{\mathbf{r}}^I \circ \dot{q}_{IE} + \dot{\mathbf{r}}^E \times \boldsymbol{\omega}^E_{IE} \\ &= q^*_{IE} \circ (\mathbf{g}^I + \mathbf{s}^I) \circ q_{IE} + \dot{\mathbf{r}}^E \times \boldsymbol{\omega}^E_{IE} \\ &\quad + \tfrac{1}{2}((q^*_{IE} \circ \dot{\mathbf{r}}^I \circ q_{IE}) \circ \boldsymbol{\omega}^E_{IE} - \boldsymbol{\omega}^E_{IE} \circ (q^*_{IE} \circ \dot{\mathbf{r}}^I \circ q_{IE})) \\ &= q^*_{IE} \circ (\mathbf{g}^I + \mathbf{s}^I) \circ q_{IE} + \dot{\mathbf{r}}^E \times \boldsymbol{\omega}^E_{IE} + (q^*_{IE} \circ \dot{\mathbf{r}}^I \circ q_{IE}) \times \boldsymbol{\omega}^E_{IE}. \end{aligned} \tag{52}$$

Using (30), we have $q^*_{IE} \circ \dot{\mathbf{r}}^I \circ q_{IE} = \dot{\mathbf{r}}^E - \mathbf{r}^E \times \boldsymbol{\omega}^E_{IE}$. Thus, (52) gives

$$\begin{aligned} \ddot{\mathbf{r}}^E &= q^*_{IE} \circ (\mathbf{g}^I + \mathbf{s}^I) \circ q_{IE} + \dot{\mathbf{r}}^E \times \boldsymbol{\omega}^E_{IE} + (\dot{\mathbf{r}}^E - \mathbf{r}^E \times \boldsymbol{\omega}^E_{IE}) \times \boldsymbol{\omega}^E_{IE} \\ &= q^*_{IE} \circ (\mathbf{g}^I + \mathbf{s}^I) \circ q_{IE} + 2\dot{\mathbf{r}}^E \times \boldsymbol{\omega}^E_{IE} - (\mathbf{r}^E \times \boldsymbol{\omega}^E_{IE}) \times \boldsymbol{\omega}^E_{IE} \\ &= q_{EB} \circ \mathbf{s}^B \circ q^*_{EB} + 2\dot{\mathbf{r}}^E \times \boldsymbol{\omega}^E_{IE} + \mathbf{g}^E_l \end{aligned} \tag{53}$$

where the gravity vector is defined as $\mathbf{g}^E_l = \mathbf{g}^E - (\mathbf{r}^E \times \boldsymbol{\omega}^E_{IE}) \times \boldsymbol{\omega}^E_{IE}$. Equation (53) is exactly the rate equation of the ground speed with respect to the Earth axes [40, 41].

B. Screw Correction versus Coning/Sculling Corrections

In Section V, the coefficients K_{ij} for the screw algorithm were optimally determined by minimizing an error criterion under the classical screw motion, which in essence combines rotational motion and translational motion together. In the conventional algorithms, however, two sets of optimal coefficients have to be determined, one for the coning algorithm and the other for the sculling algorithm. Judging from this aspect, some kind of inherent interrelationship likely lies between the coning motion and the sculling motion, which is essential to explain the “duality” or “equivalence” in the algorithm structure and the algorithm error between the optimal coning and sculling compensation algorithms [22, 35]. Before we get involved in further discussions, it should be noticed that the functionalities of coning and sculling compensations have been efficiently combined by the screw algorithm for (25).

First of all, the classical screw motion described by the twist in (40) is examined. Using the two dual numbers $\hat{\sigma} = \sigma + \varepsilon\sigma'$ and $\hat{\gamma} = \gamma + \varepsilon\gamma'$ the separation of the twist into the real part and the dual part yields

$$\begin{aligned}\hat{\boldsymbol{\omega}} &= (\gamma + \varepsilon\gamma')\begin{bmatrix} -2\sin^2\left(\frac{\sigma + \varepsilon\sigma'}{2}\right) \\ -\sin(\sigma + \varepsilon\sigma')\sin((\gamma + \varepsilon\gamma')t) \\ \sin(\sigma + \varepsilon\sigma')\cos((\gamma + \varepsilon\gamma')t) \end{bmatrix} \\ &= \gamma\begin{bmatrix} -2\sin^2\left(\frac{\sigma}{2}\right) \\ -\sin(\sigma)\sin(\gamma t) \\ \sin(\sigma)\cos(\gamma t) \end{bmatrix} \\ &\quad + \varepsilon\begin{bmatrix} -2\gamma'\sin^2\left(\frac{\sigma}{2}\right) - \sigma'\gamma\sin(\sigma) \\ -(\sigma'\gamma\cos(\sigma)\sin(\gamma t) + \gamma'\sin(\sigma)\sin(\gamma t) + \gamma\gamma' t\sin(\sigma)\cos(\gamma t)) \\ \sigma'\gamma\cos(\sigma)\cos(\gamma t) + \gamma'\sin(\sigma)\cos(\gamma t) - \gamma\gamma' t\sin(\sigma)\sin(\gamma t) \end{bmatrix}. \end{aligned} \tag{54}$$

The real part of $\hat{\boldsymbol{\omega}}$ (denoted by $\boldsymbol{\omega}$) is just the angular velocity vector corresponding to the classical coning motion [21, 23, 32, 43]; the dual part of $\hat{\boldsymbol{\omega}}$ can be rewritten in the form of (18) (see Appendix D for derivation)

$$\dot{\mathbf{t}} + \boldsymbol{\omega} \times \mathbf{t} \tag{55}$$

where

$$\mathbf{t} = \begin{bmatrix} -2\gamma'\sin^2\left(\frac{\sigma}{2}\right)t \\ \sigma'\cos(\gamma t) - \gamma' t\sin(\sigma)\sin(\gamma t) \\ \sigma'\sin(\gamma t) + \gamma' t\sin(\sigma)\cos(\gamma t) \end{bmatrix}.$$

Allowing for the definition in (26), the vector $\mathbf{t}$ is referred to the thrust velocity there (see footnote 4), so its derivative $\dot{\mathbf{t}}$ represents a linear acceleration. Therefore the linear acceleration $\dot{\mathbf{t}}$ together with the angular velocity $\boldsymbol{\omega}$ describe a kind of more complicated sculling motion than usually used in designing the conventional sculling compensation algorithm [22, 35, 40]. In other words, screw motion in itself consists of the coning motion and the sculling motion. From this point of view, it can be conjectured that the optimality under the classical screw motion should spontaneously spread into the coning motion and the sculling motion, which determines the duality and equivalence between the optimal coning and sculling corrections. This conjecture will be verified in the sequel.

Next investigate how the optimal coefficients of the screw algorithm spread into the coning algorithm and the sculling algorithm. Separating the real part from the dual part of (42) yields

$$\begin{aligned} q(t + \Delta T) &= q(t) \circ q(\Delta T) \\ q'(t + \Delta T) &= q(t) \circ q'(\Delta T) + q'(t) \circ q(\Delta T). \end{aligned} \tag{56}$$

Let $\hat{\boldsymbol{\sigma}} = \boldsymbol{\sigma} + \varepsilon\boldsymbol{\sigma}'$, (43) can be expanded to

$$\begin{aligned} q(\Delta T) &= \cos\left(\frac{\sigma}{2}\right) + \frac{\boldsymbol{\sigma}}{\sigma}\sin\left(\frac{\sigma}{2}\right) \\ q'(\Delta T) &= -\frac{\sigma'}{2}\sin\left(\frac{\sigma}{2}\right) + \frac{\sigma\boldsymbol{\sigma}' - \sigma'\boldsymbol{\sigma}}{\sigma^2}\sin\left(\frac{\sigma}{2}\right) \\ &\quad + \frac{\sigma'\boldsymbol{\sigma}}{2\sigma}\cos\left(\frac{\sigma}{2}\right). \end{aligned} \tag{57}$$

It is clear from (56) and (57) that only the real part $\boldsymbol{\sigma}$ of the screw vector $\hat{\boldsymbol{\sigma}}$ contributes to the attitude updating.

Suppose $\hat{\boldsymbol{\omega}} = \boldsymbol{\omega} + \varepsilon\boldsymbol{\omega}'$ and $\hat{\boldsymbol{\alpha}}(t) = \boldsymbol{\alpha}(t) + \varepsilon\boldsymbol{\alpha}'(t)$, we obtain from (36) that

$$\boldsymbol{\alpha}(t) = \int_{t_{n-1}}^{t} \boldsymbol{\omega}\, d\tau \qquad \text{and} \qquad \boldsymbol{\alpha}'(t) = \int_{t_{n-1}}^{t} \boldsymbol{\omega}' d\tau. \tag{58}$$

Considering the twists $\hat{\boldsymbol{\omega}}$ in (25), (28), and (32), it is shown that $\boldsymbol{\alpha}(t)$ and $\boldsymbol{\alpha}'(t)$ are the accumulated incremental angular vector and the accumulated incremental velocity vector during the integral interval, respectively. With (38), the real part of $\Delta\hat{\boldsymbol{\sigma}}_s$ is

$$\sum_{i=1}^{N}\sum_{j>i}^{N} K_{ij}(\boldsymbol{\alpha}_i \times \boldsymbol{\alpha}_j) \tag{59}$$

where $\boldsymbol{\alpha}_i$ is the incremental angular vector over the ith minor interval. Apparently, (59) is no other than the optimized coning correction in conventional attitude integration [21, 23, 43].

Because the analysis for sculling compensation is generally quite involved, we, without loss of generality, confine our discussions to the single-speed-N-interval screw algorithm for exemplification in the sequel, i.e., when a major interval is equal to a minor interval and a minor interval consists of N subminor intervals. With (14) and (56), we obtain

$$\begin{aligned} \mathbf{t}^o(t + \Delta T) &= 2q'(t + \Delta T) \circ q^*(t + \Delta T) \\ &= 2(q(t) \circ q'(\Delta T) + q'(t) \circ q(\Delta T)) \circ (q(t) \circ q(\Delta T))^* \\ &= \mathbf{t}^o(t) + q(t) \circ (2q'(\Delta T) \circ q^*(\Delta T)) \circ q^*(t) \\ &= \mathbf{t}^o(t) + q(t) \circ \Delta\mathbf{t}^{N'} \circ q^*(t) \\ \mathbf{t}^N(t + \Delta T) &= 2q^*(t + \Delta T) \circ q'(t + \Delta T) \\ &= 2(q(t) \circ q(\Delta T))^* \circ (q(t) \circ q'(\Delta T) + q'(t) \circ q(\Delta T)) \\ &= q^*(\Delta T) \circ \mathbf{t}^N(t) \circ q(\Delta T) + 2q^*(\Delta T) \circ q'(\Delta T) \\ &= q^*(\Delta T) \circ \mathbf{t}^N(t) \circ q(\Delta T) + \Delta\mathbf{t}^N \end{aligned} \tag{60}$$

where

$$\Delta \mathbf{t}^{N'} = 2q'(\Delta T) \circ q^*(\Delta T) \tag{61}$$

$$\Delta \mathbf{t}^{N} = 2q^*(\Delta T) \circ q'(\Delta T). \tag{62}$$

Note that the following relationship is satisfied

$$\boldsymbol{\sigma} \cdot \boldsymbol{\sigma} + \varepsilon 2\boldsymbol{\sigma} \cdot \boldsymbol{\sigma}' = \hat{\boldsymbol{\sigma}} \cdot \hat{\boldsymbol{\sigma}} = \hat{\sigma}^2 = \sigma^2 + \varepsilon 2\sigma\sigma' \tag{63}$$

which can also be rewritten as

$$\boldsymbol{\sigma} \cdot \boldsymbol{\sigma} = \sigma^2 \qquad \text{and} \qquad \boldsymbol{\sigma} \cdot \boldsymbol{\sigma}' = \sigma\sigma'. \tag{64}$$

Substituting (57) and (64) into (61) yields

$$\begin{aligned}
\Delta \mathbf{t}^{N'} &= 2\left(-\frac{\sigma'}{2}\sin\left(\frac{\sigma}{2}\right) + \frac{\sigma\boldsymbol{\sigma}' - \sigma'\boldsymbol{\sigma}}{\sigma^2}\sin\left(\frac{\sigma}{2}\right) + \frac{\sigma'\boldsymbol{\sigma}}{2\sigma}\cos\left(\frac{\sigma}{2}\right)\right) \\
&\quad \circ \left(\cos\left(\frac{\sigma}{2}\right) - \frac{\boldsymbol{\sigma}}{\sigma}\sin\left(\frac{\sigma}{2}\right)\right) \\
&= 2\Bigg(- \frac{\sigma'}{2}\sin\left(\frac{\sigma}{2}\right)\cos\left(\frac{\sigma}{2}\right) + \frac{\sigma'\boldsymbol{\sigma} \cdot \boldsymbol{\sigma}}{2\sigma^2}\sin\left(\frac{\sigma}{2}\right)\cos\left(\frac{\sigma}{2}\right) \\
&\quad + \frac{\sigma\boldsymbol{\sigma} \cdot \boldsymbol{\sigma}' - \sigma'\boldsymbol{\sigma} \cdot \boldsymbol{\sigma}}{\sigma^3}\sin^2\left(\frac{\sigma}{2}\right) + \frac{\sigma'\boldsymbol{\sigma}}{2\sigma}\sin^2\left(\frac{\sigma}{2}\right) \\
&\quad + \frac{\sigma'\boldsymbol{\sigma}}{2\sigma}\cos^2\left(\frac{\sigma}{2}\right) + \frac{\sigma\boldsymbol{\sigma}' - \sigma'\boldsymbol{\sigma}}{\sigma^2}\sin\left(\frac{\sigma}{2}\right)\cos\left(\frac{\sigma}{2}\right) \\
&\quad - \frac{\boldsymbol{\sigma}' \times \boldsymbol{\sigma}}{\sigma^2}\sin^2\left(\frac{\sigma}{2}\right)\Bigg) \\
&= \boldsymbol{\sigma}'\frac{\sin(\sigma)}{\sigma} + \frac{(\boldsymbol{\sigma} \cdot \boldsymbol{\sigma}') \cdot \boldsymbol{\sigma}}{\sigma^2}\left(1 - \frac{\sin(\sigma)}{\sigma}\right) \\
&\quad + 2\boldsymbol{\sigma} \times \boldsymbol{\sigma}'\left(\frac{\sin(\sigma/2)}{\sigma}\right)^2.
\end{aligned} \tag{65}$$

With (37) and (38), the twist during a minor interval is

$$\begin{aligned}
\Delta\hat{\boldsymbol{\sigma}} &= \hat{\boldsymbol{\alpha}} + \sum_{i=1}^{N}\sum_{j>i}^{N} K_{ij}(\hat{\boldsymbol{\alpha}}_i \times \hat{\boldsymbol{\alpha}}_j) \\
&= \boldsymbol{\alpha} + \sum_{i=1}^{N}\sum_{j>i}^{N} K_{ij}(\boldsymbol{\alpha}_i \times \boldsymbol{\alpha}_j) \\
&\quad + \varepsilon\left(\boldsymbol{\alpha}' + \sum_{i=1}^{N}\sum_{j>i}^{N} K_{ij}(\boldsymbol{\alpha}_i' \times \boldsymbol{\alpha}_j + \boldsymbol{\alpha}_i \times \boldsymbol{\alpha}_j')\right)
\end{aligned} \tag{66}$$

where $\boldsymbol{\alpha}_i'$ is the incremental velocity vector over the ith minor interval. Under the assumption of a major interval being equal to a minor interval, the twist during a major interval in (41) is given by $\hat{\boldsymbol{\sigma}} = \Delta\hat{\boldsymbol{\sigma}}$. With (66) and trigonometric functions being approximated by first-order Taylor expansion, i.e., $\cos(\sigma/2) = 1$ and $\sin(\sigma/2) = \sigma/2$, (65) can be reduced to

$$\begin{aligned}
\Delta \mathbf{t}^{N'} &= \boldsymbol{\sigma}' + \tfrac{1}{2}\boldsymbol{\sigma} \times \boldsymbol{\sigma}' \\
&= \boldsymbol{\alpha}' + \sum_{i=1}^{N}\sum_{j>i}^{N} K_{ij}(\boldsymbol{\alpha}_i' \times \boldsymbol{\alpha}_j + \boldsymbol{\alpha}_i \times \boldsymbol{\alpha}_j') \\
&\quad + \frac{1}{2}\left(\boldsymbol{\alpha} + \sum_{i=1}^{N}\sum_{j>i}^{N} K_{ij}(\boldsymbol{\alpha}_i \times \boldsymbol{\alpha}_j)\right) \\
&\quad \times \left(\boldsymbol{\alpha}' + \sum_{i=1}^{N}\sum_{j>i}^{N} K_{ij}(\boldsymbol{\alpha}_i' \times \boldsymbol{\alpha}_j + \boldsymbol{\alpha}_i \times \boldsymbol{\alpha}_j')\right)
\end{aligned} \tag{67}$$

which is to second-order accuracy

$$\Delta \mathbf{t}^{N'} = \boldsymbol{\alpha}' + \frac{1}{2}\boldsymbol{\alpha} \times \boldsymbol{\alpha}' + \sum_{i=1}^{N}\sum_{j>i}^{N} K_{ij}(\boldsymbol{\alpha}_i' \times \boldsymbol{\alpha}_j + \boldsymbol{\alpha}_i \times \boldsymbol{\alpha}_j'). \tag{68}$$

Considering (24), (25), and (26), the last two terms in (68) are the right rotation compensation and the sculling terms in conventional velocity integration, respectively [22, (5) and (9); 36, (4), (36), (61)]. As for (62), similar conclusions can be arrived at likewise.

C. Analysis of Algorithms Errors

It should be stated that (65) is an exact formula expressing the translational component of a infinitesimal general displacement as a function of the twist, i.e., the thrust velocity for (25), the gravitational velocity for (28), and the position vector for (32). For conveniences of analysis, trigonometric functions in (65) are approximated to the fourth order as follows

$$\begin{aligned}
\Delta \mathbf{t}^{N'} &= \boldsymbol{\sigma}'\left(1 - \frac{\sigma^2}{6} + O(\sigma^4)\right) + \frac{(\boldsymbol{\sigma} \cdot \boldsymbol{\sigma}') \cdot \boldsymbol{\sigma}}{\sigma^2}\left(\frac{\sigma^2}{6} + O(\sigma^4)\right) \\
&\quad + 2\boldsymbol{\sigma} \times \boldsymbol{\sigma}'\left(\frac{1}{4} - \frac{\sigma^2}{48} + O(\sigma^4)\right)
\end{aligned} \tag{69}$$

where $O(\sigma^4)$ denotes an equivalent infinitesimal to σ^4.

First examine the conventional algorithms. Based on the development from (65)–(68), it can be figured out that the error for conventional algorithms is due to three sources: 1) approximation of the twist during a major interval in (66), 2) truncation of trigonometric series to the first order in (67), and 3) neglect of infinitesimals higher than the second order in (68). Denote the error term of the twist in (66) by $\delta\hat{\boldsymbol{\sigma}} = \delta\boldsymbol{\sigma} + \varepsilon\delta\boldsymbol{\sigma}'$, then (67) can be written as

$$\Delta\tilde{\mathbf{t}}^{N'} = \boldsymbol{\sigma}' + \delta\boldsymbol{\sigma}' + \tfrac{1}{2}(\boldsymbol{\sigma} + \delta\boldsymbol{\sigma}) \times (\boldsymbol{\sigma}' + \delta\boldsymbol{\sigma}'). \tag{70}$$

Subtracting (69) from (70) gives

$$\begin{aligned}\Delta\tilde{\mathbf{t}}^{N'} - \Delta\mathbf{t}^{N'} &= \boldsymbol{\sigma}'\left(\frac{\sigma^2}{6}+O(\sigma^4)\right)\\ &\quad + \frac{(\boldsymbol{\sigma}\cdot\boldsymbol{\sigma}')\cdot\boldsymbol{\sigma}}{\sigma^2}\left(-\frac{\sigma^2}{6}+O(\sigma^4)\right)\\ &\quad + 2\boldsymbol{\sigma}\times\boldsymbol{\sigma}'\left(\frac{\sigma^2}{48}+O(\sigma^4)\right)+\delta\boldsymbol{\sigma}'\\ &\quad + \tfrac{1}{2}(\boldsymbol{\sigma}\times\delta\boldsymbol{\sigma}'+\delta\boldsymbol{\sigma}\times\boldsymbol{\sigma}'+\delta\boldsymbol{\sigma}\times\delta\boldsymbol{\sigma}')\end{aligned} \tag{71}$$

the magnitude of which is

$$\delta_{\mathrm{CA}} = O(\sigma'\sigma^2)+O(\delta\sigma') \tag{72}$$

where $\delta\sigma'$ satisfies (64), i.e. $\delta\boldsymbol{\sigma}\cdot\delta\boldsymbol{\sigma}' = \delta\sigma\delta\sigma'$, and the subscript CA stands for conventional algorithms.

Taking into account the fact that the twist during a major interval $\hat{\boldsymbol{\sigma}}$ is of the same order in magnitude as the incremental velocity vector $\hat{\boldsymbol{\alpha}}$ in (66), errors incurred by neglecting infinitesimals higher than the second order during the development from (67)–(68) are equivalent infinitesimal to $\sigma'\sigma^2$. Therefore, the approximation error of the translational component for CAs can be approximately quantified by (72).

On the other hand, the error for the new algorithm is due to two sources: 1) approximation of the twist during a major interval in (66), and 2) truncation of trigonometric series in (43). Note that the latter error source is the main difference between the new algorithm and the conventional algorithms. With (65), we have for fourth-order truncation of trigonometric series in (44) that

$$\begin{aligned}\Delta\tilde{\mathbf{t}}^{N'} &= (\boldsymbol{\sigma}'+\delta\boldsymbol{\sigma}')\left(1-\frac{(\sigma+\delta\sigma)^2}{6}\right)\\ &\quad + \frac{((\boldsymbol{\sigma}+\delta\boldsymbol{\sigma})\cdot(\boldsymbol{\sigma}'+\delta\boldsymbol{\sigma}'))\cdot(\boldsymbol{\sigma}+\delta\boldsymbol{\sigma})}{6}\\ &\quad + 2(\boldsymbol{\sigma}+\delta\boldsymbol{\sigma})\times(\boldsymbol{\sigma}'+\delta\boldsymbol{\sigma}')\left(\frac{1}{4}-\frac{(\sigma+\delta\sigma)^2}{48}\right).\end{aligned} \tag{73}$$

After some more tedious development than when deriving (71) and (72), it can be shown that

$$\delta_{\mathrm{DQ}} = O(\sigma'\sigma^4)+O(\delta\sigma') \tag{74}$$

where the subscript DQ stands for the new algorithm based on dual quaternions.

Comparing (72) with (74) indicates that the new algorithm may be superior to conventional ones in accuracy of the translational component (the velocity vectors for (25) and (28), the position vector for (32)). This conjecture is based on the fact that the magnitude of twist $\hat{\sigma}$ during a major interval is far less than unity. For example, if $\delta\sigma'$ were less than $\sigma'\sigma^2$ during a major interval, the new algorithm would have the upper hand in accuracy. This scenario is not just a hypothesis. For large magnitude motion arising from vehicle maneuvers, $\sigma'\sigma^2$ (or $\sigma'\sigma^4$) is likely greater than $\delta\sigma'$ because of the cubic (or quintic) increase. In other words, the new algorithm is more robust in coping with vehicle maneuvers.

Whereas it is nontrivial to figure out the degree of superiority (DOS) of the new algorithm because as shown in (72) and (74), the degree depends on the exact value of the relative magnitude of the true twist $\hat{\sigma}$ to that of the error twist $\delta\hat{\sigma}$ during each major interval. Furthermore, the exact value of the twists can hardly be known in a computer simulation, let alone in real conditions. On the other hand, however, the superiority will likely be noticeable because higher than first-order truncation of trigonometric series is indeed effective in reducing the approximation error [40, p. 299, Table 10.1]. In other words, the DOS depends on the truncation order of trigonometric series in (43). This judgment will be validated by simulations in the sequel.

Although the new algorithm is no better than the conventional ones as far as the attitude accuracy during a major interval in (59) is concerned, the superiority in velocity or position will be spread out into the whole dead-reckoning algorithm along with cycle-by-cycle runs of the navigation computer.

VII. NUMERICAL SIMULATIONS

To justify the proposed algorithm based on dual quaternions, a variety of simulations of inertial navigation are performed using the Simulink toolbox in Matlab 6.5. At the same time, we also simulate the CA in the navigation frame (or the local-level frame) mechanism based upon [36, 37] under the same conditions. The local-level frame is defined with x directing north and y directing upward vertical.

In the simulation, we use an ideal trace generator previously built for testing the CA, which feeds on two inputs: the ground speed rate and the gimbal angle rate, to produce gyro/accelerometer outputs as well as the vehicle's navigation parameters, namely, attitude angle, velocity vector and position. The variable input configuration for the ideal trace generator is shown in Table I. The frequency of the input signals varies in a wide range from 0.01 to 1 Hz. Specifically a list of discrete frequencies is selected and carried out in turn to generate the reference trajectory, i.e., $f = 0.01$, 0.02, 0.04, 0.06, 0.08, 0.1, 0.2, 0.4, 0.6, and 0.8 Hz, respectively. The measurements of gyroscopes and accelerometers are generated at a fixed frequency of 100 Hz. For the new algorithm, the single-speed screw algorithms for (25), (28), and (32) are shown in Table II. Allowing that the gravity model in the trace generator is given in the local-level frame, the gravitation for (27) is calculated in the local-level frame and then transformed to the

TABLE I
Inputs Configuration for Trace Generator

	Ground Speed Rate	Gimbal Angle Rate
	(m/s^2)	(rad/s)
Input	North/Vertical/East	Roll/Yaw/Pitch
Value	$20\sin(2\pi f t)$	$2\pi f\cos(2\pi f t)$

Note: f is the angular/linear frequency of input signals.

TABLE II
Screw Algorithms

Algorithm Equation	(25)	(28)	(32)
Intervals	2-interval	1-interval	1-interval
Execution Rate (Hz)	50	50	50

Earth frame for the sake of consistency. For a gravity model expressed in the Earth frame, readers can refer to [2, p. 145, (7.39)] and [41]. For the conventional algorithm, a typical trapezoidal integration is applied into position updating.

In the simulations, the following scenario is assumed. At the very start, the vehicle stands still on the surface of the Earth at longitude 110° and latitude 30°. The attitude angles are all zero. In order to make comparisons, the simulation result of the new algorithm is first transformed from the Earth frame to the local level frame, followed by subtracting true navigation parameters produced by the trace generator to obtain the error. The transformation from the Earth frame to the local level frame is implemented using the iterative algorithm proposed in [42].

A. Simulations with Ideal Inertial Sensors

Now we examine the performance of the new algorithm under ideal conditions, neglecting the errors incurred by nonideal gyroscopes and accelerometers. The total duration of simulation is 3600 s. The maximum absolute errors (MAE) of both the new algorithms and CAs are shown in Fig. 6 as a function of the ratio of the varying input signal frequency f to the fixed algorithm execution rate 50 Hz. In addition, we also carry out a simplified version of the new algorithm with first-order truncation of trigonometric series, i.e., let $\cos(\hat{\sigma}/2) = 1$ and $\sin(\hat{\sigma}/2) = \hat{\sigma}/2$. In Fig. 6, “DQ Simplified” stands for the simplified new algorithm with first-order truncation of trigonometric series, “DQ” the normal new algorithm with fourth-order approximation, “CA” the conventional algorithm. For the sake of quantitative evaluation, the DOS of an algorithm over the CA is defined as the negative logarithmic ratio of the algorithm’s MAE to the CAs, i.e.

$$\text{DOS of an algorithm} = -\log 10\left(\frac{\text{the algorithm's MAE}}{\text{conventional algorithm's MAE}}\right).$$

The result in Fig. 6 shows that under ideal conditions the normal new algorithm has substantial improvements in accuracy (two orders at least) over the CA. On the other hand, the simplified new algorithm is comparable in accuracy with the CA. These observations agree with the analytic conclusions drawn in the subsection C of Section VI.

Because the initial alignment errors and the inertial sensors errors are both set identically to zero, the navigation errors in this case are exclusively due to the computational errors, e.g. as a result of 1) limited computational frequency, 2) truncation of the mathematical functions, and 3) limitations on the order of numerical integration schemes. In the MAE curve of the normal new algorithm, there is an obvious turning frequency point at the input signal frequency $f = 0.1$ Hz (indicated by the solid arrows), to the right of which the MAE climbs quickly. This phenomenon occurs as a result of the fixed computational frequency 50 Hz. When the input frequency is less than the turning frequency, the algorithm execution rate is fast enough to cope with the vehicle maneuvers[6] and the computational errors are dominated by the last two sources. On the other hand, when the input frequency exceeds the turning frequency, the errors due to the algorithm’s incapability to compensate for the vehicle’s high maneuvers play the leading role. There is also a turning frequency for the CA that is located approximately between $f = 0.02$ Hz and $f = 0.06$ Hz (see the dashed arrows). But it is not as apparent as the one for the normal new algorithm.

B. Simulations with Nonideal Inertial Sensors

In real applications, the inertial sensor measurements are inevitably contaminated by errors, e.g. the random bias and the scale factor errors. Here for simplification we model the measurement errors as a compound effect of the fixed bias and the Gaussian random noise. In order to fully evaluate the new algorithm, we set up four typical configurations used for various grades of navigation systems. For example, the first configuration or even higher is necessary for a strategic missile. The details of configuration are listed in Table III. For each test, the same list of input signal frequencies is carried out to provide the reference trajectory. The total simulation time in this case is shortened to 360 s for the sake of reducing simulation burden. The simulation result for each test configuration is illustrated in Figs. 7–10, respectively. The result of the simplified new algorithm is omitted for clarity.

[6] Note that the input frequency is directly related to the vehicle’s maneuvers.

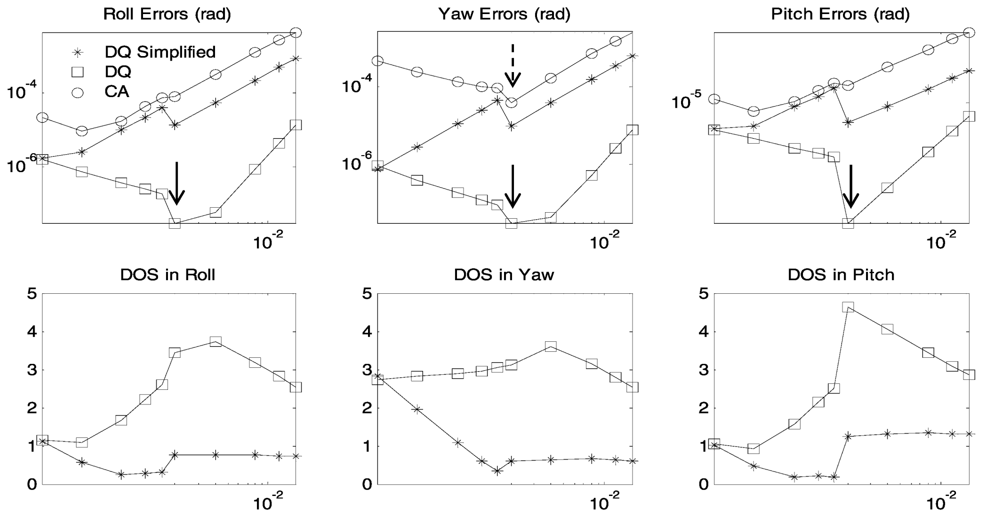


Fig. 6(a). (with ideal inertial sensors). MAEs (log-log plot) and DOS (semilog plot in *x* axis) in Euler angles as function of ratio of varying input signal frequency to fixed algorithm execution rate. (Rotation order from local-level frame to body frame: first *y* axis (yaw), then *z* axis (pitch), and finally *x* axis (roll).)

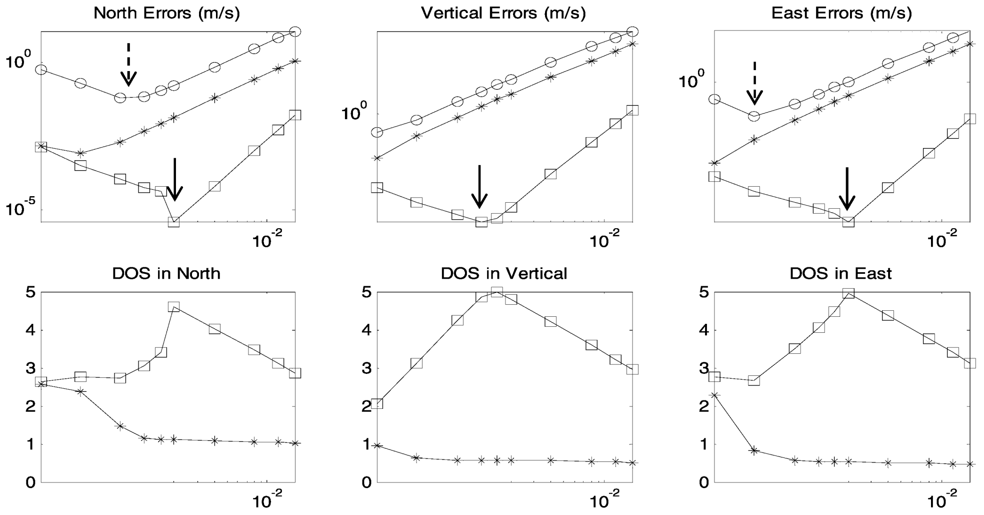


Fig. 6(b). (with ideal inertial sensors). MAEs (log-log plot) and DOS (semilog plot in *x* axis) in velocity as function of ratio of varying input signal frequency to fixed algorithm execution rate.

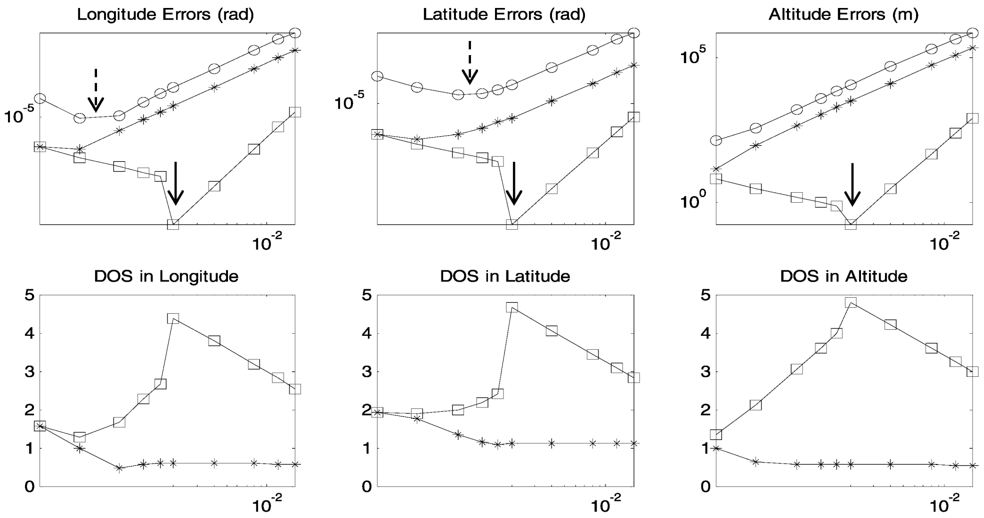


Fig. 6(c). (with ideal inertial sensors). MAEs (log-log plot) and DOS (semilog plot in *x* axis) in position as function of ratio of varying input signal frequency to fixed algorithm execution rate.

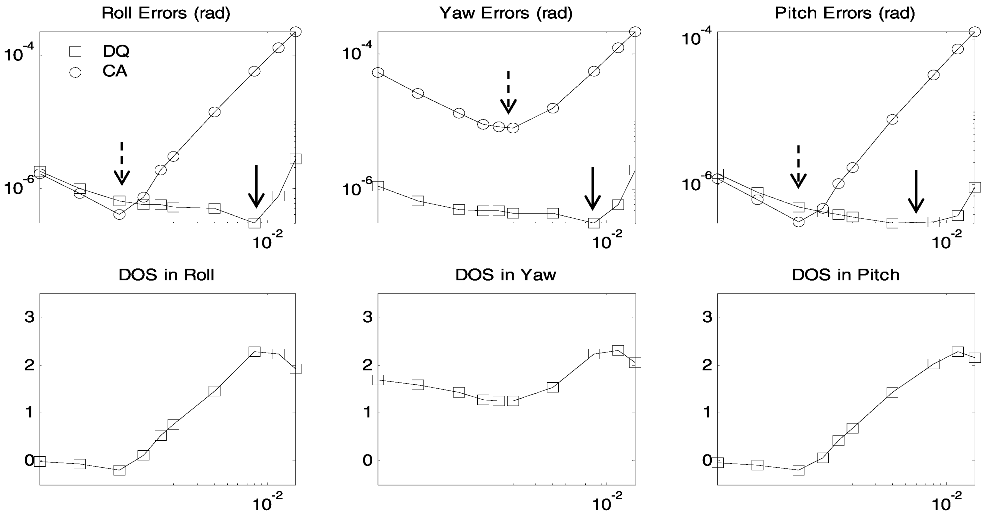


Fig. 7(a). (with nonideal inertial sensors: test #1.) MAEs (log-log plot) and DOS (semilog plot in *x* axis) in Euler angles as function of ratio of varying input signal frequency to fixed algorithm execution rate. (Rotation order from local-level frame to body frame: first *y* axis (yaw), then *z* axis (pitch), and finally *x* axis (roll).)

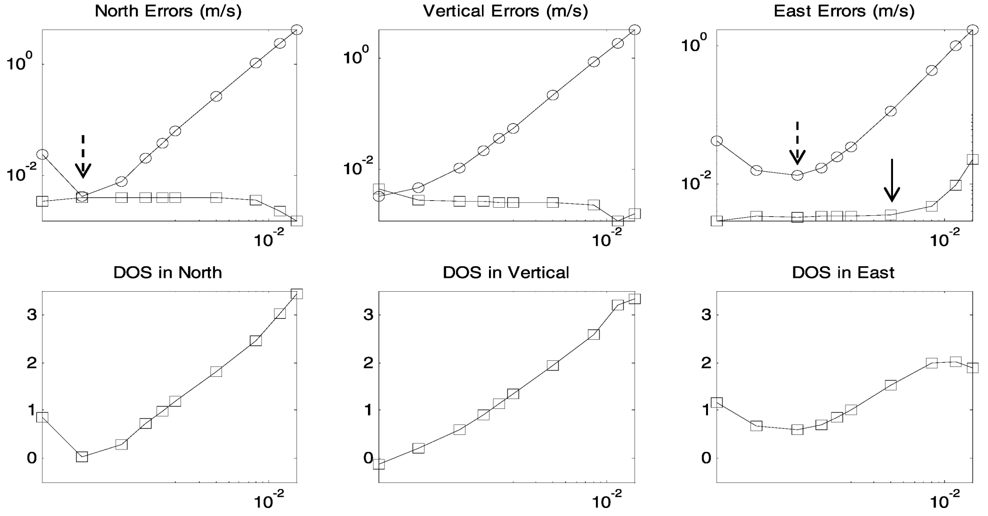


Fig. 7(b). (with nonideal inertial sensors: test #1.) MAEs (log-log plot) and DOS (semilog plot in *x* axis) in velocity as function of ratio of varying input signal frequency to fixed algorithm execution rate.

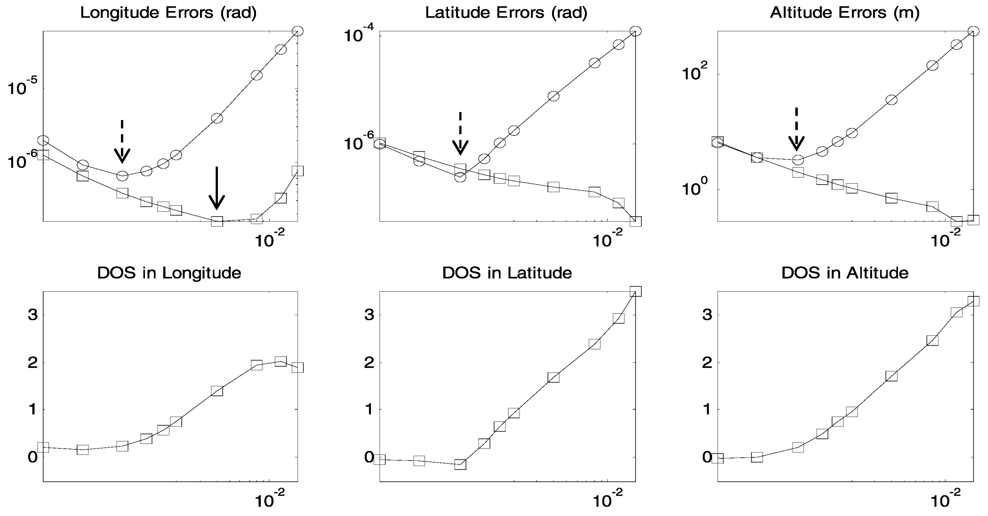


Fig. 7(c). (with nonideal inertial sensors: test #1.) MAEs (log-log plot) and DOS (semilog plot in *x* axis) in position as function of ratio of varying input signal frequency to fixed algorithm execution rate.

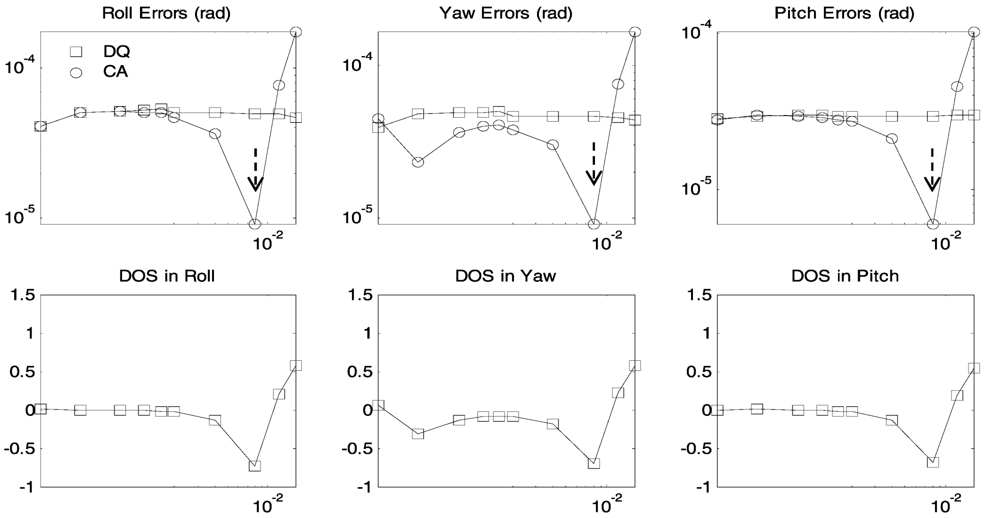


Fig. 8(a). (with nonideal inertial sensors: test #2.) MAEs (log-log plot) and DOS (semilog plot in *x* axis) in Euler angles as function of ratio of varying input signal frequency to fixed algorithm execution rate. (Rotation order from local-level frame to body frame: first *y* axis (yaw), then *z* axis (pitch), and finally *x* axis (roll).)

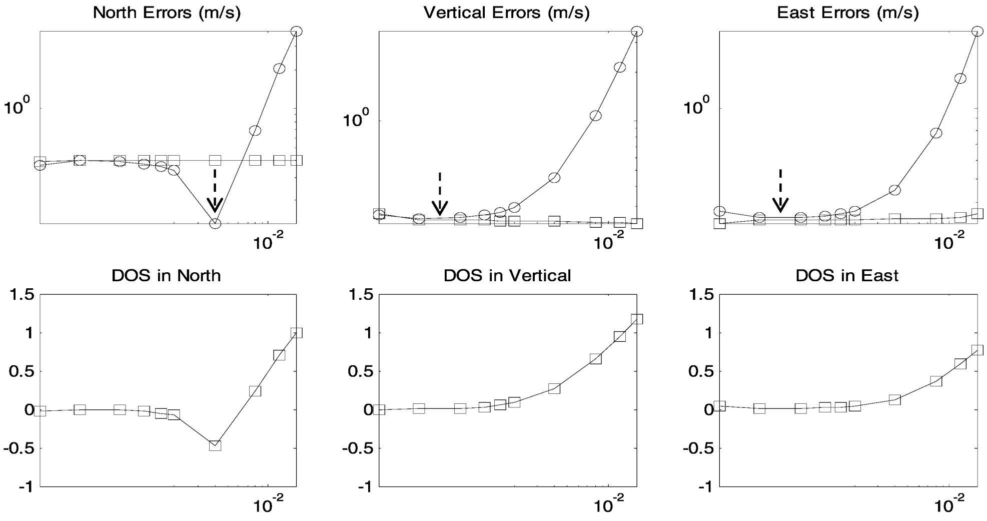


Fig. 8(b). (with nonideal inertial sensors: test #2.) MAEs (log-log plot) and DOS (semilog plot in *x* axis) in velocity as function of ratio of varying input signal frequency to fixed algorithm execution rate.

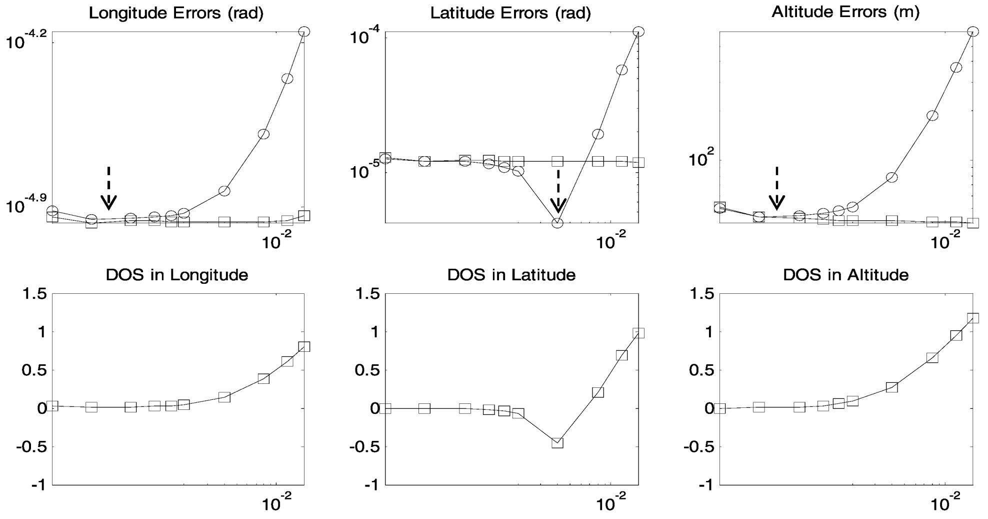


Fig. 8(c). (with nonideal inertial sensors: test #2.) MAEs (log-log plot) and DOS (semilog plot in *x* axis) in position as function of ratio of varying input signal frequency to fixed algorithm execution rate.

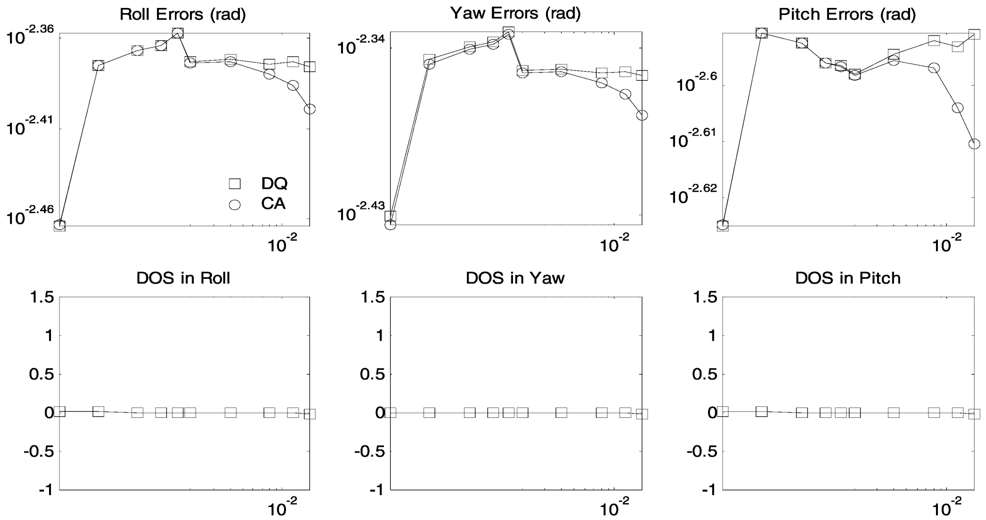


Fig. 9(a). (with nonideal inertial sensors: test #3.) MAEs (log-log plot) and DOS (semilog plot in x axis) in Euler angles as function of ratio of varying input signal frequency to fixed algorithm execution rate. (Rotation order from local-level frame to body frame: first y axis (yaw), then z axis (pitch), and finally x axis (roll).)

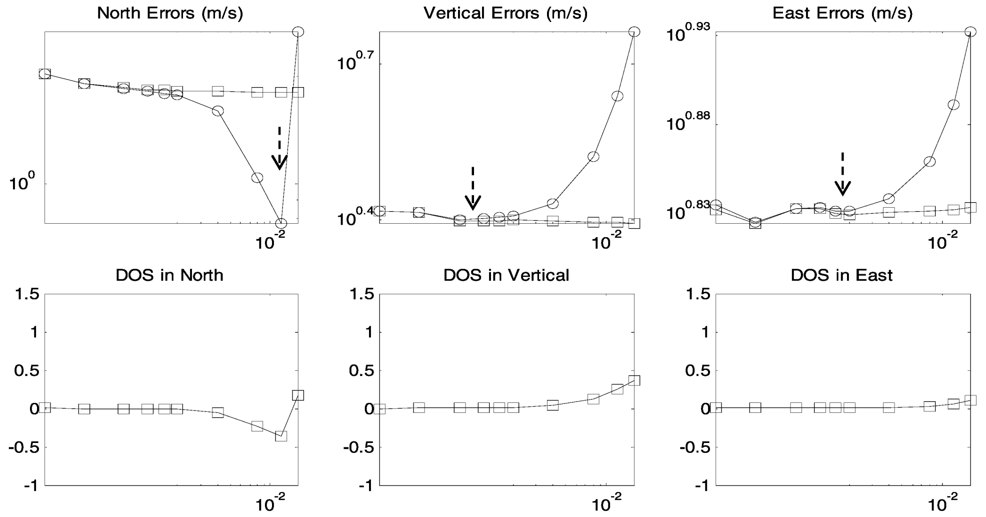


Fig. 9(b). (with nonideal inertial sensors: test #3.) MAEs (log-log plot) and DOS (semilog plot in x axis) in velocity as function of ratio of varying input signal frequency to fixed algorithm execution rate.

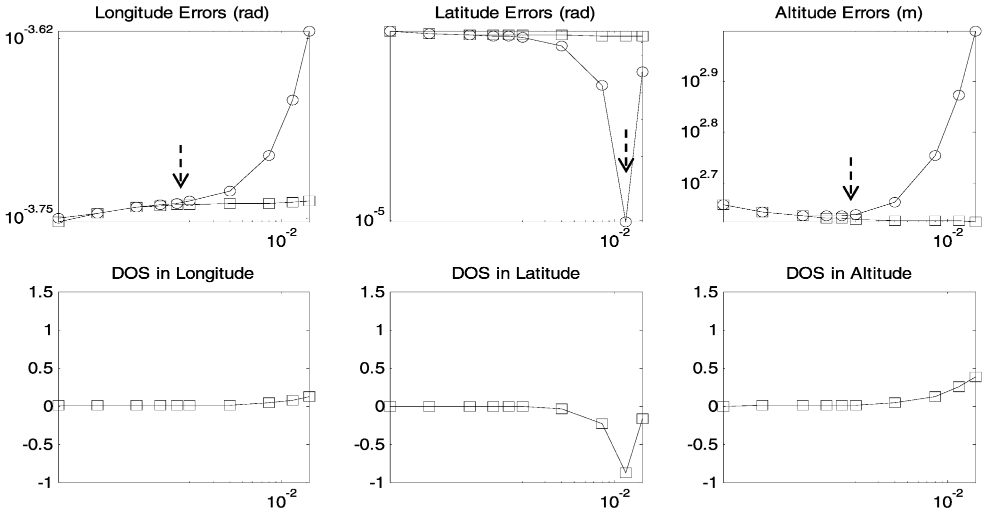


Fig. 9(c). (with nonideal inertial sensors: test #3.) MAEs (log-log plot) and DOS (semilog plot in x axis) in position as function of ratio of varying input signal frequency to fixed algorithm execution rate.

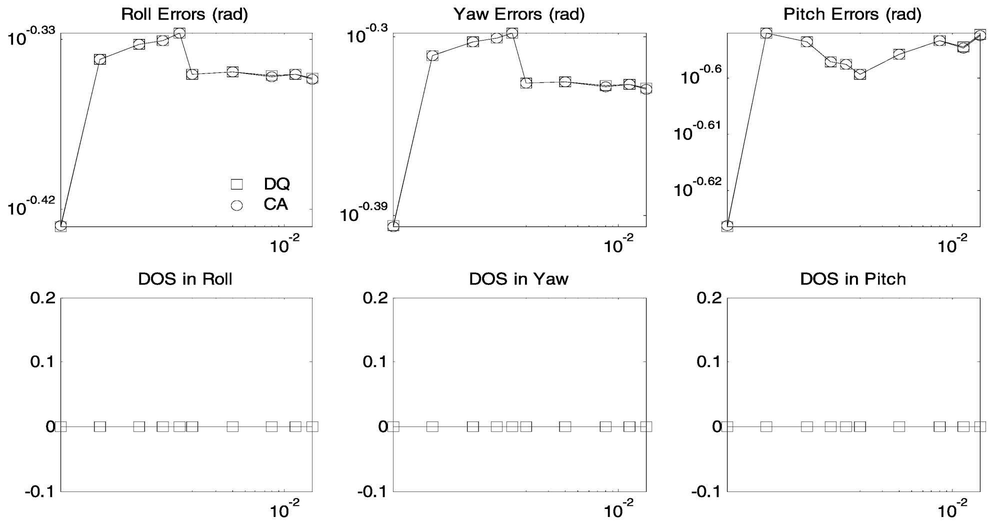


Fig. 10(a). (with nonideal inertial sensors: test #4.) MAEs (log-log plot) and DOS (semilog plot in *x* axis) in Euler angles as unction of ratio of varying input signal frequency to fixed algorithm execution rate. (Rotation order from local-level frame to body frame: first *y* axis (yaw), then *z* axis (pitch), and finally *x* axis (roll).)

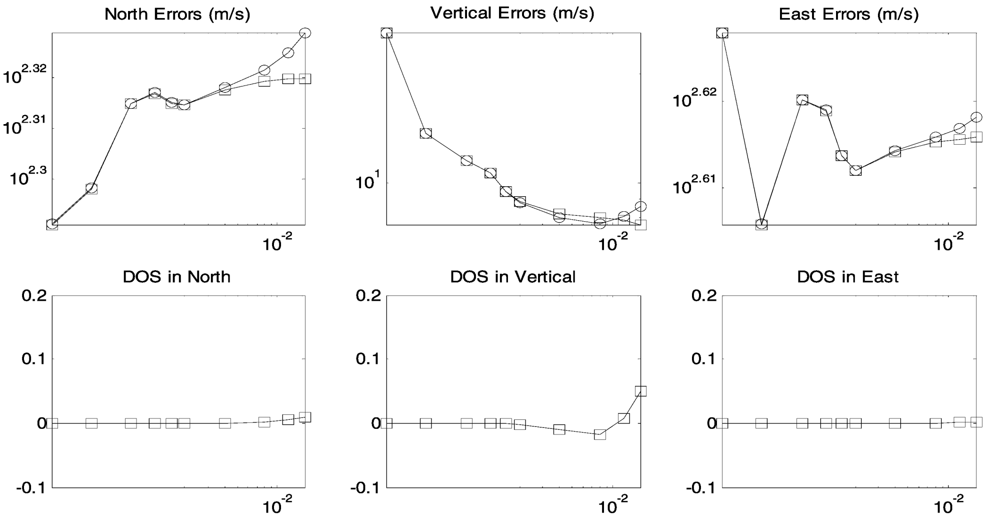


Fig. 10(b). (with nonideal inertial sensors: test #4.) MAEs (log-log plot) and DOS (semilog plot in *x* axis) in velocity as function of ratio of varying input signal frequency to fixed algorithm execution rate.

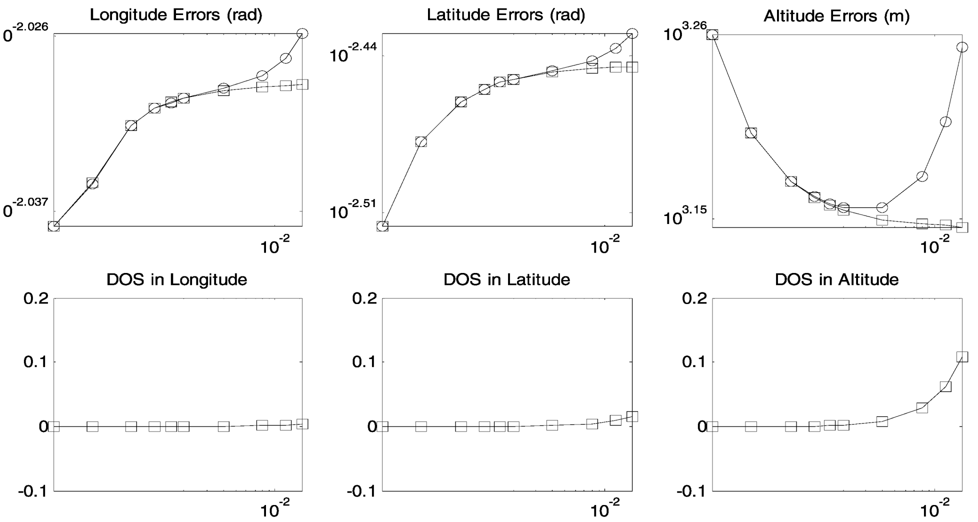


Fig. 10(c). (with nonideal inertial sensors: test #4.) MAEs (log-log plot) and DOS (semilog plot in *x* axis) in position as function of ratio of varying input signal frequency to fixed algorithm execution rate.

TABLE III
Configuration of Inertial Sensors

	Gyroscopes		Accelerometers	
		Gaussian		Gaussian
Inertial Sensors	Fixed Bias (°/h)	Noise (°/h 1σ)	Fixed Bias (μg)	Noise (μg)
Test #1	10^{-4}	10^{-5}	1	10^{-1}
Test #2	10^{-2}	10^{-3}	10^{2}	10
Test #3	1	10^{-1}	10^{3}	10^{2}
Test #4	10^{2}	10	10^{4}	10^{3}

Note: For simplification the magnitude of gravity g takes 10 m/s, in contrast to the gravity vector in Fig. 4 that should be calculated in real time.

TABLE IV
Turning Frequencies of New and Conventional Algorithms

		Turning Frequency (Hz)	
Tests		New Algorithm	Conventional Algorithm
With Ideal Sensors		0.1	0.02–0.06
With Non-Ideal Sensors	Test #1	0.4	0.04
	Test #2	> 1	0.02–0.6
	Test #3	> 1	0.1–0.6
	Test #4	> 1	≈ 1

In Fig. 7, the superiority of the new algorithm is still remarkable. The turning frequency of the new algorithm moves right to about $f = 0.4$ Hz, to the right of which the computational errors gradually get the upper hand over those incurred by the nonideal inertial sensors. Whereas the turning frequency of the CA stays at $f = 0.04$ Hz. As shown in Figs. 8–10, the observation that the new algorithm's MAE curve left to the turning frequency is relatively flat shows that the inertial sensor errors are dominating the simulation result. This is also the reason why the conventional algorithm's MAE curve left to its turning frequency is basically coincident with that of the new algorithm.

In Figs. 8–10, the turning frequency of the new algorithm has completely disappeared from the right. As for the CA, the turning frequency moves right to between $f = 0.02$ Hz and $f = 0.6$ Hz in Fig. 8, to between $f = 0.1$ Hz and $f = 0.6$ Hz in Fig. 9 and approaches $f = 1$ Hz in Fig. 10. The turning frequencies of both algorithms are summarized and listed in Table IV, which shows that the new algorithm is indeed more insensitive to and more robust in coping with the vehicle maneuvers. It is obvious that the new algorithm is sometimes inferior to some extent (less than one order) within a narrow band nearby the turning frequency of the CA, e.g. pitch errors in Fig. 8, north velocity errors in Fig. 9.

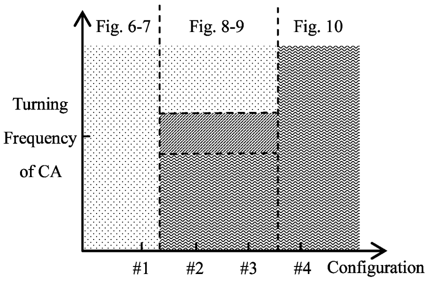


Fig. 11. Guidelines in choosing suitable algorithm based on inertial sensors configuration and turning frequency of CA.

Before the end of this section, we would like to provide our guidelines in choosing a suitable algorithm based on the inertial sensors configuration and the turning frequency of the conventional algorithm, as shown in Fig. 11. The new algorithm is recommended in the dot-filled area because it has substantially improved performance in accuracy for systems equipped with high-precision inertial sensors (see Figs. 6–7) and more robust in coping with the vehicle maneuvers (see parts of Figs. 8–9); the CA is recommended in the slash-filled area because it outperforms the new algorithm within a narrow band near its turning frequency (see parts of Figs. 8–9); and both are competent in the wave-filled area since there is hardly significant differences in accuracy between them in these cases (see Fig. 10 and parts of Figs. 8–9).

The Simulink implementation programs of both algorithms are available at website http://yuanxinwu.vip.sina.com. Interested readers are welcome to conduct test by themselves.

VIII. CONCLUSIONS

Motivated by [10], this paper proposes and examines a new algorithm based on dual quaternions, which is a most concise and efficient mathematical tool to treat the screw motion, the general displacement of a rigid body.

After a brief introduction to quaternion, dual number, and dual quaternion, the continuous differential equations for strapdown inertial navigation are derived using the tool of dual quaternion from scratch. All the equations are found out to be of the same form as the attitude quaternion differential equation in CAs. In order to efficiently solve these differential equations, a single algorithm is structured using the traditional two-speed approach originally developed in conventional attitude integration. The new algorithm combines all of the conventional coning, sculling, and scrolling corrections into a single algorithm.

Analytic comparisons indicate that the strapdown navigation equations in dual quaternion are theoretically consistent with the conventional ones. Moreover, it is revealed that the screw motion is essentially compounded of the coning and the sculling motion, which explains the duality and equivalence between the coning algorithm and the sculling algorithm. The superiority of the new algorithm over conventional ones in accuracy is analyzed. The analysis also shows that the new algorithm is promising in coping with vehicle maneuvers with more robustness.

A variety of simulations have been carried out to support the analytic conclusions, including those with ideal inertial sensors and those with nonideal inertial sensors. The numerical results agree well with the analyses. The new algorithm turns out to be a better choice than the CA for high-precision navigation systems and high-maneuver applications. Several guidelines in choosing a suitable navigation algorithm are also provided based on the inertial sensors configuration and the turning frequency of the conventional algorithm.

Quite recently, Soloviev [38] has implemented the conventional strapdown algorithm in frequency domain rather than in the customary time domain. It is claimed that the frequency-domain approach has a significant improvement in the ability to reduce the noncommutativity errors incurred by the coning and sculling motion. Redesigning the dual quaternion-based algorithm proposed here in the frequency domain would be rather interesting.

APPENDIX

A. Derivation of (13)

A Plücker line, expressed in the frame O as $\hat{\mathbf{l}}^O = \mathbf{l}^O + \varepsilon\mathbf{m}^O$, can be reexpressed in the frame N as $\hat{\mathbf{l}}^N = \mathbf{l}^N + \varepsilon\mathbf{m}^N$ where

$$\begin{aligned} \mathbf{l}^N &= q^* \circ \mathbf{l}^O \circ q \\ \mathbf{m}^N &= \mathbf{p}^N \times \mathbf{l}^N = (q^* \circ \mathbf{p}^O \circ q - \mathbf{t}^N) \times (q^* \circ \mathbf{l}^O \circ q) \\ &= q^* \circ \mathbf{m}^O \circ q - \mathbf{t}^N \times (q^* \circ \mathbf{l}^O \circ q) = q^* \circ \mathbf{m}^O \circ q \\ &\quad + \tfrac{1}{2}(\mathbf{t}^{N*} \circ q^* \circ \mathbf{l}^O \circ q + q^* \circ \mathbf{l}^O \circ q \circ \mathbf{t}^N). \end{aligned} \tag{75}$$

Define a new quaternion $q' = \frac{1}{2}q \circ \mathbf{t}^N$ and a dual quaternion $\hat{q} = q + \varepsilon q'$, it can be shown that (75) is equivalent to

$$\mathbf{l}^N + \varepsilon\mathbf{m}^N = (q + \varepsilon q')^* \circ (\mathbf{l}^O + \varepsilon\mathbf{m}^O) \circ (q + \varepsilon q'). \tag{76}$$

That is

$$\hat{\mathbf{l}}^N = \hat{q}^* \circ \hat{\mathbf{l}}^O \circ \hat{q} \tag{77}$$

where the dual quaternion is

$$\begin{aligned} \hat{q} &= q + \varepsilon q' = q + \varepsilon\tfrac{1}{2}q \circ \mathbf{t}^N = q + \varepsilon\tfrac{1}{2}q \circ q^* \circ \mathbf{t}^O \circ q \\ &= q + \varepsilon\tfrac{1}{2}\mathbf{t}^O \circ q. \end{aligned} \tag{78}$$

B. Derivation of (15)

Without loss of generality, (15) is derived using $\hat{q} = q + \varepsilon\frac{1}{2}\mathbf{t}^O \circ q$. Note that the superscript of O is neglected for brevity.

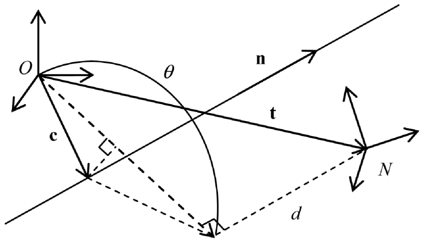


Fig. 12. Geometry of screw motion: every screw motion can be modeled as rotation with angle θ about axis at point $\mathbf{c}$ with direction $\mathbf{n}$ and subsequent translation d along axis.

The geometry of the screw motion is illustrated as shown in Fig. 12. Point $\mathbf{c}$, the projection of the origin of frame O onto the screw axis $\mathbf{n}$ [30, p. 20, (1.65)], is given by

$$\mathbf{c} = \frac{1}{2}\left(\mathbf{t} - d\mathbf{n} + \cot\left(\frac{\theta}{2}\right)\mathbf{n} \times \mathbf{t}\right) \tag{79}$$

where the pitch of the screw motion $d = \mathbf{t} \cdot \mathbf{n}$. Equation (79) can also be derived geometrically from Fig. 12. It can be verified using (79) that

$$\begin{aligned} &\sin\left(\frac{\theta}{2}\right)\mathbf{c} \times \mathbf{n} + \frac{d}{2}\cos\left(\frac{\theta}{2}\right)\mathbf{n} \\ &\quad = \frac{1}{2}\left(\sin\left(\frac{\theta}{2}\right)\mathbf{t} \times \mathbf{n} + \cos\left(\frac{\theta}{2}\right)\mathbf{t}\right). \end{aligned} \tag{80}$$

Considering

$$q = \left[\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right)\mathbf{n}\right]$$

and using (80), we have

$$\begin{aligned} \hat{q} &= q + \varepsilon\frac{1}{2}\mathbf{t} \circ q = \left[\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right)\mathbf{n}\right] \\ &\quad + \varepsilon\frac{1}{2}\left[-d\sin\left(\frac{\theta}{2}\right), \cos\left(\frac{\theta}{2}\right)\mathbf{t} + \sin\left(\frac{\theta}{2}\right)\mathbf{t} \times \mathbf{n}\right] \\ &= \left[\cos\left(\frac{\theta}{2}\right) - \varepsilon\frac{d}{2}\sin\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right)\mathbf{n}\right. \\ &\quad \left. + \varepsilon\left(\frac{d}{2}\cos\left(\frac{\theta}{2}\right)\mathbf{n} + \sin\left(\frac{\theta}{2}\right)\mathbf{c} \times \mathbf{n}\right)\right] \\ &= \left[\cos\left(\frac{\hat{\theta}}{2}\right), \sin\left(\frac{\hat{\theta}}{2}\right)\hat{\mathbf{n}}\right]. \end{aligned} \tag{81}$$

C. Derivation of (16)

Using (13), the rate of change of a dual quaternion with respect to time is given by

$$2\dot{\hat{q}} = 2\dot{q} + \varepsilon(\dot{\mathbf{t}}^O \circ q + \mathbf{t}^O \circ \dot{q}). \tag{82}$$

Substituting $\dot{q}$ by (4) gives

$$\begin{aligned} 2\dot{\hat{q}} &= \boldsymbol{\omega}_{ON}^{O} \circ q + \varepsilon(\dot{\mathbf{t}}^{O} \circ q + \tfrac{1}{2}\mathbf{t}^{O} \circ \boldsymbol{\omega}_{ON}^{O} \circ q) \\ &= \boldsymbol{\omega}_{ON}^{O} \circ q + \varepsilon(\dot{\mathbf{t}}^{O} \circ q + (\mathbf{t}^{O} \times \boldsymbol{\omega}_{ON}^{O}) \circ q + \tfrac{1}{2}\boldsymbol{\omega}_{ON}^{O} \circ \mathbf{t}^{O} \circ q) \\ &= (\boldsymbol{\omega}_{ON}^{O} + \varepsilon(\dot{\mathbf{t}}^{O} + \mathbf{t}^{O} \times \boldsymbol{\omega}_{ON}^{O})) \circ (q + \varepsilon\tfrac{1}{2}\mathbf{t}^{O} \circ q) \doteq \hat{\boldsymbol{\omega}}_{ON}^{O} \circ \hat{q}. \end{aligned} \tag{83}$$

Define a new dual vector

$$\begin{aligned} \hat{\boldsymbol{\omega}}_{ON}^{N} &= \hat{q}^{*} \circ \hat{\boldsymbol{\omega}}_{ON}^{O} \circ \hat{q} = (q + \varepsilon\tfrac{1}{2}\mathbf{t}^{O} \circ q)^{*} \\ &\quad \circ (\boldsymbol{\omega}_{ON}^{O} + \varepsilon(\dot{\mathbf{t}}^{O} + \mathbf{t}^{O} \times \boldsymbol{\omega}_{ON}^{O})) \circ (q + \varepsilon\tfrac{1}{2}\mathbf{t}^{O} \circ q) \\ &= \boldsymbol{\omega}_{ON}^{N} + \varepsilon(q^{*} \circ (\dot{\mathbf{t}}^{O} + \mathbf{t}^{O} \times \boldsymbol{\omega}_{ON}^{O}) \circ q + \boldsymbol{\omega}_{ON}^{N} \times \mathbf{t}^{N}). \end{aligned} \tag{84}$$

Using (4) and the fact that $q \circ q^{*} = 1$, it is obtained that

$$\begin{aligned} \dot{\mathbf{t}}^{N} &= d(q^{*})/dt \circ \mathbf{t}^{O} \circ q + q^{*} \circ \dot{\mathbf{t}}^{O} \circ q + q^{*} \circ \mathbf{t}^{O} \circ \dot{q} \\ &= -q^{*} \circ \dot{q} \circ q^{*} \circ \mathbf{t}^{O} \circ q + q^{*} \circ \dot{\mathbf{t}}^{O} \circ q + q^{*} \circ \mathbf{t}^{O} \circ \dot{q} \\ &= q^{*} \circ (\dot{\mathbf{t}}^{O} + \mathbf{t}^{O} \times \boldsymbol{\omega}_{ON}^{O}) \circ q. \end{aligned} \tag{85}$$

Substituting (85) into (84) gives

$$\hat{\boldsymbol{\omega}}_{ON}^{N} = \boldsymbol{\omega}_{ON}^{N} + \varepsilon(\dot{\mathbf{t}}^{N} + \boldsymbol{\omega}_{ON}^{N} \times \mathbf{t}^{N}). \tag{86}$$

Equation (83) can be rewritten as

$$2\dot{\hat{q}} = \hat{\boldsymbol{\omega}}_{ON}^{O} \circ \hat{q} = \hat{q} \circ \hat{\boldsymbol{\omega}}_{ON}^{N} \circ \hat{q}^{*} \circ \hat{q} = \hat{q} \circ \hat{\boldsymbol{\omega}}_{ON}^{N}. \tag{87}$$

D. Derivation of (55)

From the screw vector characterizing the classical screw motion in (39), the corresponding dual quaternion is

$$\begin{aligned} \hat{q} = q + \varepsilon q' &= \begin{bmatrix} \cos\left(\dfrac{\hat{\sigma}}{2}\right) \\ 0 \\ \sin\left(\dfrac{\hat{\sigma}}{2}\right)\cos(\hat{\gamma}t) \\ \sin\left(\dfrac{\hat{\sigma}}{2}\right)\sin(\hat{\gamma}t) \end{bmatrix} = \begin{bmatrix} \cos\left(\dfrac{\sigma}{2}\right) \\ 0 \\ \sin\left(\dfrac{\sigma}{2}\right)\cos(\gamma t) \\ \sin\left(\dfrac{\sigma}{2}\right)\sin(\gamma t) \end{bmatrix} \\ &+ \varepsilon \begin{bmatrix} -\dfrac{\sigma'}{2}\sin\left(\dfrac{\sigma}{2}\right) \\ 0 \\ \dfrac{\sigma'}{2}\cos\left(\dfrac{\sigma}{2}\right)\cos(\gamma t) - \gamma' t \sin\left(\dfrac{\sigma}{2}\right)\sin(\gamma t) \\ \dfrac{\sigma'}{2}\cos\left(\dfrac{\sigma}{2}\right)\sin(\gamma t) + \gamma' t \sin\left(\dfrac{\sigma}{2}\right)\cos(\gamma t) \end{bmatrix}. \end{aligned} \tag{88}$$

With (14) and (88), we have

$$\mathbf{t} = 2q^{*} \circ q' = \begin{bmatrix} -2\gamma' \sin^{2}\left(\dfrac{\sigma}{2}\right)t \\ \sigma' \cos(\gamma t) - \gamma' t \sin(\sigma)\sin(\gamma t) \\ \sigma' \sin(\gamma t) + \gamma' t \sin(\sigma)\cos(\gamma t) \end{bmatrix}. \tag{89}$$

With (89), it is straightforward to prove that (54) can be written in the following form

$$\hat{\boldsymbol{\omega}} = \boldsymbol{\omega} + \varepsilon(\dot{\mathbf{t}} + \boldsymbol{\omega} \times \mathbf{t}). \tag{90}$$

ACKNOWLEDGMENT

The authors would like to thank Robert C. Michelson and anonymous reviewers for their insightful comments and constructive suggestions. Thanks also to Dr. Hyung Keun Lee (School of Electronics, Telecommunication & Computer, Hankuk Aviation University) for reviewing an early version of this paper. We are also grateful to associate professor Zhuangzhi Yang for providing the trace generator and a doctoral candidate, Wei Shi, for his help in the numerical simulations, at the same laboratory.

REFERENCES

[1] Aspragathos, N. A., and Dimitros, J. K.
A comparative study of three methods for robot kinematics.
IEEE Transactions on Systems, Man and Cybernetics Part B: Cybernetics, **28**, 2 (Apr. 1998), 135–145.

[2] Chatfield, A. B.
Fundamentals of High Accuracy Inertial Navigation, Volume 174, Progress in Astronautics and Aeronautics, published by AIAA, Inc.

[3] Ball, R. S.
A Treatise on the Theory of Screws.
New York: Cambridge University Press, 1900.

[4] Bar-Itzhack, I. Y.
Navigation computation in terrestrial strapdown inertial navigation system.
IEEE Transactions on Aerospace and Electronic Systems, **AES-13**, 6 (Nov. 1977), 679–689.

[5] Bayro-Corrochano, E.
Motor algebra approach for visually guided robotics.
Pattern Recognition, **35** (2002), 279–294.

[6] Belinfante, J. G. F., and Kolman, B.
A Survey of Lie Groups and Lie Algebra: with Applications and Computational Methods.
Philadelphia, PA: SIAM, 1989.

[7] Bortz, J. E.
A new mathematical formulation for strapdown inertial navigation.
IEEE Transactions on Aerospace and Electronic Systems, **7**, 1 (Jan. 1971), 61–66.

[8] Bottema, O., and Roth, B.
Theoretical Kinematics.
New York: North Holland Press, 1979.

[9] Branets, V. N., and Shmyglevsky, I. P.
Application of Quaternions to the Problems of Rigid Body Orientation (In Russian).
Moscow: Nauka, ch. 3, 1973.

[10] Branets, V. N., and Shmyglevsky, I. P.
Introduction to the Theory of Strapdown Inertial Navigation System (In Russian).
Moscow: Nauka, ch. 1–6, 1992.

[11] Chasles, M.
Note sur les propriétés générales du système de deux crps semblables entr'eux et places d'une manière quelcoque dans l'espace; et sur le déplacement fini ou infiniment petit d'un corps solide libre.
Férussac, Bulletin des Sciences Mathématiques, **14** (1830), 321–326.

[12] Clifford, W.
Preliminary sketch of bi-quaternions.
Proceedings of the London Mathematical Society, **4** (1873), 381–395.

[13] Daniilidis, K.
Hand-eye calibration using dual quaternions.
International Journal of Robotics Research, **18** (1999), 286–298.

[14] Denavit, J., and Hartenberg, R. S.
A kinematic notation for the lower-pair mechanisms based on matrices.
ASME Journal of Applied Mechanics, **25**, 1 (1955), 1–29.

[15] Fischer, I. S.
Dual-Number Methods in Kinematics, Statics and Dynamics.
Boca Raton, FL: CRC Press, 1999.

[16] Funda, J., and Paul, R. P.
A computational analysis of screw transformations in robotics.
IEEE Transactions on Robotics and Automation, **6** (1990), 348–356.

[17] Funda, J., Taylor, R. H., and Paul, R. P.
On homogeneous transformations, quaternions, and computational efficiency.
IEEE Transactions on Robotics and Automation, **6** (1990), 382–388.

[18] Gusinsky, V. Z., Lesyuchevsky, V. M., Litmanovich, Yu. A., Musoff, H., and Schmidt, G. T.
New procedure for deriving optimized strapdown attitude algorithm.
Journal of Guidance, Control, and Dynamics, **20**, 4 (July–Aug. 1997), 673–680.

[19] Gusinsky, V. Z., Lesyuchevsky, V. M., Litmanovich, Yu. A., Musoff, H., and Schmidt, G. T.
Optimization of a strapdown attitude algorithm for a stochastic motion.
Navigation: Journal of The Institute of Navigation, **44**, 2 (Summer 1997), 163–170.

[20] Ignagni, M. B.
Optimal strapdown attitude integration algorithms.
Journal of Guidance, Control, and Dynamics, **13**, 2 (Mar.–Apr. 1990), 363–369.

[21] Ignagni, M. B.
Efficient class of optimized coning compensation algorithm.
Journal of Guidance, Control, and Dynamics, **19**, 2 (Mar.–Apr. 1996), 424–429.

[22] Ignagni, M. B.
Duality of optimal strapdown sculling and coning compensation algorithms.
Navigation: Journal of The Institute of Navigation, **45**, 2 (Summer 1998), 85–95.

[23] Jiang, Y. F., and Lin, Y. P.
Improved strapdown coning algorithms.
IEEE Transactions on Aerospace and Electronic Systems, **28** (Apr. 1992), 484–489.

[24] Keler, M. L.
On the theory of screws and the dual method.
In *Proceedings of A Symposium Commemorating the Legacy, Works, and Life of Sir Robert Stawell Ball Upon the 100th Anniversary of "A Treatise on the Theory of Screws"*.
University of Cambridge, Trinity College, July 9–11, 2000.

[25] Kotelnikov, A. P.
Screw calculus and some applications to geometry and mechanics.
Annuals of Imperial University of Kazan, 1895.

[26] Litmanovich, Y. A., Lesyuchevsky, V. M., and Gusinsky, V. Z.
Strapdown attitude/navigation algorithms with angular rate/specific force multiple integrals as input signals.
Presented at the Institute of Navigation 55th Annual Meeting, Cambridge, MA, June 28–30, 1999.

[27] Litmanovich, Y. A., Lesyuchevsky, V. M., and Gusinsky, V. Z.
Two new classes of strapdown navigation algorithms.
Journal of Guidance, Control, and Dynamics, **23**, 1 (Jan.–Feb. 2000), 34–44.

[28] Mark, J. G., and Tazartes, D. A.
Tuning of coning algorithm to gyro data frequency response characteristics.
Journal of Guidance, Control, and Dynamics, **24**, 4 (July–Aug. 2001), 641–647.

[29] Mark, J. G., and Tazartes, D. A.
On sculling algorithms.
In *Proceedings of the 3rd St. Petersburg International Conference on Integrated Navigation Systems, Pt. 2*, Central Scientific and Research Institute "Elektropribor," St. Petersburg, Russia, 1996, 22–26.

[30] McCarthy, J. M.
Introduction to Theoretical Kinematics.
Cambridge, MA: MIT Press, 1990, 53–66.

[31] McKern, R., and Msoff, H.
Strapdown attitude algorithms from a geometric viewpoint.
Journal of Guidance, Control, and Dynamics, **4**, 6 (Nov.–Dec. 1981), 657–661.

[32] Miller, R.
A new strapdown attitude algorithm.
Journal of Guidance, Control, and Dynamics, **6**, 4 (July–Aug. 1983), 287–291.

[33] Musoff, H., and Murphy, J. H.
Study of strapdown navigation attitude algorithm.
Journal of Guidance, Control, and Dynamics, **18**, 2 (Mar.–Apr. 1995), 287–290.

[34] Pitman, G. R.
Inertial Guidance, New York: Wiley, 1962.

[35] Roscoe, K. M.
Equivalency between strapdown inertial navigation coning and sculling integrals/algorithms.
Journal of Guidance, Control, and Dynamics, **24**, 2 (Mar.–Apr. 2001), 201–206.

[36] Savage, P. G.
Strapdown inertial navigation integration algorithm design, Part 1: attitude algorithms.
Journal of Guidance, Control, and Dynamics, **21**, 1 (Jan.–Feb. 1998), 19–28.

[37] Savage, P. G.
Strapdown inertial navigation integration algorithm design, Part 2: velocity and position algorithms.
Journal of Guidance, Control, and Dynamics, **21**, 2 (Mar.–Apr. 1998), 208–221.

[38] Soloviev, A.
Investigation into performance enhancement of integrated global positioning/inertial navigation systems by frequency domain implementation of inertial computational procedures.
Ph.D. dissertation, College of Engineering and Technology, Ohio University, 2002.

[39] Study, E.
Von den bewegungen und umlegungen.
Mathematische Annalen, **39** (1891), 441–566.

[40] Titterton, D. H., and Weston, J. L.
Strapdown Inertial Navigation Technology.
New York: Peregrinus Ltd. on behalf of the Institute of Electrical Engineers, London, 1997.

[41] Wei, M., and Schwarz, K. P.
A strapdown inertial algorithm using an Earth-fixed Cartesian frame.
Journal of The Institution of Navigation, **37**, 2 (1990), 153–167.

[42] Wu, Y. X., Wang, P., and Hu, X. P.
Algorithm of Earth-centered Earth-fixed coordinates to geodetic coordinates.
IEEE Transactions on Aerospace and Electronic Systems, **39**, 4 (2003), 1457–1461.

[43] Wu, Y. X., Hu, X. P., Hu, D. W., and Lee, H. K.
A framework of strapdown attitude algorithms using generalized kinematic vector.
Submitted for publication to *IEEE Transactions on Aerospace and Electronic Systems.*

[44] Yang, A. T.
Application of dual-number quaternion algebra and dual numbers to the analysis of spatial mechanisms.
Ph.D. dissertation, Dept. Mechanical Engineering, Columbia University, 1964.

Yuanxin Wu was born in Jinan, P. R. China, in 1976. He received the B.Sc. degree in navigation, guidance and control from the Department of Automatic Control, National University of Defense Technology, Changsha, P. R. China in 1998. He is now a Ph.D. candidate at National University of Defense Technology.

He was awarded the Guanghua Scholarship in 1998 and 2003, and received the University Ph.D. Innovation Fellowship in 2001. His current research interests include navigation system, general estimation theory, computer vision, information fusion, application Clifford algebra in engineering.

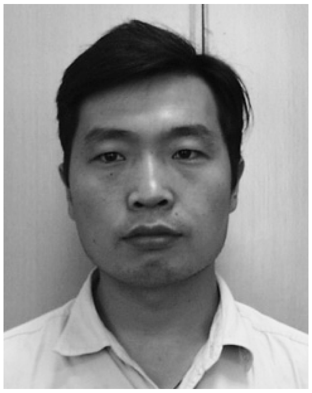
Xiaoping Hu was born in Luzhou, P. R. China, in 1960. He received the B.Sc. and M.Sc. degrees in automatic control systems and aerocraft designing from the Department of Automatic Control, National University of Defense Technology, Changsha, P. R. China, in 1982 and 1985, respectively.

Currently he is with National University of Defence Technology as a professor and the dean of College of Mechatronics and Automation. His scientific interests include inertial and satellite navigation, aerocraft guidance, and control.

Dewen Hu (M'03) was born in Hunan, P. R. China, in 1963. He received the B.Sc. and M.Sc. degrees in electrical engineering from Xi'an Jiaotong University, P. R. China, in 1983 and 1986, respectively, and Ph.D. degree in pattern analysis and intelligent systems from National University of Defense Technology in 1999.

From 1986 to 1995, he was with National University of Defense Technology. From Oct. 1995 to Oct. 1996, he was with the University of Sheffield, UK as a visiting scholar. He is now with National University of Defense Technology as a professor of the Department of Automatic Control. His research interests include image processing, system identification and control, neural networks, cognitive science.

Dr. Hu has over 80 papers and three monographs in the areas of his interests. He is the joint recipient of more than a dozen of academic prizes for his researches on neuro-control, intelligent robot and image processing.



Tao Li received his B.Sc, M.Sc., and Ph.D. degrees from the Department of Automatic Control, National University of Defense Technology, Changsha, P. R. China, in 1996, 1999, and 2003, respectively.

He is now a lecturer with National University of Defense Technology. His research interests include inertial navigation, integrated navigation and filtering technique.

Junxiang Lian was born in Hubei, P. R. China, in 1978. He received a B.S. and an M.S. in automatic control from National University of Defense Technology, Changsha, P. R. China, in 2000 and 2002, respectively. He is currently a Ph.D. student at National University of Defense Technology.

He research interest is underwater integrated navigation including GPS and inertial navigation.