

Strapdown Inertial Navigation Integration Algorithm Design Part 1: Attitude Algorithms

Paul G. Savage*

Strapdown Associates, Inc., Maple Plain, Minnesota 55359

This series of two papers provides a rigorous comprehensive approach to the design of the principal software algorithms utilized in modern-day strapdown inertial navigation systems: integration of angular rate into attitude, acceleration transformation/integration into velocity, and integration of velocity into position. The algorithms are structured utilizing the two-speed updating approach originally developed for attitude updating in which an analytically exact equation is used at moderate speed to update the integration parameter (attitude, velocity, or position) with input provided from a high-speed algorithm measuring dynamic angular rate/acceleration effects within the parameter update time interval [coning for attitude updating, sculling for velocity updating, and scrolling (writer's terminology) for high-resolution position updating]. The algorithm design approach accounts for angular rate/specific force acceleration measurements from the strapdown system inertial sensors as well as rotation of the navigation frame used for attitude referencing and velocity integration. This paper, Part 1, defines the overall design requirement for the strapdown inertial navigation integration function and develops direction cosine and quaternion forms for the attitude updating algorithms. Part 2 [Savage, P. G., "Strapdown Inertial Navigation Integration Algorithm Design Part 2: Velocity and Position Algorithms," *Journal of Guidance, Control, and Dynamics* (to be published)] deals with design of the velocity and position integration algorithms. Although Parts 1 and 2 often cover fundamental inertial navigation concepts, the material presented is intended for use by the practitioner who is already familiar with basic inertial navigation concepts.

Nomenclature

- A, A_1, A_2, A_3 = arbitrary coordinate frames
 $C_{A_2}^{A_1}$ = direction cosine matrix that transforms a vector from its A_2 frame projection form to its A_1 frame projection form
 I = identity matrix
 $q_{A_2}^{A_1}$ = attitude quaternion that transforms a quaternion vector from its A_2 frame component form to its A_1 frame component form
 $q_{A_2}^{A_1*}$ = attitude quaternion $q_{A_2}^{A_1}$ conjugate having the same first element as $q_{A_2}^{A_1}$ but with the negative of elements 2–4 in $q_{A_2}^{A_1}$
 q_1 = identity quaternion having 1 for the first element and zero for the remaining three
 V = vector without specific coordinate frame designation
 V^A = column matrix with elements equal to the projection of V on frame A axes
 $(V^A \times)$ = skew symmetric (or cross product) form of V^A , represented by the square matrix

$$\begin{bmatrix} 0 & -V_{ZA} & V_{YA} \\ V_{ZA} & 0 & -V_{XA} \\ -V_{YA} & V_{XA} & 0 \end{bmatrix}$$

where V_{XA}, V_{YA}, V_{ZA} are the components of V^A ; matrix product of $(V^A \times)$ with another A -frame vector equals the cross product of V^A with the vector in the A frame

V_q^A = quaternion four vector equivalent to V^A ,

$$\begin{bmatrix} 0 \\ V^A \end{bmatrix}$$

$\omega_{A_1 A_2}$ = angular rate of coordinate frame A_2 relative to coordinate frame A_1 ; when A_1 is the inertial I frame, $\omega_{A_1 A_2}$ is the angular rate measured by angular rate sensors mounted on frame A_2

I. Introduction

INERTIAL navigation is the process of calculating position by integration of velocity and computing velocity by integration of total acceleration. Total acceleration is calculated as the sum of gravitational acceleration, plus the acceleration produced by applied nongravitational forces (known as specific force acceleration). An inertial navigation system (INS) consists of a navigation computer for the integration function, a precision clock for timing the integration operations, an accelerometer assembly for measuring the specific force acceleration, gravitation model software resident in the navigation computer for calculating gravitational acceleration as a function of calculated position, and an attitude reference for defining the angular orientation of the accelerometer triad as part of the velocity calculation. In a modern day INS, the attitude reference is provided by a software integration function residing in the INS computer using inputs from a three-axis set of inertial angular rate sensors. The angular rate sensor and accelerometer triads are mounted to a common rigid structure within the INS chassis to maintain precision alignment between each inertial sensor. Such an arrangement has been denoted as a strapdown INS because of the rigid attachment of the inertial sensors within the chassis, hence, to the vehicle in which the INS is mounted.

The primary functions executed in the INS computer are the angular rate into attitude integration function (denoted as attitude integration), use of the attitude data to transform measured acceleration into a suitable navigation coordinate frame where it is integrated into velocity (denoted as velocity integration), and integration of the navigation frame velocity into position (denoted as position integration). Thus, three integration functions are involved, attitude, velocity, and position, each of which requires high accuracy to assure negligible error compared to inertial sensor accuracy requirements.

From a historical perspective, since the basic strapdown inertial navigation concept was originally formulated in the 1950s, strapdown analysts have primarily focused on the design of algorithms for the attitude integration function. Invariably, the design approaches were driven by the capabilities and limitations of contemporary

Received July 7, 1997; revision received Sept. 11, 1997; accepted for publication Sept. 15, 1997. Copyright © 1997 by Strapdown Associates, Inc. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

*President. Member AIAA.

flight computer technology. In the late 1950s and in the 1960s, two approaches were pursued by strapdown analysts (in various organizations) for the attitude integration function¹⁻⁵: high-speed attitude updating, e.g., 10–20 kHz, using first-order digital algorithms, and lower-speed attitude updating, e.g., 50–100 Hz, using higher-order algorithms. The high-speed approach was promoted as a means for accurately accounting for high-frequency angular rate components that can rectify into systematic three-dimensional attitude change; however, computer technology of that time period was only capable of executing simplified first-order equations of limited accuracy for the attitude updating algorithms. In contrast, the higher-order algorithm proponents touted improved analytical accuracy compared to first-order algorithms; however, the improved accuracy was degraded due to the associated increase in executable operations per attitude update cycle and, hence, a slower attitude update rate to satisfy contemporary computer throughput limitations. Tradeoffs between the two approaches were clouded by the emergence of the attitude quaternion as the preferred approach for the analytical form of the computed attitude parameter (vs the traditional direction cosine matrix attitude representation). For the algorithms investigated during that time period, the quaternion showed improved accuracy in high-frequency angular rate environments.

In 1966, the writer proposed a new two-speed approach for the attitude integration function⁶ whereby the attitude updating operation is divided into two parts: a simple high-speed, first-order algorithm portion coupled with a more complex moderate-speed, higher-order algorithm portion whose input was provided by the high-speed algorithm. The simplified high-speed portion accounted for high-frequency angular oscillations within the attitude update cycle that can rectify into systematic attitude buildup (traditionally denoted as coning). Taken together, the combined accuracy of the two-speed approach was equivalent to operating the higher-order algorithm at the high-speed rate (for improved accuracy); however, due to the simplicity of the high-speed algorithm, the combined computer throughput requirement was no greater than for original high-speed, first-order attitude updating algorithms. The utility of the Ref. 6 two-speed algorithm design approach was limited by its basic analytical formulation as a Picard-type recursive integration of the continuous form attitude rate differential equation in which both the moderate- and high-speed algorithms were generated simultaneously. The complexity of the analytical recursive integration design process limited expansion of the higher-order, moderate-speed algorithm (to only second order in Ref. 6, which was considered acceptable at that time).

In an unrelated design activity, Jordan⁷ in 1969 suggested a two-speed approach for the strapdown attitude updating function in which the analytical formulation at the onset was based on two separately defined calculations: a moderate-speed, classical closed-form, (exact) higher-order attitude updating algorithm based on input attitude change, and a simplified high-speed, second-order integration algorithm that measured the attitude change input for the moderate-speed algorithm. In 1971, Bortz⁸ extended the Jordan concept to have the high-speed calculation based on a differential equation that, when integrated, measures the exact attitude change input to the exact attitude updating algorithm. The exact moderate-speed attitude algorithm can be structured to any specified order of accuracy by truncation of two trigonometric coefficients. In practice, simplified forms of the Bortz attitude change differential equation are used for the high-speed function. References 7 and 8 thereby provided a more general form of the two-speed attitude updating approach in which the moderate-speed, higher-order algorithm and high-speed, simplified algorithm can be independently tailored to meet particular application requirements. (Interestingly, Ref. 8 proposed an analog implementation for a simplified version of the high-speed algorithm.) A secondary benefit derived from the Ref. 7 and 8 two-speed approach (proposed using direction cosines for the exact moderate-speed attitude update operation) is that the moderate-speed portion can also be formulated with an analytically exact, closed-form quaternion updating algorithm using the identical high-speed input applied for direction cosine updating. Thus, the new two-speed approach has equal accuracy for either direction cosine or quaternion updating, both of which derive from analytically exact, closed-form

equations (assuming that Taylor series expansion for trigonometric coefficients is carried out to comparable accuracy order).

Most modern-day strapdown INSs for aircraft utilize attitude updating algorithms based on a two-speed approach. The repetition rate for the moderate-speed algorithm portion, e.g., 50–200 Hz, is typically designed, based on maximum angular rate considerations, to minimize power series truncation error effects in the moderate- and high-speed algorithms. The repetition rate for the high-speed algorithm, e.g., 1–4 kHz for an aircraft INS with 1 n mph 50 percentile radial position error rate, is designed, based on the anticipated strapdown inertial sensor assembly vibration environment, to accurately account for vibration-induced coning effects. Continuing two-speed attitude algorithm development work has centered on variations for the high-speed integration function. Originally conceived as a simple first-order algorithm,⁶ today's high-speed attitude algorithms have taken advantage of increased throughput capabilities in modern-day computers and become higher order for improved accuracy (Refs. 9–11 and 12, Sec. 7.1). While the attitude updating function has been evolving to its current form, very little parallel work has been published on the development of the companion strapdown INS algorithms for acceleration transformation/velocity integration and position integration (the subject of the Ref. 13, Part 2, paper).

This paper, Part 1, defines the overall design requirement for the strapdown inertial navigation integration function and describes a comprehensive design process for developing the attitude integration algorithms based on the two-speed approach. The material presented is a condensed version of Ref. 12, Sec. 7.1 (which is an expansion of material in Ref. 9), emphasizing a more rigorous analytical formulation and the use of exact closed-form equations, where possible, for ease in computer software documentation/validation (which is also consistent with modern-day flight computer technology). Included in the attitude algorithm design process is a rigorous treatment of methods for accounting for navigation coordinate frame rotation during the attitude update time periods.

The paper is organized as follows. Section II provides background material regarding coordinate frames and attitude parameters used. Section III provides a complete set of typical strapdown inertial navigation attitude, velocity, and position equations in continuous differential equation format, which serves as a framework for the equivalent algorithm design process. Section IV develops the two-speed attitude integration algorithm (for both direction cosine and quaternion formulations including navigation frame rotation effects) in a generic form for the high-speed portion and describes a particular form to illustrate the design of one of the classical high-speed, second-order coning computation algorithms. A tabular reference summary of the attitude integration algorithms is presented in Sec. V. Section VI provides a general discussion of the process followed in selecting algorithms for a particular application and establishing their execution rates. Concluding remarks are provided in Sec. VII.

Finally, it is important to recognize that although the original intent of the two-speed approach was to overcome throughput limitations of early computer technology (1965–1975), that limitation is rapidly becoming insignificant with continuing rapid advances in modern high-speed computers. This provides the motivation to return to a simpler single-speed algorithm structure whereby all computations are executed at a repetition rate that is sufficiently high to accurately account for multi-axis high-frequency angular rate and acceleration rectification effects. The two-speed structure presented in both Parts 1 and 2 is compatible with compression into such a single-speed format as explained in the particular sections where the algorithms are formulated.

II. Coordinate Frames and Attitude Orientation Relationships

This section defines the coordinate frames used in this paper and generically describes the properties of the direction cosine matrix, the attitude quaternion, and the rotation vector, attitude parameters utilized to represent the angular relationship between two coordinate frames.

A. Coordinate Frame Definitions

A coordinate frame is an analytical abstraction defined by three consecutively numbered (or lettered) unit vectors that are mutually perpendicular to one another in the right-hand sense. It can be visualized as a set of three perpendicular lines (axes) passing through a common point (origin) with the unit vectors emanating from the origin along the axes. In this paper, the physical locations of the coordinate frame origins are arbitrary. A vector's components (or projections) in a particular coordinate frame equal the dot product of the vector with the coordinate frame unit vectors. The vectors used in this paper are classified as free vectors and, hence, have no preferred location in coordinate frames in which they are analytically described.

The coordinate frames are defined as follows.

1) The E frame is the Earth fixed coordinate frame used for position location definition. It is typically defined with one axis parallel to the Earth polar axis with the other axes fixed to the Earth and parallel to the equatorial plane.

2) The N frame is the navigation coordinate frame having its Z axis parallel to the upward vertical at the local Earth surface referenced position location. It is used for integrating acceleration into velocity and for defining the angular orientation of the local vertical in the E frame.

3) The L frame is the locally level coordinate frame parallel to the N frame but with Z axis parallel to the downward vertical, and X and Y axes along N frame Y and X axes. It is used as the reference for describing the strapdown sensor coordinate frame orientation.

4) The B frame is the strapdown inertial sensor coordinate frame (body frame) with axes parallel to nominal right-handed orthogonal sensor input axes.

5) The I frame is the nonrotating inertial coordinate frame used as a reference for angular rotation measurements. Particular orientations selected for the I frame are discussed in the sections where its orientation is pertinent to analytical operations.

B. Attitude Parameter Definitions

The direction cosine matrix is defined as a square matrix whose columns are an orthogonal set of unit vectors, each equal to a unit vector along a coordinate axis of frame A_2 as projected onto the axes of coordinate frame A_1 :

$$C_{A_2}^{A_1} = \begin{bmatrix} \mathbf{u}_{1A_2}^{A_1} & \mathbf{u}_{2A_2}^{A_1} & \mathbf{u}_{3A_2}^{A_1} \end{bmatrix} \quad (1)$$

where $\mathbf{u}_{iA_2}^{A_1}$ is the unit vector along A_2 frame axis i projected on coordinate frame A_1 axes.

From this basic definition it can be demonstrated that the element in row i , column j of $C_{A_2}^{A_1}$ equals the cosine of the angle between frame A_1 axis i and frame A_2 axis j , that the transpose of $C_{A_2}^{A_1}$ equals its inverse, the columns of $C_{A_2}^{A_1}$ transpose equal frame A_1 axis unit vectors projected on frame A_2 axes, and the product of $C_{A_2}^{A_1}$ with a vector projected on frame A_2 axes equals the components of the vector projected on frame A_1 axes (and the converse for $C_{A_2}^{A_1}$ transpose):

$$\mathbf{V}^{A_1} = C_{A_2}^{A_1} \mathbf{V}^{A_2}, \quad \mathbf{V}^{A_2} = (C_{A_2}^{A_1})^T \mathbf{V}^{A_1} = C_{A_1}^{A_2} \mathbf{V}^{A_1} \quad (2)$$

Equations (2) can be used to derive the direction cosine matrix chain rule,

$$C_{A_1}^{A_3} = C_{A_2}^{A_3} C_{A_1}^{A_2} \quad (3)$$

The rotation vector defines an axis of rotation and magnitude for rotation about the axis. Imagine frame A_1 being rotated from its starting attitude to a new attitude by rotation about the rotation vector through an angle equal to the rotation vector magnitude. Now call frame A_2 the new attitude of frame A_1 . By this definition of frame A_2 , an arbitrarily defined rotation vector uniquely defines the attitude of frame A_2 relative to the original frame A_1 attitude. Conversely, for a given attitude of frame A_2 relative to frame A_1 , a rotation vector can be defined that is consistent with this attitude. Thus, a rotation vector can be used to define the attitude of frame A_2 relative to frame A_1 . Analytically, it can be shown (Refs. 4, 9, and

12, Sec. 3.2.2.1) that the relationship between the rotation vector and the direction cosine matrix is given by

$$C_{A_2}^{A_1} = \left[\mathbf{I} + \frac{\sin \phi}{\phi} (\phi \times) \sin \phi + \frac{(1 - \cos \phi)}{\phi^2} (\phi \times)^2 \right] \quad (4)$$

where ϕ and ϕ are the rotation vector and its magnitude. A unique property of the rotation vector is that it has identical components in the A_1 and A_2 frames (Ref. 12, Sec. 3.2.2.1); hence, ϕ in Eq. (4) represents either ϕ^{A_1} or ϕ^{A_2} .

The attitude quaternion is a four vector, i.e., four components, defined as a function of the rotation vector (Refs. 4 and 9; 12, Sec. 3.2.4; and 14, pp. 73–76)

$$q_{A_2}^{A_1} = \begin{bmatrix} \cos 0.5 \phi \\ \frac{\sin 0.5 \phi}{0.5 \phi} 0.5 \phi \end{bmatrix} \quad (5)$$

From Eq. (5), it is easily verified that the sum of the squares of the $q_{A_2}^{A_1}$ elements is unity. The coordinate frame transformation equations associated with $q_{A_2}^{A_1}$ are in quaternion algebra (Refs. 4, 9, and 12, Sec. 3.2.4.1)

$$V_q^{A_1} = q_{A_2}^{A_1} V_q^{A_2} q_{A_2}^{A_1*}, \quad V_q^{A_2} = q_{A_2}^{A_1*} V_q^{A_1} q_{A_2}^{A_1} = q_{A_1}^{A_2} V_q^{A_1} q_{A_1}^{A_2*} \quad (6)$$

Equations (6) can be used to derive the attitude quaternion chain rule,

$$q_{A_1}^{A_3} = q_{A_2}^{A_3} q_{A_1}^{A_2} \quad (7)$$

C. Attitude Parameter Rate Equations

The rates of change of the Sec. II.B attitude parameters (Refs. 4, 8, 9, and 12, Sec. 3.3) are given by

$$\dot{C}_{A_2}^{A_1} = C_{A_2}^{A_1} (\omega_{IA_2}^{A_2} \times) - (\omega_{IA_1}^{A_1} \times) C_{A_2}^{A_1} \quad (8)$$

$$\dot{q}_{A_2}^{A_1} = \frac{1}{2} q_{A_2}^{A_1} \omega_{IA_2q}^{A_2} - \frac{1}{2} \omega_{IA_1q}^{A_1} q_{A_2}^{A_1} \quad (9)$$

$$\begin{aligned} \dot{\phi} &= \omega_{A_1A_2}^{A_1} + \frac{1}{2} \phi \times \omega_{A_1A_2}^{A_1} \\ &+ \frac{1}{\phi^2} \left[1 - \frac{\phi \sin \phi}{2(1 - \cos \phi)} \right] \phi \times (\phi \times \omega_{A_1A_2}^{A_1}) \end{aligned} \quad (10)$$

III. Continuous Form Strapdown Inertial Navigation Equations

The differential equations that define the primary operations typically performed in a strapdown inertial navigation system (Refs. 9; 12, Chap. 4; and 15, pp. 77–103 and 156–177) are given as follows.

Attitude rate

$$\dot{C}_B^L = C_B^L (\omega_{IB}^B \times) - (\omega_{IL}^L \times) C_B^L \quad (11)$$

or, alternatively,

$$\dot{q}_B^L = \frac{1}{2} q_B^L \omega_{IBq}^B - \frac{1}{2} \omega_{ILq}^L q_B^L \quad (12)$$

Local level frame rotation rate

$$\omega_{IL}^L = C_N^L (\omega_{IE}^N + \omega_{EN}^N) \quad (13)$$

$$\omega_{IE}^N = (C_N^E)^T \omega_{IE}^E \quad (14)$$

$$\omega_{EN}^N = F_C (\mathbf{u}_{ZN}^N \times \mathbf{v}^N) + \rho_{ZN} \mathbf{u}_{ZN}^N \quad (15)$$

Acceleration transformation

$$\mathbf{a}_{SF}^L = C_B^L \mathbf{a}_{SF}^B \quad (16)$$

or, alternatively,

$$\mathbf{a}_{SFq}^L = q_B^L \mathbf{a}_{SFq}^B q_B^{L*} \quad (17)$$

$$\mathbf{a}_{SF}^N = C_L^N \mathbf{a}_{SF}^L \quad (18)$$

Velocity rate

$$\mathbf{g}_P^N = \mathbf{g}^N - (\boldsymbol{\omega}_{IE}^N \times) (\boldsymbol{\omega}_{IE}^N \times) \mathbf{R}^N \quad (19)$$

$$\dot{\mathbf{v}}^N = \mathbf{a}_{SF}^N + \mathbf{g}_P^N - (\boldsymbol{\omega}_{EN}^N + 2\boldsymbol{\omega}_{IE}^N) \times \mathbf{v}^N \quad (20)$$

Position rate

$$\dot{C}_N^E = C_N^E (\boldsymbol{\omega}_{EN}^N \times) \quad (21)$$

$$\dot{h} = \mathbf{u}_{ZN}^N \cdot \mathbf{v}^N \quad (22)$$

where

- \mathbf{R} = position vector from Earth's center to the INS
- \mathbf{v} = velocity (rate of change of position) relative to the Earth defined analytically as the time derivative of \mathbf{R} in the E frame
- h = altitude above the Earth defined as the distance from the INS to the Earth surface measured along a line from the INS that is perpendicular to a tangent plane on the Earth's reference geoid¹⁶ surface
- F_C = curvature matrix (3×3) that is a function of position (C_N^E, h) with elements $3,i$ and $i,3$ equal to zero and the remaining elements symmetrical about the diagonal. For a spherical Earth model, the remaining elements are zero off the diagonal and the reciprocal of the radial distance from Earth center to the INS on the diagonal. For an oblate Earth model, the remaining terms represent the local curvature on the Earth surface projected to the INS altitude (see Ref. 12, Sec. 5.3, for closed-form expression)
- \mathbf{u}_{ZN} = unit vector upward along the geodetic vertical (the Z axis of the N frame)
- ρ_{ZN} = vertical component of $\boldsymbol{\omega}_{EN}^N$; the value selected for ρ_{ZN} depends on the type of N frame utilized (e.g., wander azimuth or free azimuth designed to assure that $\boldsymbol{\omega}_{EN}^N$ is nonsingular for all Earth locations (Refs. 12, Sec. 4.6, and 15, pp. 88–99))
- \mathbf{a}_{SF} = specific force acceleration defined as the acceleration relative to nonrotating inertial space produced by applied nongravitational forces, measured by accelerometers
- \mathbf{g} = mass attraction gravitational acceleration or gravitation (a function of \mathbf{R})
- \mathbf{g}_P = plumb-bob gravity or gravity, which, for a stationary INS, lies along the line of a plumb bob

Analytical models for \mathbf{g} can be found in Refs. 16; 17, Sec. 4.4; and 18, Sec. 6.3. See Ref. 12, Sec. 5.4.1, for N frame components of \mathbf{g}_P .

In performing the strapdown inertial navigation function, the strapdown INS computer integrates the latter attitude rate, velocity rate, and position rate equations using suitable integration algorithms.

The following points are worthy of note regarding the form of the latter navigation equations. Both direction cosine and quaternion attitude forms are shown for the body attitude rate/acceleration transformation operations. Either can be used in practice with virtually identical results. The velocity is defined relative to the Earth (E frame) and the velocity rate equation is written in the locally level defined N frame (for integration into velocity). This is typical for many terrestrial navigation applications, e.g., aircraft INS. Other coordinate frame options are also used for velocity definition and the velocity rate equation, e.g., for tactical and strategic missile guidance. The position rate equations define position as altitude plus the angular orientation of the N frame relative to the E frame [from which latitude and longitude can be extracted and \mathbf{R} calculated (Refs. 12, Secs. 4.5.1 and 4.5.3, and 15, pp. 88, 89)]. Position can also be defined for the position rate equation as simply \mathbf{R} [from which C_N^E and h can be calculated (Ref. 12, Sec. 4.5.4)]. Altitude rate equation (22) appears trivial, but not necessarily when one considers a rotating oblate Earth model, a rotating N frame over the Earth, and the stated altitude definition. Reference 12, Secs. 4.4 and 5.5, shows

that Eq. (22) is exact for a rotating oblate Earth model. If vertical channel gravity/divergence control is to be incorporated to prevent exponentially unstable vertical channel error growth, Eqs. (20) and (22) would include an additional vertical control term (Refs. 12, Sec. 4.4.1; 15, pp. 102–103; and 18, Sec. 10.3).

IV. Attitude Update Algorithms

In this section we develop algorithmic forms for direction cosine matrix rate equation (11) and attitude quaternion rate equation (12) suitable for integration in a digital computer. The algorithms will be structured using what is now the traditional two-speed approach^{6–8} in which analytically exact closed-form equations are applied for the basic attitude update function using inputs from a higher speed algorithm designed to measure attitude change over the basic attitude update cycle.

A. Attitude Direction Cosine Matrix

The updating algorithm for the C_B^L direction cosine matrix is designed to achieve the same numerical result at the attitude update times as would the formal continuous integration of the Eq. (11) \dot{C}_B^L expression at the same time instant. The algorithm is constructed by envisioning the body B frame and local level L frame orientation histories in the digital updating world [produced in Eq. (11) by $\boldsymbol{\omega}_{IB}^B$ and $\boldsymbol{\omega}_{IL}^L$] as being constructed of successive discrete orientations relative to nonrotating inertial space (I) at each update time instant. To be completely general, we also allow that C_B^L updating operations for L frame angular motion may not necessarily occur at the same time instant that C_B^L is updated for B frame motion, e.g., for a multirate digital computation loop structure where C_B^L is updated at a higher rate for B frame rotation than for L frame rotation. In the interests of minimizing computer throughput requirements, the software architecture might have L frame updates occurring 5–10 times slower than B frame updates. The nomenclature we adopt to describe the coordinate frame orientation history is as follows:

- $B_{I(m)}$ = discrete orientation of the body B frame in nonrotating inertial space I at computer update time t_m
- m = computer cycle index for B frame angular motion updates to C_B^L
- $L_{I(n)}$ = discrete orientation of the locally level L frame in nonrotating inertial space (I) at computer update time t_n
- n = computer cycle index for L frame angular motion updates to C_B^L

With these definitions, the general updating algorithm for C_B^L is constructed as follows using the Eq. (3) direction cosine matrix product chain rule:

$$C_{B_{I(m)}}^{L_{I(n-1)}} = C_{B_{I(m-1)}}^{L_{I(n-1)}} C_{B_{I(m)}}^{B_{I(m-1)}} \quad (23)$$

$$C_{B_{I(m)}}^{L_{I(n)}} = C_{L_{I(n-1)}}^{L_{I(n)}} C_{B_{I(m)}}^{L_{I(n-1)}} \quad (24)$$

where

- $C_{B_{I(m-1)}}^{L_{I(n-1)}}$ = C_B^L relating the B frame at time t_{m-1} to the L frame at time t_{n-1}
- $C_{B_{I(m)}}^{L_{I(n)}}$ = C_B^L relating the B frame at time t_m to the L frame at time t_n
- $C_{B_{I(m)}}^{B_{I(m-1)}}$ = direction cosine matrix that accounts for B frame rotation relative to inertial space from its orientation at time t_{m-1} to its orientation at time t_m
- $C_{L_{I(n-1)}}^{L_{I(n)}}$ = direction cosine matrix that accounts for L frame rotation relative to inertial space from its orientation at time t_{n-1} to its orientation at time t_n

The algorithm described by Eqs. (23) and (24) relates body B frame and local-level L frame orientations at separate times and provides for B and L frame inertial angular motion updates to C_B^L at different

update rates. Unlike the B frame (which can be rotating dynamically at 200–300 deg/s), the inertial angular rate of the local level L frame is generally small, equal to Earth's rotation rate plus L frame angular rate relative to the Earth (transport rate, which is typically never larger than a few Earth rates). Consequently, the L frame update can generally be performed at a lower rate than the B frame update with comparable accuracy. Note the update rate requirement for B and L frame motion is based, in part, on minimizing errors in the approximate high-speed algorithm used to measure attitude change (see Secs. IV.A.1 and IV.A.2). The B and L frame motion updates to C_B^L are performed by the $C_{B_{I(m)}}^{B_{I(m-1)}}$ and $C_{L_{I(n-1)}}^{L_{I(n)}}$ terms in Eqs. (23) and (24), algorithms for which are derived separately next.

1. Body Frame Rotation

Equation (23) updates the C_B^L attitude direction cosine matrix using $C_{B_{I(m)}}^{B_{I(m-1)}}$ to account for angular rate of the strapdown sensor (body) B frame relative to nonrotating space ω_{IB}^B . The formal definition for $C_{B_{I(m)}}^{B_{I(m-1)}}$ is

$$C_{B_{I(m)}}^{B_{I(m-1)}} = I + \int_{t_{m-1}}^{t_m} \dot{C}_{B_{I(t)}}^{B_{I(t-1)}} dt \quad (25)$$

where $B(t)$ is the B frame attitude at an arbitrary time in the interval t_{m-1} to t_m .

The $C_{B_{I(m)}}^{B_{I(m-1)}}$ matrix can also be expressed in terms of a rotation vector defining the frame $B_{I(m)}$ attitude relative to frame $B_{I(m-1)}$. Applying Eq. (4) using Taylor series expansion for the coefficient terms obtains

$$\begin{aligned} C_{B_{I(m)}}^{B_{I(m-1)}} &= I + \frac{\sin \phi_m}{\phi_m} (\phi_m \times) + \frac{1 - \cos \phi_m}{\phi_m^2} (\phi_m \times) (\phi_m \times) \\ \frac{\sin \phi_m}{\phi_m} &= 1 - \frac{\phi_m^2}{3!} + \frac{\phi_m^4}{5!} - \dots \\ \frac{(1 - \cos \phi_m)}{\phi_m^2} &= \frac{1}{2!} - \frac{\phi_m^2}{4!} + \frac{\phi_m^4}{6!} - \dots \end{aligned} \quad (26)$$

where ϕ_m is the rotation vector defining the frame $B_{I(m)}$ attitude relative to frame $B_{I(m-1)}$ at time t_m . The ϕ_m rotation vector can be computed by treating ϕ as a general rotation vector defining the general B frame attitude relative to frame $B_{I(m-1)}$ for time greater than t_{m-1} . Then ϕ is calculated as the integral from time t_{m-1} of the general $\dot{\phi}$ equation, with ϕ for Eq. (26) evaluated as the integral solution at time t_m . Treating frame $B_{I(m-1)}$ for ϕ definition as the nonrotating inertial reference frame I , we obtain the following for the general ϕ expression by application of Eq. (10) with general frame A_2 replaced by body frame B and general frame A_1 replaced by inertial frame I for angular rate description:

$$\dot{\phi} = \omega_{IB}^B + \frac{1}{2} \phi \times \omega_{IB}^B + \frac{1}{\phi^2} \left(1 - \frac{\phi \sin \phi}{2(1 - \cos \phi)} \right) \phi \times (\phi \times \omega_{IB}^B) \quad (27)$$

where ϕ is the rotation vector defining the general attitude of frame B relative to frame $B_{I(m-1)}$ for time greater than t_{m-1} . Equation (27), commonly referred to as the Bortz equation,⁸ relates the change in B frame attitude to the B frame inertial angular rate ω_{IB}^B that would be measured by strapdown angular rate sensors.

The attitude rotation vector ϕ_m for Eq. (26) is then obtained as the integral of Eq. (27) from time t_{m-1} , evaluated at time t_m

$$\phi(t) = \int_{t_{m-1}}^t \dot{\phi}(\tau) d\tau, \quad \phi_m = \phi(t_m) \quad (28)$$

where τ is the running integration time variable. To reduce the number of computations involved in calculating ϕ with Eq. (27), simplifying assumptions are incorporated. For example, through a power

series expansion, the scalar multiplier of the $\phi \times (\phi \times \omega_{IB}^B)$ term in Eq. (27) can be approximated as

$$\frac{1}{\phi^2} \left(1 - \frac{\phi \sin \phi}{1 - \cos \phi} \right) = \frac{1}{12} \left(1 + \frac{1}{60} \phi^2 + \dots \right) \approx \frac{1}{12} \quad (29)$$

hence, Eq. (27) to second order in ϕ is given by

$$\dot{\phi} \approx \omega_{IB}^B + \frac{1}{2} \phi \times \omega_{IB}^B + \frac{1}{12} \phi \times (\phi \times \omega_{IB}^B) \quad (30)$$

Through simulation and analysis (analytical expansion under hypothesized analytically definable angular motion conditions), it can be shown that to second-order accuracy in ϕ

$$\frac{1}{2} (\phi \times \omega_{IB}^B) + \frac{1}{12} \phi \times (\phi \times \omega_{IB}^B) \approx \frac{1}{2} \alpha \times \omega_{IB}^B \quad (31)$$

where

$$\alpha(t) = \int_{t_{m-1}}^t \omega_{IB}^B d\tau \quad (32)$$

Equation (31) is extremely significant because it enables Eq. (27) to be simplified to second-order accuracy, i.e., in error to third order in ϕ , by retaining only first-order terms. Thus, Eq. (27) becomes to second-order accuracy

$$\dot{\phi} \approx \omega_{IB}^B + \frac{1}{2} \alpha \times \omega_{IB}^B \quad (33)$$

Substituting Eq. (33), Eq. (28) is given by

$$\phi_m = \int_{t_{m-1}}^{t_m} \left[\omega_{IB}^B + \frac{1}{2} (\alpha(t) \times \omega_{IB}^B) \right] dt \quad (34)$$

Finally, with Eq. (32) we obtain

$$\phi_m = \alpha_m + \beta_m \quad (35)$$

with

$$\alpha(t) = \int_{t_{m-1}}^t \omega_{IB}^B d\tau, \quad \alpha_m = \alpha(t_m) \quad (36)$$

$$\beta_m = \frac{1}{2} \int_{t_{m-1}}^{t_m} (\alpha(t) \times \omega_{IB}^B) dt$$

where β_m is the coning attitude motion from t_{m-1} to t_m . The β_m term has been coined the coning term because it measures the effects of coning motion components present in ω_{IB}^B . Coning motion is defined as the condition whereby an angular rate vector is itself rotating. For ω_{IB}^B exhibiting pure coning motion (the ω_{IB}^B magnitude being constant but the vector rotating), a fixed axis in the B frame that is approximately perpendicular to the plane of the rotating ω_{IB}^B vector will generate a conical surface as the angular rate motion ensues (hence, the term coning to describe the motion). Under coning angular motion conditions, B frame axes perpendicular to ω_{IB}^B appear to oscillate (in contrast with nonconing or spinning angular motion in which axes perpendicular to ω_{IB}^B rotate around ω_{IB}^B).

For situations where ω_{IB}^B is not rotating, it is easily seen from Eq. (36) that $\alpha(t)$ will be parallel to ω_{IB}^B ; hence, the cross product in the β_m integrand will be zero and β_m will be zero. Under these conditions, Eq. (34) reduces to the simplified form

$$\phi_m = \int_{t_{m-1}}^{t_m} \omega_{IB}^B dt \quad (37)$$

when ω_{IB}^B is not rotating. Note that Eq. (37) also applies to the exact ϕ_m Eqs. (27) and (28) for a nonrotating ω_{IB}^B , i.e., without approximation. This is readily verified by observing from Eq. (27) that $\phi(t)$ will initially be aligned with ω_{IB}^B as the $\phi(t)$ integration begins and will then remain parallel to ω_{IB}^B because its cross products with $\phi(t)$ in the $\dot{\phi}(t)$ expression will remain zero. Under these conditions, Eqs. (27) and (28) also reduce to Eq. (37).

Integrated angular rate and coning increment algorithms are discussed next. A discrete digital algorithm form of the α_m integrated

rate and β_m coning expressions in Eq. (36) can be developed by considering β_m to be the value at $t = t_m$ of the general function $\beta(t)$, where from Eq. (36)

$$\beta(t) = \frac{1}{2} \int_{t_{m-1}}^t (\alpha(\tau) \times \omega_{IB}^B) d\tau \quad (38)$$

Let us now consider the integration of Eq. (38) as divided into a portion up to and after a general time t_{l-1} within the t_{m-1} to t_m interval so that Eq. (38) is equivalently

$$\begin{aligned} \beta(t) &= \beta_{l-1} + \Delta\beta(t), & \beta_m &= \beta(t_m) \\ \Delta\beta_l &= \frac{1}{2} \int_{t_{l-1}}^{t_l} (\alpha(t) \times \omega_{IB}^B) dt \end{aligned} \quad (39)$$

where β_{l-1} is the value of $\beta(t)$ at $t = t_{l-1}$ and l is the computer cycle index for $t = t_l$ cycle times. Note that by its definition, the l cycle index is faster than the m cycle index. We now define the next l cycle time point t_l within the t_{m-1} to t_m interval so that at t_l , Eq. (39), including initial conditions, become

$$\begin{aligned} \beta_l &= \beta_{l-1} + \Delta\beta_l, & \beta_m &= \beta_l(t_l = t_m) \\ \beta_l &= 0 \quad \text{at} \quad t = t_{m-1} \\ \Delta\beta_l &= \frac{1}{2} \int_{t_{l-1}}^{t_l} (\alpha(t) \times \omega_{IB}^B) dt \end{aligned} \quad (40)$$

Through a similar process, the $\alpha(t)$ expression for Eq. (40) is obtained by manipulation of $\alpha(t)$ in Eqs. (36) as

$$\begin{aligned} \alpha(t) &= \alpha_{l-1} + \Delta\alpha(t), & \Delta\alpha(t) &= \int_{t_{l-1}}^t \omega_{IB}^B d\tau \\ \Delta\alpha_l &= \Delta\alpha(t_l), & \alpha_l &= \alpha_{l-1} + \Delta\alpha_l \\ \alpha_m &= \alpha_l(t_l = t_m), & \alpha_l &= 0 \quad \text{at} \quad t = t_{m-1} \end{aligned} \quad (41)$$

With Eqs. (41), Eqs. (40) are equivalently

$$\begin{aligned} \Delta\beta_l &= \frac{1}{2}(\alpha_{l-1} + \Delta\alpha_l) + \frac{1}{2} \int_{t_{l-1}}^{t_l} (\Delta\alpha(t) \times \omega_{IB}^B) dt \\ \beta_l &= \beta_{l-1} + \Delta\beta_l, & \beta_m &= \beta_l(t_l = t_m) \\ \beta_l &= 0 \quad \text{at} \quad t = t_{m-1} \end{aligned} \quad (42)$$

Equations (41) and (42) constitute the construct of a digital recursive algorithm at the l computer cycle rate for calculating α_m and the β_m coning term as a summation of changes in α , β over the t_{l-1} to t_l interval. It remains to determine a digital equivalent for the Eq. (42) integral term in $\Delta\beta_l$.

Continuing work in attitude algorithm development has centered on the design of digital algorithms for evaluating the coning equation (42) integral term. In general, the methods utilized assume a general analytical form for the angular rate profile ω_{IB}^B in the t_{l-1} to t_l time interval, e.g., a truncated general polynomial in time. The Eq. (42) integral is then analytically determined as a function of the general rate profile coefficients, e.g., the polynomial coefficients. Finally, the coefficients for the angular rate profile are calculated to fit successive integrated angular rate increment measurements. For the example that follows, the angular rate profile is approximated as a constant plus a linear buildup in time with the constant and ramping coefficients calculated from the current and previous values of $\Delta\alpha_l$. A more sophisticated version of this algorithm might include a parabolic-with-time term in the assumed angular rate profile, utilizing the current, past, and past-past values of $\Delta\alpha_l$ for coefficient determination. Recent work in this area^{10,11} calculates the angular rate profile coefficients from angular rate sensor measurements taken within the $t_l - t_{l-1}$ time interval (an extension of the technique proposed in Ref. 19 for single-speed algorithm enhancement), thereby incorporating a third computation cycle rate into the overall attitude

update process architecture: attitude (C_B^L) update and coning (β_l) update (as discussed thus far) and sensor sampling for the coning update. Refinements on the latter technique^{10,11} utilize a general angular rate profile that is defined directly in terms of its impact on the Eq. (42) integral as a sum of weighted cross products between successive integrated angular rate increment sensor samples taken over the $t_l - t_{l-1}$ time interval (similar to the approach presented in Ref. 19 over the t_{m-1} to t_m interval). The weighting coefficients in the latter case are then optimized for best average performance in a pure coning environment, i.e., ω_{IB}^B constant in magnitude, but rotating. Each of the latter design approaches are based on curve fitting techniques for an assumed angular rate profile shape. Each resulting algorithm behaves differently in rate environments for which it was not designed and in the presence of angular rate sensor quantization noise. Selection of the preferred algorithm should include simulation analysis to confirm acceptable performance under operational rate environments and sensor noise characteristics.

We conclude this section by providing an example of an algorithm for the Eq. (42) integral term based on the body rate term ω_{IB}^B being approximated to first order by the truncated power series expansion

$$\omega_{IB}^B \approx \mathbf{A} + \mathbf{B}(t - t_{l-1}), \quad \mathbf{A} \text{ and } \mathbf{B} = \text{const} \quad (43)$$

References 9–11 and 12, Sec. 7.1.1.1.1, show that for the Eq. (43) motion over the interval from t_{l-2} to t_l

$$\int_{t_{l-1}}^{t_l} (\Delta\alpha(t) \times \omega_{IB}^B) dt = \frac{1}{12}(\Delta\alpha_{l-1} + \Delta\alpha_l) \quad (44)$$

Substituting Eq. (44) into Eq. (42) then yields

$$\Delta\beta_l = \frac{1}{2}(\alpha_{l-1} + \frac{1}{6}\Delta\alpha_{l-1}) \times \Delta\alpha_l \quad (45)$$

Equation (45) has been classified as a second-order algorithm for β_m because it includes current and past l cycle $\Delta\alpha$ products in the $\Delta\beta_l$ equation. From the analysis leading to Eq. (44), the l and $l-1$ $\Delta\alpha$ product term in $\Delta\beta_l$, i.e., the $\frac{1}{6}$ term, stems from the approximation of linearly ramping angular rate in the t_{l-2} to t_l time interval. If the angular rate was approximated as a parabolically varying function of time, a third-order algorithm would result containing l , $l-1$, and $l-2$ $\Delta\alpha$ products. If the angular rate was approximated as a constant over t_{l-1} to t_l , the $\frac{1}{6}$ term for $\Delta\beta_l$ in Eq. (45) would vanish, resulting in a first-order algorithm for β_m . Finally, if angular rates are slowly varying, we can approximate β_m as being equal to zero. Alternatively (and more accurately), we can set the l cycle rate equal to the m cycle rate, which equates β_m in Eqs. (45) to $\Delta\beta_l$ calculated once at time t_m [and noting from the initial condition definition in Eq. (41) that α_{l-1} would be zero]. The latter algorithm was developed in Ref. 4. Note that setting the l and m rates equal can also be achieved by increasing the m rate to match the l rate. The result is a single, high-speed, higher-order algorithm with a simpler software architecture than the two-speed approach, but requiring more throughput. Continuing advances in the speed of modern-day computers may make this the preferred approach for the future.

The overall digital algorithm for α_m and β_m in Eq. (35) is determined from the given results as a composite of Eqs. (41), (42), and (45)

$$\begin{aligned} \Delta\alpha_l &= \int_{t_{l-1}}^{t_l} d\alpha, & \alpha_l &= \alpha_{l-1} + \Delta\alpha_l \\ \alpha_m &= \alpha_l(t_l = t_m), & \alpha_l &= 0 \quad \text{at} \quad t = t_{m-1} \end{aligned} \quad (46)$$

$$\begin{aligned} \Delta\beta_l &= \frac{1}{2}(\alpha_{l-1} + \frac{1}{6}\Delta\alpha_{l-1}) \times \Delta\alpha_l, & \beta_l &= \beta_{l-1} + \Delta\beta_l \\ \beta_m &= \beta_l(t_l = t_m), & \beta_l &= 0 \quad \text{at} \quad t = t_{m-1} \end{aligned} \quad (47)$$

where

- $d\alpha$ = differential integrated angular rate increment, i.e., analytical representation of pulse output from strapdown angular rate sensors, $\omega_{IB}^B dt$
 $\Delta\alpha_l$ = summation of integrated angular rate output increments from angular rate sensors

2. Local Level Frame Rotation

Equation (24) updates the C_B^L attitude direction cosine matrix using $C_{L_{I(n-1)}}^{L_{I(n)}}$ to account for angular rate of the local-level coordinate L frame relative to nonrotating space ω_{IL}^L . The derivation for $C_{L_{I(n-1)}}^{L_{I(n)}}$ directly parallels that used to determine $C_{B_{I(m)}}^{B_{I(m-1)}}$ in Sec. IV.A.1. The formal definition for $C_{L_{I(n-1)}}^{L_{I(n)}}$ is

$$C_{L_{I(n-1)}}^{L_{I(n)}} = I + \int_{t_{n-1}}^{t_n} \dot{C}_{L_{I(n-1)}}^{L_{I(n)}} dt \quad (48)$$

where $L(t)$ is the L frame attitude at an arbitrary time in the interval t_{n-1} to t_n .

The $C_{L_{I(n-1)}}^{L_{I(n)}}$ matrix can also be expressed in terms of the rotation vector defining the frame $L_{I(n)}$ attitude relative to frame $L_{I(n-1)}$. Applying Eq. (4) with Taylor series expansion for the coefficient terms obtains

$$C_{L_{I(n-1)}}^{L_{I(n)}} = I - \frac{\sin \zeta_n}{\zeta_n} (\zeta_n \times) + \frac{(1 - \cos \zeta_n)}{\zeta_n^2} (\zeta_n \times) (\zeta_n \times) \\ \frac{\sin \zeta_n}{\zeta_n} = 1 - \frac{\zeta_n^2}{3!} + \frac{\zeta_n^4}{5!} - \dots \quad (49) \\ \frac{(1 - \cos \zeta_n)}{\zeta_n^2} = \frac{1}{2!} - \frac{\zeta_n^2}{4!} + \frac{\zeta_n^4}{6!} - \dots$$

where ζ_n is the rotation vector defining the frame $L_{I(n)}$ attitude at time t_n relative to the frame $L_{I(n-1)}$ attitude at time t_{n-1} . Note in Eq. (49) that the sign for the $[(\sin \zeta_n / \zeta_n) (\zeta_n \times)]$ term is negative in contrast with the similar term in the Eq. (26) $C_{B_{I(m)}}^{B_{I(m-1)}}$ expression. This is because the $C_{L_{I(n-1)}}^{L_{I(n)}}$ matrix has the opposite phase sense from $C_{B_{I(m)}}^{B_{I(m-1)}}$ [or $C_{A_1}^{A_2}$ in Eq. (4)] in that $C_{L_{I(n-1)}}^{L_{I(n)}}$ transforms vectors from $L_{I(n-1)}$ to $L_{I(n)}$, whereas $C_{B_{I(m)}}^{B_{I(m-1)}}$ transforms vectors from $B_{I(m)}$ to $B_{I(m-1)}$. As such, the $C_{L_{I(n-1)}}^{L_{I(n)}}$ form in Eq. (49) is the transpose of the Eq. (26) $C_{B_{I(m)}}^{B_{I(m-1)}}$ expression form.

Because the t_{n-1} to t_n update cycle is relatively short, ζ_n will be very small in magnitude. Because ω_{IL}^L is small and slowly changing over a typical t_{n-1} to t_n update cycle (due to small changes in velocity and position over this time period) the L frame rate vector ω_{IL}^L can be approximated as nonrotating. The result is that ζ_n for Eq. (49) can be calculated as the integral of the simplified form of the Eq. (10) rotation vector rate equation whereby the cross-product terms are neglected,

$$\zeta_n \approx \int_{t_{n-1}}^{t_n} \omega_{IL}^L dt \quad (50)$$

We note in passing that based on the smallness of ζ_n as already discussed, Eq. (49) for $C_{L_{I(n-1)}}^{L_{I(n)}}$ can also be simplified. For example, a second-order version (accurate to second order in ζ_n) is from Eq. (49),

$$C_{L_{I(n-1)}}^{L_{I(n)}} \approx I - (\zeta_n \times) + \frac{1}{2} (\zeta_n \times) (\zeta_n \times) \quad (51)$$

The computer memory/throughput advantages of utilizing a simplified form of Eq. (49) for $C_{L_{I(n-1)}}^{L_{I(n)}}$ [such as Eq. (51)] are trivial for today's modern computer technology compared to the disadvantages of increased software validation/documentation complexity and loss in accuracy. The accuracy loss is generally minor during

navigation; however, it might not be negligible during initial alignment operations (prior to the start of inertial navigation) where the $C_{L_{I(n-1)}}^{L_{I(n)}}$ matrix is used to apply tilt updates to C_B^L (Refs. 12, Sec. 6.1.2, and 15, pp. 120–121). Initial tilt alignment corrections to C_B^L can be fairly large, e.g., 0.1–1.0 deg, which can produce undesirable errors in C_B^L during the initial alignment process if too simplified a version of Eq. (49) is utilized. The closed-loop servo action of the initial alignment operations would eventually correct the resulting attitude error generated in C_B^L ; however, it could leave a residual orthogonality/normality error in the C_B^L rows (and columns). The result would be the requirement to include an orthogonality/normalization correction algorithm (see Sec. IV.A.3) as an outer loop in the C_B^L update processing.

A discrete digital algorithm for the Eq. (50) ζ_n integral can be constructed by first combining Eqs. (13) and (15) to obtain the ω_{IL}^L integrand and then approximating

$$\omega_{IL}^L \approx C_N^L \left[\omega_{IE_{n-\frac{1}{2}}}^N + \rho_{ZN_{n-\frac{1}{2}}} \mathbf{u}_{ZN}^N + F_{C_{n-\frac{1}{2}}} (\mathbf{u}_{ZN}^N \times \mathbf{v}^N) \right] \quad (52)$$

where the subscript $n - \frac{1}{2}$ is the value for $()$ midway between times t_{n-1} and t_n . Using Eq. (52) in Eq. (50) then obtains

$$\zeta_n \approx C_N^L \left[\omega_{IE_{n-\frac{1}{2}}}^N T_n + \rho_{ZN_{n-\frac{1}{2}}} \mathbf{u}_{ZN}^N T_n + F_{C_{n-\frac{1}{2}}} \left(\mathbf{u}_{ZN}^N \times \sum^j \Delta \mathbf{R}_m^N \right) \right] \quad (53)$$

with ω_{IE}^N evaluated using Eq. (14) and

$$\Delta \mathbf{R}_m^N \equiv \int_{t_{m-1}}^{t_m} \mathbf{v}^N dt \quad (54)$$

where T_n is the computer n cycle update period $t_n - t_{n-1}$ and j is the number of computer m cycles over the t_{n-1} to t_n n -cycle computer update period.

The subscripted $n - \frac{1}{2}$ terms in Eq. (53) are all functions of position, which (from Part 2, Ref. 13) is updated following the attitude update at the n -cycle rate. Hence, to calculate these terms in Eq. (52), an approximate extrapolation formula must be used based on previously computed values of the $()$ parameters. For example, a linear extrapolation formula using the last two computed values for $()$ would be

$$()_{n-\frac{1}{2}} \approx ()_{n-1} + \frac{1}{2} [()_{n-1} - ()_{n-2}] = \frac{3}{2} ()_{n-1} - \frac{1}{2} ()_{n-2} \quad (55)$$

In Part 2 (Ref. 13) we find that the \mathbf{v}^N velocity update follows the attitude update. Therefore, current and past m -cycle values of \mathbf{v}^N are available for evaluating the Eq. (54) integral for $\Delta \mathbf{R}_m^N$. Using a trapezoidal integration algorithm for Eq. (54) obtains

$$\Delta \mathbf{R}_m^N \approx \frac{1}{2} (\mathbf{v}_m^N + \mathbf{v}_{m-1}^N) T_m \quad (56)$$

where T_m is the computer m cycle update period $t_m - t_{m-1}$.

Part 2 (Ref. 13) also develops a high-resolution version of $\Delta \mathbf{R}_m^N$ for precision position updating that accounts for dynamic angular rates and accelerations within the $m - 1$ to m cycle update interval.

3. Normalization and Orthogonalization

From its basic definition in Sec. II.B, the columns (and rows) of C_B^L represent orthogonal unit vectors, which, therefore, should be unity in magnitude (normality condition) and mutually orthogonal to one another (orthogonality condition). In addition to the basic C_B^L update algorithms already described, a normalization and orthogonalization algorithm is frequently included to ensure that the C_B^L rows and columns remain normal and orthogonal. Factors that cause C_B^L orthogonality/normality error include C_B^L orthogonality/normality initialization error, software programming error, roundoff error due to insufficient computer wordlength for the total number of C_B^L algorithm update cycles expected, and insufficient number of terms carried in the Eqs. (26) and (49) Taylor series expansions (truncation error). It is important to note (Ref. 12, Sec. 3.4.1)

that orthogonality and normalization errors can only be produced from errors in the software implementation of Eqs. (23), (24), (26), and (49), not from errors in the algorithms feeding these equations or from inertial sensor input errors. The overall design/verification process for the C_B^L integrational algorithm software must assure error-free programming and acceptable roundoff/truncation error for the angular rate environment anticipated over the expected navigation time period, a readily achievable goal with today's computer/software development technology. Nevertheless, inclusion of a C_B^L orthogonality/normality correction algorithm has been traditionally employed in many strapdown inertial navigation software packages for enhanced accuracy and to relax the more stringent requirement of not allowing any orthogonality/normalization error in the basic C_B^L updating operations. The algorithms used for normalization/orthogonalization are based on the property that the transpose of a direction cosine matrix equals its inverse (see Sec. II.B); consequently, the product of C_B^L with its transpose should be identity. Variations from this condition measure the orthogonality/normality error, which can then be used by a control algorithm in iterative fashion for correction (Refs. 9; 12, Secs. 7.1.1.3; and 15, pp. 216–218).

B. Attitude Quaternion

The updating algorithm for the q_B^L attitude quaternion is designed to achieve the same numerical result at the attitude update times as would the formal continuous integration of the Eq. (12) \dot{q}_B^L expression at the same time instant. The updating algorithm for the q_B^L attitude quaternion is developed following the identical procedure used for the C_B^L updating algorithm derivation in Sec. IV.A. Thus, using the Eq. (7) attitude quaternion chain rule, we write

$$q_{B_{I(m)}}^{L_{I(n-1)}} = q_{B_{I(m-1)}}^{L_{I(n-1)}} q_{B_{I(m)}}^{B_{I(m-1)}} \quad (57)$$

$$q_{B_{I(m)}}^{L_{I(n)}} = q_{L_{I(n-1)}}^{L_{I(n)}} q_{B_{I(m)}}^{L_{I(n-1)}} \quad (58)$$

where

$$\begin{aligned} q_{B_{I(m-1)}}^{L_{I(n-1)}} &= q_B^L \text{ relating the } B \text{ frame at time } t_{m-1} \text{ to the} \\ &\quad L \text{ frame at time } t_{n-1} \\ q_{B_{I(m)}}^{L_{I(n)}} &= q_B^L \text{ relating the } B \text{ frame at time } t_m \text{ to the} \\ &\quad L \text{ frame at time } t_n \\ q_{B_{I(m)}}^{B_{I(m-1)}} &= \text{attitude quaternion that accounts for } B \text{ frame} \\ &\quad \text{rotation relative to inertial space from its} \\ &\quad \text{orientation at time } t_{m-1} \text{ to its orientation} \\ &\quad \text{at time } t_m \\ q_{L_{I(n-1)}}^{L_{I(n)}} &= \text{attitude quaternion that accounts for } L \text{ frame} \\ &\quad \text{rotation relative to inertial space from its} \\ &\quad \text{orientation at time } t_{n-1} \text{ to its orientation at time } t_n \end{aligned}$$

The updates for q_B^L are performed by $q_{B_{I(m)}}^{B_{I(m-1)}}$ and $q_{L_{I(n-1)}}^{L_{I(n)}}$ in Eqs. (57) and (58), algorithms for which are derived separately next.

1. Body Frame Rotation

Equation (57) updates the q_B^L attitude quaternion using $q_{B_{I(m)}}^{B_{I(m-1)}}$ to account for angular rotation rate ω_{IB}^B of the strapdown sensor (body) B frame relative to nonrotating space. The formal definition for $q_{B_{I(m)}}^{B_{I(m-1)}}$ is

$$q_{B_{I(m)}}^{B_{I(m-1)}} = q_1 + \int_{t_{m-1}}^{t_m} \dot{q}_{B_{I(m)}}^{B_{I(m-1)}} dt \quad (59)$$

where $B(t)$ is the B frame attitude at an arbitrary time in the interval t_{m-1} to t_m .

The $q_{B_{I(m)}}^{B_{I(m-1)}}$ attitude quaternion can also be expressed in terms of a rotation vector defining the frame $B_{I(m)}$ attitude relative to frame $B_{I(m-1)}$. Applying Eq. (5) with Taylor series expansion for the coefficient terms obtains

$$q_{B_{I(m)}}^{B_{I(m-1)}} = \begin{bmatrix} \cos 0.5 \phi_m \\ \frac{\sin 0.5 \phi_m}{0.5 \phi_m} 0.5 \phi_m \end{bmatrix} \quad (60)$$

$$\frac{\sin 0.5 \phi_m}{0.5 \phi_m} = 1 - \frac{(0.5 \phi_m)^2}{3!} + \frac{(0.5 \phi_m)^4}{5!} - \dots$$

$$\cos 0.5 \phi_m = 1 - \frac{(0.5 \phi_m)^2}{2!} + \frac{(0.5 \phi_m)^4}{4!} - \dots$$

The ϕ_m rotation vector in Eq. (60) for attitude quaternion updating is identical to ϕ_m used in Sec. IV.A.1 for C_B^L direction cosine matrix updating and is calculated using the identical algorithm provided by Eqs. (35), (41), and (42) or Eqs. (35), (46), and (47).

2. Local Level (L) Frame Rotation

Equation (58) updates the q_B^L attitude quaternion using $q_{L_{I(n-1)}}^{L_{I(n)}}$ to account for angular rate of the local-level coordinate L frame relative to nonrotating space ω_{IL}^L . The formal definition for $q_{L_{I(n-1)}}^{L_{I(n)}}$ is

$$q_{L_{I(n-1)}}^{L_{I(n)}} = q_1 + \int_{t_{n-1}}^{t_n} \dot{q}_{L_{I(n-1)}}^{L_{I(n)}} dt \quad (61)$$

with $L(t)$ in Eq. (61) representing the L frame attitude at an arbitrary time in the interval t_{n-1} to t_n .

The $q_{L_{I(n-1)}}^{L_{I(n)}}$ attitude quaternion can also be expressed in terms of the rotation vector defining the frame $L_{I(n)}$ attitude relative to frame $L_{I(n-1)}$. Applying Eq. (5) with Taylor series expansion for the identical terms yields

$$q_{L_{I(n-1)}}^{L_{I(n)}} = \begin{bmatrix} \cos 0.5 \zeta_n \\ -\frac{\sin 0.5 \zeta_n}{0.5 \zeta_n} 0.5 \zeta_n \end{bmatrix}$$

$$\frac{\sin 0.5 \zeta_n}{0.5 \zeta_n} = 1 - \frac{(0.5 \zeta_n)^2}{3!} + \frac{(0.5 \zeta_n)^4}{5!} - \dots \quad (62)$$

$$\cos 0.5 \zeta_n = 1 - \frac{(0.5 \zeta_n)^2}{2!} + \frac{(0.5 \zeta_n)^4}{4!} - \dots$$

The negative sign on ζ_n accounts for the opposite phase sense of $q_{L_{I(n-1)}}^{L_{I(n)}}$, which describes the frame $L_{I(n-1)}$ attitude relative to frame $L_{I(n)}$ compared with the rotation vector ζ_n phase sense, which describes the frame $L_{I(n)}$ attitude relative to frame $L_{I(n-1)}$. The ζ_n rotation vector in Eqs. (62) is identical to ζ_n used for C_B^L direction cosine matrix updating and is calculated using the identical computational algorithm described in Sec. IV.A.2 and provided by Eqs. (53), (55), and (56).

An approximate form of Eqs. (62) that is comparable in accuracy to direction cosine updating Eq. (51) is readily obtained by substitution and truncation

$$q_{L_{I(n-1)}}^{L_{I(n)}} = \begin{pmatrix} 1 - 0.5(0.5 \zeta_n)^2 \\ -0.5 \zeta_n \end{pmatrix} \quad (63)$$

The comments in Sec. IV.A.2 regarding the advisability of using the simplified Eq. (51) direction cosine local-level frame updating algorithm also apply regarding use of Eq. (63) for attitude quaternion updating rather than the complete Eqs. (62) form.

3. Normalization

To preserve the fundamental attitude quaternion normality characteristic discussed in Sec. II.B, a normalization algorithm is frequently incorporated as an outer-loop function in the q_B^L attitude quaternion updating process. The discussion in Sec. IV.A.3 for direction cosine matrices regarding the need for a normalization/orthogonalization function is equally applicable for the attitude quaternion, the only exception being that orthogonalization has no meaning in the definition for the quaternion (as it does for the attitude direction cosine matrix); hence, the orthogonalization discussion in

Table 1 Summary of strapdown INS attitude computation algorithms

Algorithm function	Input	Output	Equation number
<i>High-speed calculations</i>			
Integrated B frame angular rate increments	$\Delta\alpha_l$	α_l, α_m	(41) or (46)
Coning increment	$\Delta\alpha_l, \alpha_l$	β_m	(42) or (47)
<i>Normal-speed calculations for Earth related parameters</i>			
N frame Earth rate components	C_N^E	ω_{IE}^N	(14)
Vertical transport rate component	C_N^E	ρ_{ZN}	Ref. 12, Sec. 4.6
Curvature matrix	C_N^E, h	F_C	Ref. 12, Sec. 5.3
<i>Normal-speed velocity calculations</i>			
N frame velocity update	—	v^N	Part 2 (Ref. 13)
<i>Normal-speed attitude calculations</i>			
B frame rotation vector	α_m, β_m	ϕ_m	(35)
B frame rotation matrix (for attitude direction cosine matrix updating)	ϕ_m	$C_{B^{l(m)}}^{B^{l(m-1)}}$	(26)
B frame rotation quaternion (for attitude quaternion updating)	ϕ_m	$q_{B^{l(m)}}^{B^{l(m-1)}}$	(60)
Attitude update for B frame rotation (direction cosine matrix form)	$C_{B^{l(m-1)}}^{L^{l(m-1)}}, C_{B^{l(m)}}^{B^{l(m-1)}}$	$C_{B^{l(m)}}^{L^{l(m-1)}}$	(23)
Attitude update for B frame rotation (quaternion form)	$q_{B^{l(m-1)}}^{L^{l(m-1)}}, q_{B^{l(m)}}^{B^{l(m-1)}}$	$q_{B^{l(m)}}^{L^{l(m-1)}}$	(57)
N frame position increment	v^N	ΔR_m^N	(56)
L frame rotation vector	$\omega_{IE}^N, \rho_{ZN}, F_C, \Delta R_m^N$	ζ_n	(53), (55)
L frame rotation matrix for attitude direction cosine matrix updating (exact form)	ζ_n	$C_{L^{l(n-1)}}^{L^{l(n)}}$	(49)
L frame quaternion for attitude quaternion updating (exact form)	ζ_n	$q_{L^{l(n-1)}}^{L^{l(n)}}$	(62)
Attitude update for L frame rotation (direction cosine matrix form)	$C_{B^{l(m)}}^{L^{l(m-1)}}, C_{L^{l(n-1)}}^{L^{l(n)}}$	$C_{B^{l(m)}}^{L^{l(n)}}$	(24)
Attitude update for L frame rotation (quaternion form)	$q_{B^{l(m)}}^{L^{l(m-1)}}, q_{L^{l(n-1)}}^{L^{l(n)}}$	$q_{B^{l(m)}}^{L^{l(n)}}$	(58)
Normalization and orthogonalization corrections (for attitude direction cosine matrix)	C_B^L	C_B^L	Sec. IV.A.3
Normalization corrections (for attitude quaternion)	q_B^L	q_B^L	Sec. IV.B.3
<i>Normal-speed position calculations</i>			
Position direction cosine matrix and altitude update	—	C_N^E, h	Part 2 (Ref. 13)

Sec. IV.A.3 does not apply. If a quaternion normalization algorithm is to be utilized, it is based on comparing the magnitude of q_B^L with unity and using the variation from unity to iteratively update q_B^L with a control algorithm (Refs. 9; 12, Sec. 7.1.2.3; and 15, pp. 216–218).

V. Attitude Integration Algorithm Summary

Table 1 summarizes the algorithms described for the strapdown inertial navigation attitude integration function listed in the order they would be executed in the navigation computer. Table 1 lists the algorithm function, input parameters, output parameters, and equation number.

VI. Algorithm and Execution Rate Selection

Faced with the multitude of potential strapdown inertial navigation algorithms to choose from, the software designer must ultimately choose one set for the application at hand. The algorithms presented in this Part 1 and the subsequent Part 2 (Ref. 13) papers are but one version of many similar algorithms developed over the years by several authors. The process of selecting the algorithm set for a particular application should consider the allowable algorithm error under anticipated angular rates/accelerations/vibrations, the capability of the projected target navigation computer for the required algorithm execution rate, and the complexity of the design procedure for software validation/documentation with the selected algorithms.

Evaluation of candidate algorithm error characteristics is generally performed using computerized time-domain simulators that exercise the algorithms in particular groupings at their selected repetition rates. The simulators generate simulated strapdown inertial sensor angular rate/acceleration profiles for algorithm test input together with known navigation parameter solutions for algorithm output comparison, e.g., Ref. 12, Sec. 11.2. For the attitude algorithms discussed, simplified analytical error models can also be

used to predict high-speed coning algorithm error under specified coning rates/amplitudes as a function of algorithm repetition rate (Refs. 9–11 and 12, Sec. 10). The coning rates/amplitudes must be derived either from empirical data or, more commonly, from analytical models of the sensor assembly mount imbalance and its response to external input vibration at particular frequencies (Ref. 12, Sec. 10). Frequency-domain simulators can be used to evaluate high-speed coning algorithm error under specified input vibration power spectral density profiles and sensor assembly mount imbalance as a function of algorithm repetition rate (Ref. 12, Sec. 10). For example, the coning algorithm described by Eqs. (46) and (47) can be shown by such simulators to have an error rate of 0.00037 deg/h when operated at a 2-kHz repetition rate under exposure to 7.6 g rms wideband random linear input vibration (flat 0.04 g²/Hz density from 20 to 1000 Hz, then decreasing logarithmically to 0.01 g²/Hz at 2000 Hz). The linear vibration generates a 0.0003-rad multi-axis angular oscillation of the sensor assembly with a corresponding coning rate of 9.9 deg/h due to the following typical sensor assembly mount characteristics selected as simulator input parameters: 50-Hz linear vibration mode undamped natural frequency, 0.125 linear vibration mode damping ratio, 71-Hz rotary vibration mode undamped natural frequency, 0.18 rotary vibration mode damping ratio, 5% sensor assembly mount mechanical isolator spring and damping imbalance, and 1.4% sensor assembly center of mass offset from mechanical c.g. mount center (percent of distance between isolators).

The capabilities of modern-day computer and INS software technology make it reasonable to specify that the attitude algorithm error be no greater than 5% of the equivalent error produced by the INS inertial sensors (whose cost increases dramatically with accuracy demands). For an INS with a 0.007-deg/h angular rate sensor bias accuracy requirement (for a typical aircraft INS having 1 n mph 50 percentile radial position error rate), the 0.00037-deg/h coning algorithm error rate satisfies the 5% allowance.

So long as the selected integration algorithm is analytically valid, it can be improved in accuracy by increasing its repetition rate. Continuing computer technology advances (increasing speed and decreasing program memory cost), therefore, tend to diminish any advantages one algorithm might have over another (usually measured, primarily, by accuracy for a given repetition rate and, secondarily, by required program memory). Excessively high repetition rates are to be avoided, however (even if computer throughput allowances permit) to limit error buildup caused by computer finite wordlength effects and rectification of high-frequency multi-axis sensor errors (high-frequency error output from one inertial sensor that is frequency correlated with outputs from sensors in the other axes, denoted as pseudoconing error for the coning computation in Part 1 and pseudosculling error for the sculling part of the velocity calculation in Ref. 13, Part 2). The finite computer wordlength error effect is generally not a major factor with modern computer technology, typically having 64-bit double precision floating point wordlengths. The pseudoconing/sculling issue must be resolved on an individual design basis depending on the characteristics of high-frequency error effects anticipated from the inertial sensor assembly in its operational dynamic environment. A general ground rule to follow in coning/sculling algorithm repetition rate selection is to run the algorithms only as fast as required to accurately measure anticipated real multi-axis high-frequency angular rates/accelerations that can potentially rectify into real attitude/velocity change, but no faster, to minimize the likelihood of rectifying high-frequency sensor output error into attitude/velocity error buildup.

The ultimate selection of algorithms to be used in a particular application is generally made based on the previous experience of the responsible design engineer. The author has had long experience with the algorithms described and feels comfortable adapting them to any strapdown application. They are well defined analytically, can be programmed using a simple sequential software executive structure, readily lend themselves to straightforward validation procedures, and are easily adapted to the requirements and constraints of particular applications.

VII. Concluding Remarks

We have defined the overall requirement for the strapdown inertial navigation integration function (in the form of continuous differential equations) and developed the attitude integration algorithms based on the two-speed updating approach: an exact algorithm for moderate speed updating fed by a simplified high-speed algorithm. The high-speed algorithm contains a simple summing operation of angular rate sensor inputs plus an approximate coning motion integration function. Under conditions where the angular rate vector is not rotating, i.e., zero coning, the coning term becomes zero, the simple summing operation becomes an analytically exact representation of the attitude change, and the overall attitude update operation is error free. Where computer throughput restrictions are not at issue, the two-speed structure presented can be compressed into a single high-speed format by operating the moderate-speed algorithm at the high-speed rate. This general form for the two-speed attitude

algorithm defines a framework for design of the velocity/position integration algorithms in Part 2 (Ref. 13) to have similar characteristics: analytical exactness under constant angular rate/specific force acceleration and using a small approximate high-speed computation to measure deviations from the latter condition (denoted as sculling for the velocity algorithm and scrolling for the position algorithm). A summary of the attitude integration algorithms developed is provided in Table 1 as a listing in the order they would be executed in the navigation computer. A similar table is provided in Part 2 (Ref. 13) for the velocity/position integration algorithms.

References

- ¹Minor, J. W., "Low-Cost Strapdown-Inertial Systems," AIAA/ION Guidance and Control Conf., Aug. 1965.
- ²Hills, F. B., "A Study of Coordinate-Conversion Errors in Strapped-Down Navigation," MIT Electronics Systems Lab., E SL-4-244, Massachusetts Inst. of Technology, Cambridge, MA, Aug. 1965.
- ³Turley, A. R., "A Solution for the Problems of the No-Gimbal Inertial Navigator Concept," Air Force Avionics Lab., AFAL-TR-64-307, Wright-Patterson AFB, OH, Jan. 1965.
- ⁴Mckern, R. A., "A Study of Transformation Algorithms for Use in a Digital Computer," M.S. Thesis, Dept. of Aeronautics and Astronautics, Massachusetts Inst. of Technology, Cambridge, MA, Jan. 1968.
- ⁵United Aircraft Corp., "A Study of Critical Computational Problems Associated with Strapdown Inertial Navigation Systems," NASA CR-968, April 1968.
- ⁶Savage, P. G., "A New Second-Order Solution for Strapped-Down Attitude Computation," AIAA/JACC Guidance and Control Conf., Aug. 1966.
- ⁷Jordan, J. W., "An Accurate Strapdown Direction Cosine Algorithm," NASA TN-D-5384, Sept. 1969.
- ⁸Bortz, J. E., "A New Mathematical Formulation for Strapdown Inertial Navigation," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-7, No. 1, 1971, pp. 61-66.
- ⁹Savage, P. G., "Strapdown System Algorithms," *Advances in Strapdown Inertial Systems*, NATO AGARD Lecture Series No. 133, May 1984.
- ¹⁰Ignagni, M. B., "Optimal Strapdown Attitude Integration Algorithms," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 2, 1990, pp. 363-369.
- ¹¹Ignagni, M. B., "Efficient Class of Optimized Coning Compensation Algorithms," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 2, 1996, pp. 424-429.
- ¹²Savage, P. G., *Strapdown Analytics*, Strapdown Associates, Inc., Maple Plain, MN (to be published).
- ¹³Savage, P. G., "Strapdown Inertial Navigation Integration Algorithm Design Part 2: Velocity and Position Algorithms," *Journal of Guidance, Control, and Dynamics* (to be published).
- ¹⁴Morse, P. M., and Feshbach, H., *Methods of Theoretical Physics*, McGraw-Hill, New York, 1953.
- ¹⁵Savage, P. G., *Introduction to Strapdown Inertial Navigation Systems*, 8th printing, Strapdown Associates, Inc., Maple Plain, MN, 1997.
- ¹⁶"Department of Defense World Geodetic System 1984," Defense Mapping Agency, Technical Rept. DMATR 8350.2, 2nd printing, Sept. 1987.
- ¹⁷Britting, K. R., *Inertial Navigation System Analysis*, Wiley, New York, 1971.
- ¹⁸Macomber, G. R., and Fernandes, M., *Inertial Guidance Engineering*, Prentice-Hall, Englewood Cliffs, NJ, 1962.
- ¹⁹Miller, R., "A New Strapdown Attitude Algorithm," *Journal of Guidance, Control, and Dynamics*, Vol. 6, No. 4, 1983, pp. 287-291.