1. INTRODUCTION

Attitude computation by integrating angular velocity is an important way to acquire the attitude, rotation, or orientation [1], [2]. Attitude information is essential to many areas like unmanned vehicle navigation and control, virtual/augmented reality, satellite communication, robotics, and computer vision [3]. The modern-day attitude algorithm structure in the inertial navigation field, established in 1970s [4], [5], has relied on a simplified rotation vector differential equation for incremental attitude update [6], [7]. In parallel, a number of fields prefer the quaternion to deal with attitude computation as it is singularity free and has a linear differential equation, e.g., robotics [8], [9], space applications [10], and computational mathematics [11]–[13], where the structure-preserving attributes of geometric integration is preferred. It might be argued that the common numerical methods such as the Runge-Kutta can also be used to implement the attitude integration but they are not norm-preserving and thus inevitably lead to accuracy loss in attitude computation [14]. However, the above-mentioned methods have no way to drive the non-commutativity error to zero with practically finite sampling intervals. It has long been believed by navigation practitioners that the modern-day attitude algorithm is already good enough for applications [6]–[15]. However, dynamic situations like the actual military projectiles with complex high-speed rotations and the cold-atom interference gyroscopes of ultra-high precision on the horizon demand for more accurate attitude algorithms.

Recently, in the navigation community, independent works on high-accurate attitude algorithms have appeared or been under way [16]–[18]. They share the same spirit of trying to accurately solve the attitude kinematic equation based on the fitted angular velocity polynomial function. The main difference among these works is the chosen attitude parameterization. The quaternion is employed in [18], which first demonstrated in the public literature the practical potential of the Picard-type successive approximation method. The direction cosine matrix (DCM) could be used instead by the Taylor series expansion [19]. These methods can be traced back to the Russian seminal work in 1990s [20], where they were, respectively, classified as Type-I and Type-II methods. In specific, the Type-II method used the rotation vector instead of DCM. The three-component attitude parameterization is minimal and does not need to satisfy the inherent constraints of the redundant-component parameterizations, such as the quaternion with the unit-norm constraint and the DCM with the orthogonal and ±1-determinant constraints. In view of the finite-polynomial-like differential equation of the three-component Rodrigues vector, Wu [16] proposes the RodFIter method to reconstruct the attitude, which makes a natural use of the iterative function integration of the Rodrigues vector’s kinematic equation. It highlights the capability of analytical attitude reconstruction over the whole update interval and provably converges to the true attitude if only the angular velocity is exact. Unfortunately, all high-accurate attitude algorithms
face the problem of high computational burden, especially for real-time applications. Hence, to improve the computational efficiency, Wu [17] further comes up with a substantially fast version of RodFIter at little expense of accuracy by exploiting the excellent property of Chebyshev polynomial. It reformulates the original RodFIter in terms of the iterative computation of the Rodrigues vector’s Chebyshev polynomial coefficients and exerts the Chebyshev polynomial truncation. In principle, the idea of RodFIter could be extended to various attitude parameters including the quaternion, but a question worthy of being investigated is whether the unit-norm constraint of quaternion affects the accuracy to be achieved. It is right the concern that has prevented one from using quaternion for attitude computation before Yan’s work [18].

The contribution of this paper is that the technique of iterative functional integration and Chebyshev polynomial truncation is applied to the quaternion kinematic equation (named as the QuatFIter, hereafter) for attitude reconstruction. It is shown that the QuatFIter has about two times better computational efficiency than the RodFIter, thanks to the linear quaternion kinematic equation. The remainder of the paper is organized as follows: Section II is devoted to developing the QuatFIter in the framework of iterative functional integration and analyzing its convergence property. The QuatFIter’s relationship to the Picard-type quaternion method is discussed. Section III improves the computational efficiency of the QuatFIter by appropriate Chebyshev polynomial truncation, and then analyzes the error characteristics accordingly. Section IV evaluates the QuatFIter in terms of convergence speed, accuracy, and computation efficiency as compared with the RodFIter as well as two common numerical integration methods by numerical simulations. A brief summary is given in the last section of the paper.

II. QUATFITER FOR ATTITUDE RECONSTRUCTION

Quaternion is an extension of complex number and constitutes a noncommutative group [21]. Attitude quaternion \( q \) can be represented as a four-dimensional (4-D) column vector of unit magnitude, i.e., \( q = [s \quad \eta^T]^T \), where \( s \) is the scalar part and \( \eta \) is the vector part. If these two parts are regarded as a scalar quaternion and a vector quaternion, respectively, then quaternion can be alternatively written as \( q = s + \eta \).

The attitude quaternion kinematic equation is related to the angular velocity as
\[
\dot{q} = \frac{1}{2} q \circ \omega
\]
(1)
where \( \omega \) is the angular velocity vector quaternion with zero scalar part formed by the 3-D angular velocity vector. With some abuse of symbols, a vector quaternion is taken equally as a 3-D column vector throughout the paper. The operator \( \circ \) means the multiplication of quaternions defined as
\[
q_1 \circ q_2 = \begin{bmatrix} s_1 \\ \eta_2 \end{bmatrix} \begin{bmatrix} s_2 \\ -\eta_1 \end{bmatrix} = \begin{bmatrix} s_1 s_2 - \eta_1 \eta_2 \\ s_1 \eta_2 + s_2 \eta_1 + \eta_1 \times \eta_2 \end{bmatrix}.
\]
(2)

The two quaternion multiplication matrices, \( [q] \) and \( [\bar{q}] \), are, respectively, defined by [3]
\[
[q] \triangleq \begin{bmatrix} s \\ -\eta^T \\ \eta s I_3 + \eta \times \end{bmatrix}, \quad [\bar{q}] \triangleq \begin{bmatrix} s \\ -\eta^T \\ \eta s I_3 - \eta \times \end{bmatrix}
\]
(3)
where \( \eta \times \) is the antisymmetric matrix formed by the vector part \( \eta \) and \( I_3 \) is the identical matrix of dimension three. Note the quaternion kinematic equation could be written as a form of linear system in quaternion as follows:
\[
\dot{q} = [\omega/2] q \overset{\Delta}{=} Wq.
\]
(4)

For the situation that the direction of the angular velocity remains fixed over a time interval \( [0 \ t] \), the attitude quaternion solution can be solved as \( q(t) = \exp(\int_0^t W \ dt)q(0) \) [2]. However, there is no explicit solution for a general rotation of interest in this paper.

A. QuatFIter and Its Convergence Property

Without the loss of generality, consider the incremental/relative attitude over the time interval \( [0 \ t] \) with the initial quaternion given by \( q(0) = [1 \ 0 \ 0 \ 0]^T \overset{\Delta}{=} 1 \). Integrating (1) yields
\[
q = q(0) + \frac{1}{2} \int_0^t q \circ \omega \ d\tau = 1 + \frac{1}{2} \int_0^t q \circ \omega \ d\tau.
\]
(5)

Applying the functional iteration integration technique [16], the quaternion can be iteratively computed as
\[
q_{l+1} = 1 + \frac{1}{2} \int_0^t q_l \circ \omega \ d\tau, \quad l \geq 0
\]
(6)
with some chosen initial attitude quaternion function, say an identity quaternion, namely, \( q_0(t) \equiv [1 \ 0 \ 0 \ 0]^T \). It can be readily checked that the above iteration process will produce such a sequence of quaternion functions as
\[
q_0 = 1 \\
q_1 = 1 + \frac{1}{2} \int_0^t q_0 \circ \omega \ d\tau = 1 + \frac{1}{2} \int_0^t \omega \ d\tau \\
q_2 = 1 + \frac{1}{2} \int_0^t q_1 \circ \omega \ d\tau = 1 + \frac{1}{2} \int_0^t \left( \left( 1 + \frac{1}{2} \int_0^t \omega \ d\tau_2 \right) \circ \omega \ d\tau_1 \right)
\]
\[
q_l = 1 + \frac{1}{2} \int_0^t q_{l-1} \circ \omega \ d\tau + \cdots + \frac{1}{2^l} \int_0^t \left( \cdots \left( \int_0^t \omega \ d\tau_{l-2} \right) \circ \omega \ d\tau_{l-1} \right) \circ \omega \ d\tau_1.
\]
(7)

For a general linear system of the form (4), there exists a general convergence result stating that the iterative
Given the true angular velocity function \( \omega \) over the interval \([0, t]\), the iterative process as given in (7) converges to the true attitude quaternion function solution to (1) for bounded \( t \sup |\omega| \).

**Proof** For the convenience of convergence analysis, the sequence of quaternion functions is written as

\[
q_j = q_0 + \sum_{j=0}^{l-1} (q_{j+1} - q_j), \quad l \geq 1.
\]

Denote \( \alpha = \Delta \sup_{\tau \in [0, t]} |\omega(\tau)| \), where \( |\cdot| \) denotes the vector norm. Then,

\[
|q_0| = 1,
\]

\[
|q_1 - q_0| = \frac{1}{2} \int_0^t \omega d\tau = \frac{1}{2} \int_0^t W[1 0 0 0]^T d\tau \leq \frac{1}{2} \|W\| d\tau \leq \alpha t,
\]

\[
|q_2 - q_1| = \frac{1}{2} \int_0^t (q_1 - q_0) \omega d\tau = \frac{1}{2} \int_0^t W (q_1 - q_0) d\tau \leq \frac{1}{2} \|W\| \int_0^t |q_1 - q_0| d\tau \leq \frac{1}{2} \|W\| \int_0^t \alpha^2 d\tau = \frac{1}{2} \alpha^2 t^2.
\]

\[
|q_{j+1} - q_j| = \frac{1}{2} \int_0^t (q_j - q_{j-1}) \omega d\tau \leq \frac{1}{(j+1)!} \alpha^j t^{j+1}
\]

where \( \|\cdot\| \) denotes the matrix Frobenius norm, namely the root of the sum of all matrix elements squared. Note that the sum of the bounds on the right side \( 1 + \alpha t + \frac{1}{2} \alpha^2 t^2 + \cdots + \frac{1}{(j+1)!} (\alpha t)^{j+1} + \cdots = e^{\alpha t} \). According to the Weierstrass M-test (see Appendix), the iterative process (7) converges uniformly and absolutely on the interval \([0, t]\). Finally, it can be readily checked that the limit of the quaternion function sequence satisfies (1), i.e., \( \frac{d q_\infty}{d\tau} = \frac{1}{2} q_\infty \omega \). It means that \( q_\infty \) is the solution to the quaternion kinematic (1), that is to say, the iterative process (7) converges to the true attitude quaternion function.

For those fixed-axis rotation cases that the \( W \)-commutativity condition holds, the limit of the quaternion function sequence is reduced to \( q_\infty = \exp\{\int_0^t W d\tau\} q(0) \) (see Proposition 1 in Appendix).

It may be argued that the region \( t \sup |\omega| < 2 \) is already sufficient for practical applications using the RodFIter for incremental attitude update, because the Rodrigues vector is by definition singular at rotation angles \( \pm \pi \).

**B. Relation to Picard-Type Successive Approximation Method**

The Picard-type successive approximation method [18, 20] repeatedly uses (5) to derive the approximation of the attitude quaternion. In specific, it substitutes (5) repeatedly into the right side of the resultant as follows:

\[
q = 1 + \frac{1}{2} \int_0^t q \omega d\tau + \frac{1}{2} \int_0^t \frac{1}{2} \int_0^t q \omega d\tau_2 \omega d\tau_1 = 1 + \frac{1}{2} \int_0^t \frac{1}{2} \int_0^t \frac{1}{2} \int_0^t q \omega d\tau_3 \omega d\tau_2 \omega d\tau_1.
\]

A second substitution of (5) into the above gives

\[
q = 1 + \frac{1}{2} \int_0^t q \omega d\tau + \frac{1}{2} \int_0^t \left( \frac{1}{4} \int_0^t q \omega d\tau_{4} \right) \omega d\tau_2 \omega d\tau_1 = 1 + \frac{1}{2} \int_0^t \frac{1}{4} \int_0^t \frac{1}{4} \int_0^t q \omega d\tau_3 \omega d\tau_2 \omega d\tau_1.
\]

Of course, we might substitute (10) instead of (5) into (10) and it yields

\[
q = 1 + \frac{1}{2} \int_0^t q \omega d\tau + \frac{1}{2} \int_0^t \left( \frac{1}{8} \int_0^t \frac{1}{8} \int_0^t q \omega d\tau_4 \right) \omega d\tau_2 \omega d\tau_1
\]

Anyway, the Picard-type method [18–20] takes the sum of all terms but the last one as an approximation of the attitude quaternion, which is identical to the QuatFIter sequence (7). Therefore, the last residual term containing the unknown attitude quaternion explicitly quantifies the approximation error of the QuatFIter sequence (7).

Denote the additive error of the quaternion function sequence (7) from the true quaternion function as \( \delta q_i \).
\[ \mathbf{q}_t - \mathbf{q} \] Using (12), it can be derived that

\[ |\delta \mathbf{q}| = \left| \int_0^t \left( \int_0^{\tau_1} \cdots \left( \int_0^{\tau_{n-1}} \mathbf{q} \circ \omega_2 d\tau_2 \right) \cdots \left( \int_0^{\tau_1} \mathbf{q} \circ \omega_2 d\tau_1 \right) \right| \]

\[ = \int_0^t \int_0^{\tau_1} \cdots \left( \omega(\tau_1) \mathbf{W}(\tau_2) \cdots \mathbf{W}(\tau_l) \mathbf{q} \right) d\tau_l \cdots d\tau_2 d\tau_1 \]

\[ \leq \int_0^t \int_0^{\tau_1} \cdots \left( \omega(\tau_1) \mathbf{W}(\tau_2) \cdots \mathbf{W}(\tau_l) \right) \mathbf{q} d\tau_l \cdots d\tau_2 d\tau_1 \]

\[ \leq \int_0^t \int_0^{\tau_1} \cdots \int_0^{\tau_l-1} \omega(\tau_1) \mathbf{W}(\tau_2) \cdots \mathbf{W}(\tau_l) \mathbf{q} d\tau_l \cdots d\tau_2 d\tau_1 \]

which approaches zero as \( l \to \infty \) for any bounded \( t \sup |\omega| \). It can be regarded as an alternative proof of Theorem 1.

III. FAST QUATFILTER BY CHEBYSHEV POLYNOMIAL APPROXIMATION AND TRUNCATION

A. Angular Velocity Fitting by Chebyshev Polynomial

The Chebyshev polynomial of the first kind is defined over the interval \([-1,1]\) by the recurrence relation as

\[ F_0(x) = 1, \quad F_1(x) = x, \quad F_{i+1}(x) = 2x F_i(x) - F_{i-1}(x) \]

where \( F_i(x) \) is the \( i \)-th degree Chebyshev polynomial of the first kind. For any \( j, k \geq 0 \), the Chebyshev polynomial of first kind satisfies the equality [23] as follows:

\[ F_j(\tau) F_k(\tau) = \frac{1}{2} (F_{j+k}(\tau) + F_{|j-k|}(\tau)) \]  

Then the integrated \( i \)-th degree Chebyshev polynomial can be expressed as a linear combination of Chebyshev polynomials, given by

\[ G_{i,[-1,1]}(\tau) = \int_{-1}^{\tau} F_i(\tau) d\tau \]

\[ = \left( \frac{F_i(\tau)}{2\tau + 1} - \frac{F_i(\tau)}{2\tau - 1} \right) - \frac{(-1)^i}{2\tau} F_0(\tau) \quad \text{for} \quad i \neq 1 \]

\[ = \frac{F_i(\tau)}{4} - \frac{F_0(\tau)}{4} \quad \text{for} \quad i = 1 \]

where \( F_i(-1) = (-1)^i \).

Assume discrete angular velocity or angular increment measurements at time instants \( t_k \) \((k = 1, 2, \ldots, N)\) by a triad of gyroscopes. In order to apply the Chebyshev polynomials, the actual time interval is mapped onto \([-1,1]\) by letting \( \tau = \frac{t_k}{T} \). The RodFilter [16, 17] employs the Chebyshev polynomials to fit the angular velocity polynomial as

\[ \hat{\omega} = \sum_{i=0}^{n} \mathbf{c}_i F_i(\tau), \quad n \leq N - 1 \]

where the coefficient \( \mathbf{c}_i \) is determined by the least-square method using the discrete angular velocity or angular increment measurements. With the converging condition of Theorem 1 in mind, substituting (17) into the iteration process (6) is supposed to well reconstruct the attitude quaternion function as a finite polynomial, as the group of polynomials are closed under elementary arithmetic operations.

B. Fast QuatFilter by Iterative Computation of Chebyshev Polynomial Coefficients

Next, we will reformulate the QuatFilter as the iterative computation of Chebyshev polynomial coefficients and perform the analysis of computational complexity. Fast QuatFilter implementation will be achieved by appropriate truncation of Chebyshev polynomials.

If the angular velocity is smooth, the absolute value of Chebyshev coefficients \( \mathbf{c}_i \) in (17) will decrease exponentially [23]. Assume the quaternion estimate at the \( l \)-th iteration is given by a weighted sum of Chebyshev polynomials, say

\[ \mathbf{q}_{l+1} = \frac{\Delta}{m_l} \sum_{i=0}^{m_l} \mathbf{b}_{l,i} F_i(\tau) \]  

where \( m_l \) is the maximum degree and \( \mathbf{b}_{l,i} \) is the coefficient of \( i \)-th degree Chebyshev polynomial at the \( l \)-th iteration. The integral in (6) is transformed to that over the mapped interval of Chebyshev polynomials, that is,

\[ \mathbf{q}_{l+1} = 1 + \frac{1}{2} \int_0^t \mathbf{q}_o \circ \omega d\tau = 1 + \frac{1}{4} \int_0^t \mathbf{q}_o \circ \omega d\tau. \]

Substituting (17)–(19), the integral on the right side has the form

\[ \int_{-1}^{\tau} \mathbf{q}_o \circ \omega d\tau \]

\[ = \int_{-1}^{\tau} \left[ \sum_{j=0}^{m_l} \mathbf{b}_{l,j} F_j(\tau) \right] \circ \sum_{j=0}^{n} \mathbf{c}_j F_j(\tau) d\tau \]

\[ = \sum_{j=0}^{n} \sum_{j=0}^{m_l} \mathbf{b}_{l,i} \circ \mathbf{c}_j \int_{-1}^{\tau} F_i(\tau) F_j(\tau) d\tau \]

\[ = \frac{1}{2} \sum_{j=0}^{n} \sum_{j=0}^{m_l} \mathbf{b}_{l,i} \circ \mathbf{c}_j \left( F_{i+j}(\tau) + F_{|i-j|}(\tau) \right) d\tau \]

\[ = \frac{1}{2} \sum_{j=0}^{n} \sum_{j=0}^{m_l} \mathbf{b}_{l,i} \circ \mathbf{c}_j \left( G_{i+j,[-1,1]}(\tau) + G_{|i-j|,[-1,1]}(\tau) \right). \]

So (19) is expressed as

\[ \mathbf{q}_{l+1} = 1 + \frac{T_N}{8} \sum_{j=0}^{n} \sum_{j=0}^{m_l} \mathbf{b}_{l,i} \circ \mathbf{c}_j \left( G_{i+j,[-1,1]}(\tau) + G_{|i-j|,[-1,1]}(\tau) \right) \]

\[ \Delta = \sum_{i=0}^{m_l} \mathbf{b}_{l+1,i} F_i(\tau) \]  

where \( m_{l+1} = m_l + n + 1 \) with \( m_0 = 0 \). It can be readily verified that \( m_l = (n + 1)! \). As a quaternion multiplication (2) involves 16 scalar multiplications, the computational complexity of (21) in terms of scalar multiplications is proportion to \( O(16nm_l) = O(16n(n + 1)!). \) For instance, for \( n = 8 \) at the seventh iteration, \( m_7 = 63 \) and the QuatFilter’s computational complexity is proportional to \( O(8000). \)
The computational burden can be reduced by means of appropriate polynomial truncation. Regarding (17), the angular velocity is fitted by a Chebyshev polynomial of degree \( n \) in time and thus the first integral of (7), namely, the integrated angular velocity, is accurate up to degree \( n + 1 \). It is reasonable to abandon those polynomials of higher order than some prescribed threshold, say \( n_T \), to save computation

\[
\hat{\mathbf{q}}_{l+1} = 1 + \frac{f_N}{8} \sum_{i=0}^{n_T} \sum_{j=0}^{n} \mathbf{b}_{l,i} \circ \mathbf{c}_j \left( G_{i+j,-[l+j] \tau} + G_{i+j,[l+j] \tau} \right)
\]

\[
\approx \sum_{i=0}^{n_T} \mathbf{b}_{l+1,i} F_i (\tau)
\]

(22)

According to the coefficient-decreasing and \( \pm 1 \)-bounded properties of Chebyshev polynomials [23], the truncation error is bounded by the coefficient of the first neglected Chebyshev polynomial, namely, \( |\delta \mathbf{q}_l^t| \leq |\mathbf{b}_{l,n_T+1}| \). The superscript “\( t \)” denotes that the error is owed to the polynomial truncation.

The iteration (21) becomes

\[
\hat{\mathbf{q}}_{l+1} = 1 + \frac{f_N}{8} \sum_{i=0}^{n_T} \sum_{j=0}^{n} \mathbf{b}_{l,i} \circ \mathbf{c}_j \left( G_{i+j,-[l+j] \tau} + G_{i+j,[l+j] \tau} \right)
\]

\[
\approx \sum_{i=0}^{n_T} \mathbf{b}_{l+1,i} F_i (\tau)
\]

(23)

where the last approximation is due to the truncation at each iteration. By way of polynomial truncation, the computational complexity is further reduced to \( O(16nn_T) \). For instance, it will be proportional to \( O(1200) \) for \( n_T = n + 1 \) and \( n = 8 \) at each iteration. In contrast, the fast RodFIter’s computational complexity is \( O(6nn_T + 6nn_T^2) = O(4300) \) at each iteration [17]. The significant reduction of 3~4 times is owed to the quaternion’s linear rate relation to the quaternion itself and the angular velocity as shown in (1), while the Rodrigues vector rate equation is quadratic with respect to the Rodrigues vector itself [16], [17].

Table I lists the main steps of the QuatFIter and compares with the RodFIter for easy reference. It should be noted that the quaternion normalization is performed outside of the essential iteration (Steps 3–4), otherwise the quaternion estimate \( \hat{\mathbf{q}} \) is not a finite polynomial any more. The RotFIter in [16] based on the approximate rotation vector kinematics also fits into the framework of iterative function integration but is not listed above, because it compares unfavorably with RodFIter therein.

**Theorem 2** Given the true angular velocity function \( \mathbf{\omega} \) over the interval \([0 \tau]\), the fast QuatFIter (23) converges to the true attitude quaternion function up to the truncated polynomial degree \( n_T \).

**Proof** Interested readers are referred to Theorem 2 in [17] for some details.
C. Error Analysis

It is obvious that the fast QuatFIter has three error sources: the angular velocity’s error $\delta \omega$, the initial iteration error $\delta \hat{q}_0$, and the truncation error at the $l$th iteration $\delta q'_l$. The additive quaternion error (to first order) of the fast QuatFIter propagates as

$$\left| \delta \hat{q}_1 \right| = \left\| \delta \hat{q}_0 \right\| \omega dt + \frac{1}{2} \int_0^t \left\| \delta \omega dt \right\| + \left| \delta q'_1 \right|$$

$$\leq \int_0^t \left\| \delta \omega dt \right\| + \frac{1}{2} \int_0^t \left\| \delta \omega dt \right\| + \left| \delta q'_1 \right|$$

$$\leq t \sup |\omega| \sup |\delta \hat{q}_0| + t \sup |\delta \omega| \sup |\hat{q}_0| + \left| \delta q'_1 \right|$$

and

$$\left| \delta \hat{q}_2 \right| \leq \int_0^t \left\| \delta \omega dt \right\| + \frac{1}{2} \int_0^t \left\| \delta \omega dt \right\| + \left| \delta q'_2 \right|$$

$$\leq \int_0^t \sup |\omega| \left( t \sup |\omega| \sup |\delta \hat{q}_0| + t \sup |\delta \omega| \sup |\hat{q}_0| + \left| \delta q'_1 \right| \right)$$

$$= \frac{t^2}{2} \left( \sup |\omega| \right)^2 \sup |\hat{q}_0| + \frac{t^2}{2} \sup |\omega| \sup |\delta \hat{q}_0| + \left| \delta q'_1 \right|$$

$$\leq \frac{t^2}{2} \left( \sup |\omega| \right)^2 \sup |\hat{q}_0| + t \sup |\omega| \left( t \sup |\omega| \right) \sum_{k=0}^{l-1} \frac{(t \sup |\omega|)^k}{(k+1)!}$$

$$\times \sup |\hat{q}_{l-1-k}| + \frac{t \sup |\omega|}{k!} \sum_{k=0}^{l-1} \left( t \sup |\omega| \right)^k \sup |\delta q'_{l-k}|.$$

(25)

Among the above error components, the first term is owed to the initial quaternion error and quickly vanishes for large iterations for any bounded $t \sup |\omega|$. It means the QuatFIter converges regardless of the initial quaternion function. The third term is owed to the polynomial truncation at each iteration, in which the weights of the early iterations are much smaller than those of later iterations, and thus can be approximated by the last truncation error, i.e.,

$$\sum_{k=0}^{l-1} \frac{(t \sup |\omega|)^k}{k!} \sup |\delta q'_{l-k}|$$

$$\approx \sum_{k=0}^{l-1} \frac{(t \sup |\omega|)^k}{k!} |\hat{q}_{l-k,n+1}| \approx \frac{|\hat{q}_{l,n+1}|}{n!}.$$

(26)

The second term depends on the angular velocity error and the norm of the quaternion estimate, and it can be approximated by

$$t \sup |\delta \omega| \sum_{k=0}^{l-1} \frac{(t \sup |\omega|)^k}{(k+1)!}$$

$$\approx t \sup |\delta \omega| \sum_{k=0}^{l-1} \frac{(t \sup |\omega|)^k}{(k+1)!} \approx t \sup |\delta \omega|$$

(27)

as $\sup |\hat{q}| \approx 1$. Therefore, for a large number of iterations, the quaternion error of the fast QuatFIter in (23) is approximately bounded by

$$\sup |\delta \hat{q}_l| \approx t \sup |\delta \omega| + \left| b_{l,n+1} \right|.$$  (28)

This indicates that the fast QuatFIter’s error is generally dominated by the angular velocity error and the last truncation error as well. An ideal case is when $\delta \omega = 0$, namely the angular velocity fitting error is zero, for which higher order of truncation means higher accuracy.

IV. SIMULATION RESULTS

In this section, simulations are performed in the context of coning motion and constant rotation scenarios to evaluate the QuatFIter as compared with the RodFIter [16], [17].

The coning motion has explicit analytical expressions in angular velocity and the associated Rodrigues vector and attitude quaternion, so it is widely employed as a standard criterion for algorithm accuracy assessment in the inertial navigation field [1]. The angular velocity of the coning motion is given by $\omega = \Omega \left[ -2 \sin^2(\alpha/2) - \sin(\alpha) \sin(\bar{\Omega}t) \sin(\alpha) \cos(\bar{\Omega}t) \right]^T$. The corresponding true attitude quaternion is $q = \frac{2g}{\sqrt{\bar{g}^2 - 4}}$, where the Rodrigues vector $g = 2\tan(\alpha/2)$ $[\cos(\bar{\Omega}t) \sin(\bar{\Omega}t)]^T$. The coning angle is set to $\alpha = 10^\circ$, the coning frequency $\Omega = 0.74\pi$ rad/s. The angular increment measurement from gyroscopes is assumed and the discrete sampling rate is nominally set to 100 Hz.

The following principal angle metric is used to quantify the attitude computation error:

$$\varepsilon_{att} = 2 \left| \left[ q^* \circ \hat{q} \right]_{2:4} \right|$$

(29)

where $\hat{q}$ denotes the quaternion estimate, and the superscript * means the quaternion conjugate and the operator $[\cdot]_{2:4}$ takes the vector part of the error quaternion. Hereafter, the order of the fitted angular velocity is uniformly set to $n = N - 1$.

Fig. 1 plots attitude errors of QuatFIter and RodFIter at each iteration in the first update interval for $N = 5$ (subfigures in the upper row) and $N = 8$ (subfigures in the lower row). Truncation order $n_T = n + 2$ is used for both, in contrast to $n_T = n + 1$ in [17], for the sake of marginally better accuracy. The quaternion estimate of QuatFIter is normalized right before using (29) to compute the principal angle error, as explicitly given by Step 5 in Table I. Lines with same colors indicate results of the same iteration. As for RodFIter denoted by decoration lines, the attitude errors reduce uniformly along with the iterations, while the
QuatFIter’s errors have a nonuniform error-reducing pace. For the $l$th iteration, the QuatFIter’s errors stay consistently above those of RodFIter; yet, the QuatFIter’s errors at the $(\ell + 1)$-th iteration stay consistently below those of RodFIter at the $l$th iteration. It means that RodFIter has a faster convergence rate with respect to the iterations, although they have almost identical final errors up to machine precision. The subfigures in the right column present the quaternion norm error from unity. It shows that the norm of the quaternion estimate approaches unity as the iteration goes. No clear relationship between the attitude error and the norm error can be identified. For instance, the quaternion norm errors at the second and third iterations are relatively close, but the attitude error at the third iteration is reduced by about four orders from that at the second iteration.

Fig. 2 presents the attitude computation error of the QuatFIter for $10$ s, compared with the RodFIter ($N = 8$ at the fourth, fifth, and seventh iterations) and the mainstream two-sample algorithm in the practical inertial navigation system. It confirms the above observation that the QuatFIter has a slower convergence rate relative to the number of iteration but has final accuracy comparable with the RodFIter.

Table II compares the time cost ($N = 8$ at seven iterations) averaged across 50 runs on the Matlab platform. It shows that the RodFIter’s computational cost is about 2.5 times of that the QuatFIter.
For the special case of constant angular velocity, we set $\omega = [1 3 2]^T$ rad/s, for which the noncommutativity vanishes and the mainstream two-sample algorithm is exact. This case belongs to the situation of $\delta \omega = 0$ discussed in the text below (28), where higher truncation order leads to better accuracy, so the truncation order is specially increased to $n_T = n + 10$. Fig. 3 plots the attitude error of the QuatFIter, as compared with the RodFIter ($N = 2$ and 8) and the mainstream two-sample algorithm (exact in this special case). Similar phenomenon is observed that the QuatFIter’s errors stay consistently above those of RodFIter for the same number of iterations and it takes the QuatFIter four more iterations than the RodFIter to achieve the accuracy level up to machine precision, namely, 9 and 11 iterations versus 5 and 7 iterations for $N = 2$ and $N = 8$, respectively. That is to say, the RodFIter has a faster convergence rate and a rather uniform error reduction with respect to the number of iterations. However, the QuatFIter’s disadvantage of slower convergence is not a big concern and can be well remedied by spending several more iterations thanks to its faster speed of implementation. Table II also presents the time cost for both to achieve comparable accuracy and the QuatFIter’s computational cost further reduces to below one-third of the RodFIter’s, mainly due to the computational complexity of the latter proportional to the truncation order squared.

Finally, we compare the QuatFIter with the mainstream two-sample attitude algorithm, as well as two common numerical integration methods addressed in [10], namely, the classical fourth-order Runge-Kutta algorithm (RK4n) and the fourth-order Crouch–Grossman Lie group method (CG4n). The update intervals of RK4n and CG4n are set to be consistent with the mainstream two-sample algorithm. Figs. 4 plots the comparison results for two different set-

| TABLE II Comparison of Computation Time (10-s data, $N = 8$) |
|-----------------|-----------------|-----------------|
|                | QuatFIter       | RodFIter        | Mainstream (N=2) |
| Coning motion  | 1.04s ($l = 7$) | 2.65s ($l = 7$) | 0.07s            |
| Constant angular velocity ($n_T = n + 10$) | 1.47s ($l = 11$) | 4.83s ($l = 7$) |                 |
tings of coning motions: $\alpha = 10^\circ$, $\Omega = 0.74\pi$ rad/s and $\alpha = 90^\circ$, $\Omega = 1.74\pi$. It shows the RK4n and the CG4n are inferior in accuracy to the mainstream two-sample algorithm, especially in the case of significant coning motions (Fig. 4, right subfigure). This observation accords with the common belief in the navigation community. The QuatFIter performs still the best with a remarkable advantage of at least 6–7 orders in accuracy.

V. DISCUSSIONS AND CONCLUSION

There has been a long-time consistent pursuit of accurate attitude computation by integrating angular velocity in many engineering fields. A significant advance has recently appeared in the framework of functional iterative integration of attitude kinematics, for example, the RodFIter based on the Rodrigues vector that is capable of analytically reconstructing the attitude over the whole update time interval. This paper further exploits its benefit and comes up with the QuatFIter for attitude reconstruction using the quaternion in view of its linear kinematic equation. It is shown that the QuatFIter is equivalent to the previous Picard-type successive approximate quaternion method. Chebyshev polynomial approximation and truncation are applied to reduce the computational cost of QuatFIter. Numerical results show that the QuatFIter has about two times better computational efficiency at comparable accuracy to the RodFIter, although the latter has the advantage of relatively uniform and faster error reduction with respect to the number of iterations. It is suspected that the unity-norm constraint of the quaternion contributes to QuatFIter’s nonuniform error reduction, but no apparent relationship between the attitude error and the quaternion norm error has been identified so far.

APPENDIX

(Weierstrass M-Test [22]): Suppose $\{f_j\}_{j=0}^{\infty}$ is a sequence of real functions and the sequence of positive real numbers $\sum_{j=0}^{\infty} \alpha_j$ converges. If $\|f_j(t)\| \leq \alpha_j$ ($j \geq 0$) on a time interval of interest, then the series of real functions $\sum_{j=0}^{\infty} f_j(t)$ converge uniformly and absolutely on the time interval.

PROPOSITION [24], [25] If the commutativity $W(t_1) W(t_2) = W(t_2) W(t_1)$ holds for any $t_1, t_2 \in [0, t]$, then $q_\infty = \exp(\int_0^t W d\tau) q(0)$ is the quaternion solution to (4).

PROOF Denote $I_l(t) = \int_0^t W(\tau) \int_0^\tau W(\tau_2) \cdots \int_0^{\tau_{l-1}} W(\tau_l) d\tau_l \cdots d\tau_2 d\tau_1$. It will be proved that $I_l(t) = \frac{1}{l!}(\int_0^t W(\tau) d\tau)^l$. This is obviously true for $l = 0, 1$. Assume it is true for $l$, then we have

$$I_{l+1}(t) = \int_0^t W(\tau) I_l(\tau) d\tau$$

$$= \int_0^t W(\tau) \frac{1}{l!} \left( \int_0^\tau W(\tau) d\tau \right)^l d\tau. \quad (30)$$

The integrand on the right side can be rewritten as

$$W(\tau) \frac{1}{l!} \left( \int_0^\tau W(\tau) d\tau \right)^l$$

$$= d \left[ \frac{1}{(l+1)!} \left( \int_0^\tau W(\tau) d\tau \right)^{l+1} \right] \quad (31)$$

which uses the derivative chain rule and the assumed commutativity condition. Therefore, $I_{l+1}(t) = \frac{1}{l+1!}(\int_0^t W(\tau) d\tau)^{l+1}$. Then the limit of the quaternion
function sequence of (7) is
\[
q_\infty = \lim_{l \to \infty} \int_0^l \left( \int_0^{l_1} \cdots \int_0^{l_{n-1}} q(0) \cdot \frac{\omega}{2} d\tau_1 \right) \\
\cdots \cdot \frac{\omega}{2} d\tau_1 \\
= \lim_{l \to \infty} \int_0^l W(\tau_1) \int_0^{l_1} W(\tau_2) \cdots \\
\int_0^{l_{n-1}} W(\tau) d\tau_1 \cdots d\tau_2 d\tau_1 \cdot q(0) \\
= \sum_{l=0}^{\infty} \frac{1}{l!} \left( \int_0^l W(\tau) d\tau \right) q(0) \\
= \exp \left\{ \int_0^l W d\tau \right\} q(0). \tag{32}
\]

It can be readily checked that the above \( q_\infty \) satisfies the quaternion kinematic (4).

ACKNOWLEDGMENT
The authors thank Dr. Qi Cai for Matlab code optimization.

REFERENCES
[1] P. D. Groves
Principles of GNSS
In Inertial, and Multisensor Integrated Navigation Systems, 2nd ed. Boston, MA, USA and London, U.K.: Artech House, 2013.
[2] D. H. Titterton and J. L. Weston
Strapdown Inertial Navigation Technology, 2nd ed. London, U.K.: The Institute of Electrical Engineers, 2007.
[3] F. L. Markley and J. L. Crassidis
Fundamentals of Spacecraft Attitude Determination and Control. New York, NY, USA: Springer-Verlag, 2014.
[4] J. W. Jordan
An accurate strapdown direction cosine algorithm
, NASA. Washington, DC, USA, Tech. Rep. NASA-TN-D-5384, 1969.
[5] J. E. Bortz
A new mathematical formulation for strapdown inertial navigation
IEEE Trans. Aerosp. Electron. Syst., vol. 7, no. 1, pp. 61–66, Jan. 1971.
[6] P. G. Savage
Strapdown Analytics, 2nd ed. Maple Plain, MN, USA: Strapdown Associates, 2007.
[7] M. Wang, W. Wu, J. Wang, and X. Pan
High-order attitude compensation in coning and rotation coexisting environment
IEEE Trans. Aerosp. Electron. Syst., vol. 51, no. 2, pp. 1178–1190, Apr. 2015.
[8] C. Rucker
Integrating rotations using nonunit quaternions
IEEE Robot. Autom. Lett., vol. 3, no. 4, pp. 2779–2986, Oct. 2018.
[9] J. Park and W.-K. Chung
Geometric integration on euclidean group with application to articulated multibody systems
IEEE Trans. Robot., vol. 21, no. 5, pp. 850–863, Oct. 2005.
[10] M. S. Andrle and J. L. Crassidis
Geometric integration of quaternions
J. Guid., Control Dyn., vol. 36, no. 2, pp. 1178–1177, 2013.
[11] M. Boyle
The integration of angular velocity
Adv. Appl. Clifford Algebr., vol. 27, pp. 2345–2374, 2017.
[12] P. Krysl and L. Endres
Explicit Newmark/Verlet algorithm for time integration of the rotational dynamics of rigid bodies
Int. J. Numer. Methods Eng., vol. 62, pp. 2154–2177, 2005.
[13] E. Hairer, C. Lubich, and G. Wanner
Geometric Numerical Integration: Structure Preserving Algorithms for Ordinary Differential Equations. New York, NY, USA: Springer-Verlag, 2006.
[14] H. Munthe-Kaas
Runge-Kutta methods on lie groups
BIT Numer. Math., vol. 38, pp. 92–111, 1998.
[15] P. Savage
Down-Summing Rotation Vectors for Strapdown Attitude Updating (SAI WBN-14019).
Maple Plain, MN, USA: Strapdown Associates, 2017. [Online]. Available: http://strapdown associates.com/Rotation%20Vector%20Down_Summing.pdf
[16] Y. Wu
RodFIter: Attitude reconstruction from inertial measurement by functional iteration
IEEE Trans. Aerosp. Electron. Syst., vol. 54, no. 5, pp. 2131–2142, Oct. 2018.
[17] Y. Wu, Q. Cai, and T.-K. Truong
Fast rodftiter for attitude reconstruction from inertial measurement
IEEE Trans. Aerosp. Electron. Syst., vol. 55, no. 1, pp. 419–428, 2019.
[18] G. Yan, J. Weng, X. Yang, and Y. Qin
An accurate numerical solution for strapdown attitude algorithm based on picard iteration
J. Astronaut., vol. 38, pp. 65–71, 2017.
[19] Z. Xu, J. Xie, Z. Zhou, J. Zhao, and Z. Xu
Accurate direct strapdown direction cosine algorithm
IEEE Trans. Aerosp. Electron. Syst., to be published.
[20] V. N. Branets and I. P. Shmyglevsky
Introduction to the Theory of Strapdown Inertial Navigation System, Moscow, Russia: Nauka, 1992.
[21] J. Kuipers
Quaternions and Rotation Sequences. Princeton, NJ, USA: Princeton Univ. Press, 1999.
[22] W. J. Rugh
Linear System Theory, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 1996.
[23] W. H. Press
Numerical Recipes: The Art of Scientific Computing, 3rd ed. Cambridge, U.K.: Cambridge Univ. Press, 2007.
[24] R. W. Brockett
Finite Dimensional Linear Systems. New York, NY, USA: Wiley, 1970.
[25] M. Baake and U. Schlaegel
The Peano Baker Series
Proc. Steklov Inst. Math., vol. 275, pp. 155–159, 2011.
Yuanxin Wu (SM’14) received the B.Sc. and Ph.D. degrees in navigation from the Department of Automatic Control, National University of Defense Technology, Changsha, China, in 1998 and 2005, respectively.

He was a Lecturer and an Associate Professor with the National University of Defense Technology, from 2006 to 2007 and from 2008 to 2012, respectively, a Visiting Postdoctoral Fellow with the Department of Geomatics Engineering, University of Calgary, Calgary, AB, Canada from February 2009 to February 2010, and a Professor with the Central South University, Changsha, China from 2013 to 2015. He is currently a Professor with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests include inertial-based navigation system, state estimation, inertial-visual fusion, and wearable human motion sensing.

Prof. Wu was the recipient of National Excellent Doctoral Dissertation (2008), New Century Excellent Talents in University (2010), Fok Ying Tung Education Fellowship (2012), NSFC Award for Excellent Young Scientists (2014), Natural Science and Technology Award in University (2008, 2016), and Elsevier’s Most Cited Chinese Researchers (Aerospace Engineering, 2015–2018). He serves as an Associate Editor for the Journal of Navigation and for the IEEE TRANSACTIONS ON AEROSPACE AND ELECTRONIC SYSTEMS.

Gongmin Yan received the B.Sc., M.S., and Ph.D. degrees in guidance, navigation, and control from the School of Automation, Northwestern Polytechnic University (NWPU), Xi’an, China, in 2000, 2003, and 2006, respectively.

From 2006 to 2008, he was a Postdoctoral Research Fellow with the School of Marine Technology, NWPU, Xi’an, China. From 2008 to 2019, he was a Lecturer with the School of Automation, NWPU. From 2012 to 2013, he was a Visiting Scholar with the Civil Engineering, University of New South Wales, Sydney, NSW, Australia. He is currently an Associate Professor with the School of Automation, NWPU. His research interests include strapdown inertial navigation system, autonomous navigation system, Kalman filtering, and multisource information fusion.