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
- $C^A_{A_2}$: direction cosine matrix that transforms a vector from its $A_2$ frame projection form to its $A_1$ frame projection form
- $I$: identity matrix
- $q_{A_1}^A$: 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 $q_{A_1}^A$, conjugate having the same first element as $q_{A_1}^A$ but with the negative of elements 2–4 in $q_{A_1}^A$
- $q_1$: identity quaternion having 1 for the first element and zero for the remaining three
- $V$: vector without specified 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

\[
\begin{pmatrix}
0 & -V_A & V_Y \\
V_A & 0 & -V_Z \\
-V_Y & V_Z & 0
\end{pmatrix}
\]

where $V_{X_A}, V_{Y_A}, V_{Z_A}$ 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 $V^A$
- $\omega_{A_1 A_2}$: 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 shown to be 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
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 function\textsuperscript{3}: 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 function\textsuperscript{6} 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 being modified for direction cosine attitude updating. 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, Jordan\textsuperscript{7} 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 separate coordinate algorithm portions: 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, Bortz\textsuperscript{8} 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 then 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 algorithm for 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 percent 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\textsuperscript{5} 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 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 generic design 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 described 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 downward vertical, and \( X \) and \( Y \) axes along \( N \) frame \( X \) and \( Y \) axes. 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 \( X \) and \( Y \) 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 orthonormal 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 \( A_1 \):

\[
C_{A_1}^{A_2} = \begin{bmatrix}
    u_{A_1x}^{A_2} & u_{A_1y}^{A_2} & u_{A_1z}^{A_2}
\end{bmatrix}
\]

(1)

where \( u_{A_1i}^{A_2} \) 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 \( C_{A_1}^{A_2} \) equals the cosine of the angle between frame \( A_1 \) axis \( i \) and frame \( A_2 \) axis \( j \), that is, the transpose of \( C_{A_1}^{A_2} \) equals its inverse, the columns of \( C_{A_1}^{A_2} \) transpose equal frame \( A_1 \) axis unit vectors projected on frame \( A_2 \) axes, and the product of \( C_{A_1}^{A_2} \) 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 \( C_{A_1}^{A_2} \) transpose):

\[
V_{A_1} = C_{A_1}^{A_2} V_{A_2} \quad \text{and} \quad \bar{V}_{A_2} = (C_{A_1}^{A_2})^T V_{A_1} = C_{A_2}^{A_1} V_{A_1}
\]

(2)

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

\[
C_{A_1}^{A_2} = C_{A_1}^{A_3} C_{A_3}^{A_2}
\]

(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

\[
\begin{align*}
C_{A_1}^{A_2} &= \left[ 1 + \frac{\sin \phi}{\phi} (\phi \times \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}) + \frac{(1 - \cos \phi)}{\phi^2} (\phi \times \phi) \right] \\
&= \begin{bmatrix}
    \cos \phi & 0 & 0 \\
    0 & \cos \phi & 0 \\
    0 & 0 & 1
\end{bmatrix}
\end{align*}
\]

(4)

where \( \phi \) and \( \phi \) 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, \( \phi \) in Eq. (4) represents either \( \phi_1 \) or \( \phi_2 \).

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)

\[
q_{A_2}^{A_1} = \begin{bmatrix}
    \cos \frac{0.5 \phi}{2} & -\sin \frac{0.5 \phi}{2} & 0 & 0
\end{bmatrix}
\]

(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).

\[
V_{A_1} = q_{A_2}^{A_1} V_{A_1} q_{A_2}^{-1}, \quad q_{A_2}^{A_1} = q_{A_1}^{A_2} q_{A_1}^{-1}, \quad q_{A_1}^{A_2} q_{A_2}^{A_1} = q_{A_1}^{A_2} q_{A_2}^{A_1}
\]

(6)

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

\[
q_{A_1}^{A_2} = q_{A_1}^{A_3} q_{A_3}^{A_2}
\]

(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

\[
\begin{align*}
\dot{C}_{A_1}^{A_2} &= C_{A_1}^{A_3} (\omega_{I_1}^{A_3} \times) - (\omega_{I_1}^{A_3} \times) C_{A_3}^{A_2} \\
\dot{q}_{A_2}^{A_1} &= \frac{1}{2} \varphi \times \omega_{A_1}^{A_2} \\
\phi &= \omega_{A_1}^{A_2} + \frac{1}{2} \varphi \times \omega_{A_1}^{A_2} \\
&= \frac{1}{2} \left[ \varphi \times \left( \omega_{A_1}^{A_2} \times \omega_{A_1}^{A_2} \right) \right]
\end{align*}
\]

(8)

(9)

(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**

\[
\dot{C}_{I_1}^{I_2} = C_{I_1}^{I_2} (\omega_{I_1}^{I_2} \times) - (\omega_{I_1}^{I_2} \times) C_{I_2}^{I_2}
\]

(11)

or, alternatively,

\[
\dot{q}_{I_1}^{I_2} = \frac{1}{2} q_{I_1}^{I_2} \partial_{I_1}^{I_2} - \frac{1}{2} q_{I_1}^{I_2} \partial_{I_1}^{I_2}
\]

(12)

**Local level frame rotation rate**

\[
\omega_{I_2}^{I_2} = C_{I_2}^{E} (\omega_{I_2}^{E} + \omega_{I_2}^{E})
\]

(13)

\[
\omega_{I_2}^{E} = C_{I_2}^{E} (\omega_{I_2}^{E} \times) \omega_{I_2}^{E}
\]

(14)

\[
\omega_{I_2}^{E} = F_C (a_{2N}^{E} \times v^N) + \partial_{2N}^{E} u_{2N}^{E}
\]

(15)

**Acceleration transformation**

\[
a_{I_1}^{I_2} = C_{I_1}^{I_2} a_{I_1}^{I_2}
\]

(16)

or, alternatively,

\[
a_{I_1}^{I_2} = q_{I_1}^{I_2} q_{I_1}^{I_2} a_{I_1}^{I_2}
\]

(17)

\[
a_{I_1}^{I_2} = C_{I_1}^{I_2} a_{I_1}^{I_2}
\]

(18)
Velocity rate
\[ g^N_N = g^N - (\omega^N_{1E} \times (\omega^N_{1E} \times R^N)) \]  
\[ (19) \]
\[ v^N = a^N_{SG} + g^N_p - (\omega^N_{EN} + 2\omega^N_{2E}) \times v^N \]  
\[ (20) \]
Position rate
\[ C^E_N = C^E_N (\omega^E_{EN} \times ) \]  
\[ h = a^N_{ZU} \times v^N \]  
\[ (22) \]

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

Analytical models for \( g \) 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 \( g_p \).

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/rotation 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 \( R \) 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 \( R \) [from which \( C^E_N \) and \( h \) can be calculated (Ref. 12, Sec. 4.5.4)]. Attitude 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 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 \( C^E_N \) 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) \( C^E_N \) 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 \( \omega^B_{1} \) and \( \omega^B_{2} \) 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 \( C^L_B \) updating operations for \( L \) frame angular motion may not necessarily occur at the same time instant that \( C^E_N \) is updated for \( B \) frame motion, e.g., for a multirate digital computation loop structure where \( C^L_B \) 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:

\[ B_{1(m)} = \text{discrete orientation of the body } B \text{ frame in nonrotating inertial space } I \text{ at computer update time } t_m \]
\[ m = \text{computer cycle index for } B \text{ frame angular motion updates to } C^L_B \]
\[ L_{1(m)} = \text{discrete orientation of the locally level } L \text{ frame in nonrotating inertial space } I \text{ at computer update time } t_m \]
\[ n = \text{computer cycle index for } L \text{ frame angular motion updates to } C^E_N \]

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

\[ C^L_{B_{m(1)}} C^B_{B_{m(1)}} C^B_{B_{m(1)}} = C^L_{B_{m(n)}} C^L_{I_{m(n)}} C^E_{B_{m(n)}} \]
\[ C^L_{B_{m(n)}} C^B_{B_{m(n)}} C^B_{B_{m(n)}} = C^L_{B_{m(n-1)}} C^L_{I_{m(n-1)}} C^E_{B_{m(n-1)}} \]
\[ C^L_{B_{m(n)}} C^B_{B_{m(n)}} C^B_{B_{m(n)}} = C^L_{I_{m(n)}} C^E_{B_{m(n)}} \]

where

\[ C^L_{B_{m(1)}} C^B_{B_{m(1)}} C^B_{B_{m(1)}} = C^L_{B_{m(n)}} C^L_{I_{m(n)}} C^E_{B_{m(n)}} \]
\[ C^L_{B_{m(n)}} C^B_{B_{m(n)}} C^B_{B_{m(n)}} = C^L_{B_{m(n-1)}} C^L_{I_{m(n-1)}} C^E_{B_{m(n-1)}} \]
\[ C^L_{B_{m(n)}} C^B_{B_{m(n)}} C^B_{B_{m(n)}} = C^L_{I_{m(n)}} C^E_{B_{m(n)}} \]

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 \( C^E_N \) 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 $C_{B}^{L}(n-1)$ and $C_{L}^{B}(n)$ terms in Eqs. (23) and (24), algorithms for which are derived separately next.

1. **Body Frame Rotation**

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

$$C_{B}^{I}(n-1) = 1 + \int_{t_{n-1}}^{t_{n}} C_{B}^{I}(\tau) \, d\tau$$

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

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

$$C_{B}^{I}(n-1) = 1 + \frac{1}{2!} \sin \phi_{m} \cos \phi_{m} (\phi_{m} \times) (\phi_{m} \times)$$

$$+ \frac{1}{3!} (1 - \cos \phi_{m}) \phi_{m}^{2} - \cdots$$

$$+ \frac{1}{2!} \sin \phi_{m} \cos \phi_{m} (\phi_{m} \times) (\phi_{m} \times)$$

where $\phi_{m}$ is the rotation vector defining the frame $B_{I}$ attitude relative to frame $B_{I-1}$ at time $t_{n}$. The $C_{B}^{I}$ rotation vector can be computed by treating $\phi_{m}$ as a general rotation vector defining the general $B$ frame attitude relative to frame $B_{I-1}$ for time greater than $t_{n-1}$. Then $\phi_{m}$ is calculated as the integral from $t_{n-1}$ to the general $B$ frame attitude, with $\phi_{m}$ for Eq. (26) evaluated as the integral solution at time $t_{n}$. Treating frame $B_{I}$ for $\phi_{m}$ as the nonrotating inertial reference frame $I$, we obtain the following for the general $\phi_{m}$ 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:

$$\phi = \omega_{B}^{I} + \frac{1}{2} \phi \times \omega_{B}^{I} + \frac{1}{3!} (1 - \cos \phi_{m}) \phi \times (\phi \times \omega_{B}^{I})$$

(27)

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

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

$$\phi(t) = \int_{t_{n-1}}^{t_{n}} \phi(\tau) \, d\tau, \quad \phi_{m} = \phi(t_{n})$$

(28)

where $\tau$ is the running integration time variable. To reduce the number of computations involved in calculating $\phi$ with Eq. (27), simplifying assumptions are incorporated. For example, through a power series expansion, the scalar multiplier of the $\phi \times (\phi \times \omega_{B}^{I})$ term in Eq. (27) can be approximated as

$$\frac{1}{\phi_{m}} \left( 1 - \frac{\phi \sin \phi}{1 - \cos \phi} \right) = \frac{1}{12} \left( 1 + \frac{1}{60} \phi_{m}^{2} + \cdots \right) \approx \frac{1}{12}$$

hence, Eq. (27) to second order in $\phi$ is given by

$$\phi \approx \omega_{B}^{I} + \frac{1}{12} \phi \times \omega_{B}^{I} + \frac{1}{720} \phi \times (\phi \times \omega_{B}^{I})$$

(30)

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

$$\frac{1}{2} (\phi \times \omega_{B}^{I}) + \frac{1}{720} \phi \times (\phi \times \omega_{B}^{I}) \approx \frac{1}{4} \phi \times \omega_{B}^{I}$$

(31)

where

$$\alpha(t) = \int_{t_{n-1}}^{t_{n}} \omega_{B}^{I} \, d\tau$$

(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 $\phi$, by retaining only first-order terms. Thus, Eq. (27) becomes to second-order accuracy

$$\phi \approx \omega_{B}^{I} + \frac{1}{12} \phi \times \omega_{B}^{I}$$

(33)

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

$$\phi_{m} = \int_{t_{n-1}}^{t_{n}} \omega_{B}^{I} + \frac{1}{12} (\alpha(t) \times \omega_{B}^{I}) \, d\tau$$

(34)

Finally, with Eq. (32) we obtain

$$\alpha(t) = \int_{t_{n-1}}^{t_{n}} \omega_{B}^{I} \, d\tau, \quad \alpha_{m} = \alpha(t_{n})$$

(36)

where $\beta_{m}$ is the coning attitude motion from $t_{n-1}$ to $t_{n}$. The $\beta_{m}$ term has been coined the coning term because it measures the effects of coning motion components present in $\omega_{B}^{I}$. Coming motion is defined as the condition whereby an angular rate vector is itself rotating. For $\omega_{B}^{I}$ exhibiting pure coning motion (the $\omega_{B}^{I}$ magnitude being constant but the vector rotating), a fixed axis in the $B$ frame that is approximately perpendicular to the plane of the rotating $\omega_{B}^{I}$ 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 $\omega_{B}^{I}$ appear to oscillate (in contrast with nonconing or spinning angular motion in which axes perpendicular to $\omega_{B}^{I}$ rotate around $\omega_{B}^{I}$).

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

$$\phi_{m} = \int_{t_{n-1}}^{t_{n}} \omega_{B}^{I} \, d\tau$$

(37)

when $\omega_{B}^{I}$ is not rotating. Note that Eq. (37) also applies to the exact $\phi_{m}$ Eqs. (27) and (28) for a nonrotating $\omega_{B}^{I}$, i.e., without approximation. This is readily verified by observing from Eq. (27) that $\phi(t)$ will initially be aligned with $\omega_{B}^{I}$ as the $\phi(t)$ integration begins and will then remain parallel to $\omega_{B}^{I}$ because its cross products with $\phi(t)$ in the $\phi(t)$ 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 $\alpha_{m}$ integrated
rate and $\beta_\alpha$ coning expressions in Eq. (36) can be developed by
considering $\beta_\alpha$ to be the value at $t = t_m$ of the general function
$\beta(t)$, where from Eq. (36)

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

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

$$\beta(t) = \beta_{t-1} + \Delta \beta(t), \quad \beta_m = \beta(t_m) \tag{39}$$

$$\Delta \beta_i = \frac{1}{2} \int_{t_{i-1}}^{t_i} \left( \alpha(t) \times \omega_{lb}^B \right) \, dt$$

where $\beta_{t-1}$ is the value of $\beta(t)$ at $t = t_{i-1}$ and $l$ is the computer
index for $t = t_i$ 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 $t_i$ within the $t_{m-1}$ to $t_m$ interval so that at $t_i,
Eq. (39)$, including initial conditions, become

$$\beta_i = \beta_{t-1} + \Delta \beta_i, \quad \beta_m = \beta(t_i = t_m)$$

$$\beta_i = 0 \quad \text{at} \quad t = t_{m-1} \tag{40}$$

$$\Delta \beta_i = \frac{1}{2} \int_{t_{i-1}}^{t_i} \left( \alpha(t) \times \omega_{lb}^B \right) \, dt$$

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

$$\alpha(t) = \alpha_{t-1} + \Delta \alpha(t), \quad \Delta \alpha(t) = \int_{t_{i-1}}^{t} \omega_{lb}^B \, d\tau \tag{41}$$

$$\Delta \alpha_m = \Delta \alpha(t_i), \quad \alpha_i = \alpha_{t-1} + \Delta \alpha_i$$

$$\alpha_m = \alpha_i(t_i = t_m), \quad \alpha_i = 0 \quad \text{at} \quad t = t_{m-1} \tag{42}$$

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

$$\Delta \beta_i = \frac{1}{2} \Delta \alpha_{t-1} + \Delta \alpha_i + \frac{1}{2} \int_{t_{i-1}}^{t_i} \left( \Delta \alpha(t) \times \omega_{lb}^B \right) \, dt$$

$$\beta_i = \beta_{t-1} + \Delta \beta_i, \quad \beta_m = \beta(t_i = t_m)$$

$$\beta_i = 0 \quad \text{at} \quad t = t_{m-1} \tag{43}$$

Equations (41) and (42) constitute the construct of a digital recursive
algorithm at the computer cycle rate for calculating $\alpha_m$ and the $\beta_m$
coning term as a summation of changes in $\alpha, \beta$ over the $t_{i-1}$ to $t_i$ time interval. It remains to determine a digital equivalent for the Eq. (42)
integral term in $\Delta \beta_i$.

Continuing work in attitude algorithm development has centered
on the design of digital algorithms for evaluating the coning equa-
tion (42) integral term. In general, the methods utilized assume a
general analytical form for the angular rate profile $\omega_{lb}^B$ in the $t_{i-1}$
to $t_i$ 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 ramp-
ing coefficients calculated from the current and previous values of $\Delta \alpha_i.$ A more sophisticated version of this algorithm might include a
parabolic-with-time term in the assumed angular rate profile, utiliz-
ing the current, past, and past past-values of $\Delta \alpha_i$ for coefficient de-
termination. Recent work in this area[10,11] calculates the angular rate
profile coefficients from angular rate sensor measurements taken
within the $t_{i-1}$ to $t_{i+1}$ time interval (an extension of the technique pro-
bposed in Ref. 19 for single-speed algorithm enhancement), thereby
incorporating a third computation cycle rate into the overall attitude
update process architecture.

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

$$\omega_{lb}^B = A + B(t - t_{i-1}), \quad A$ and $B = \text{const} \tag{43}$$

References 9–11 and 12, Sec. 7.1.1.1.1, show that for the Eq. (43)
motion over the interval from $t_{i-1}$ to $t_i$

$$\int_{t_{i-1}}^{t_i} \left( \Delta \alpha(t) \times \omega_{lb}^B \right) \, dt = \frac{1}{12} \left( \Delta \alpha_{t-1} + \Delta \alpha_t \right) \tag{44}$$

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

$$\Delta \beta_i = \frac{1}{2} (\alpha_{t-1} + \Delta \alpha_{t-1}) \times \Delta \alpha_i \tag{45}$$

Equation (45) has been classified as a second-order algorithm for $\beta_\alpha$
because it includes current and past $l$ cycle $\Delta \alpha$ products in the $\Delta \beta_i$
equation. From the analysis leading to Eq. (44), the $l$ and $l - 1 \Delta \alpha$
products term in $\Delta \beta_i$, i.e., the $^2$ term, stems from the approxima-
tion of linearly ramping angular rate in the $t_{i-2}$ to $t_i$ 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 $t_{i-2}$ to $t_i$, the $^2$ term for $\Delta \beta_i$ in Eq. (45) would vanish,
resulting in a first-order algorithm for $\beta_\alpha.$ Finally, if angular
rates are slowly varying, we can approximate $\beta_\alpha$ as being equal
to zero. Alternatively (and more accurately), we can set the $l$ cycle
rate equal to the $m$ cycle rate, which equates $\beta_m$ in Eqs. (45) to
$\Delta \beta_i$ calculated once at time $t_m$ (and noting from the initial
condition definition in Eq. (41) that $\alpha_{t-1}$ 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 $\alpha_m$ and $\beta_m$ in Eq. (35) is deter-
mined from the given results as a composite of Eqs. (41), (42), and (45)

$$\Delta \alpha_i = \int_{t_{i-1}}^{t_i} \, d\alpha, \quad \alpha_i = \alpha_{t-1} + \Delta \alpha_i \tag{46}$$

$$\alpha_m = \alpha_i(t_i = t_m), \quad \alpha_i = 0 \quad \text{at} \quad t = t_{m-1}$$

$$\Delta \beta_i = \frac{1}{2} (\alpha_{t-1} + \Delta \alpha_{t-1}) \times \Delta \alpha_i, \quad \beta_i = \beta_{t-1} + \Delta \beta_i \tag{47}$$

$$\beta_m = \beta_i(t_i = t_m), \quad \beta_i = 0 \quad \text{at} \quad t = t_{m-1}$$
where
\[ d\alpha = \text{differential integrated angular rate increment,} \]
\[ \Delta \alpha = \text{summation of integrated angular rate output} \]
\[ \text{increments from angular rate sensors} \]

### 2. Local Level Frame Rotation

Equation (24) updates the \( C_L^{(0)} \) attitude direction cosine matrix using \( C_{L(n)}^{(0)} \) to account for angular rate of the local-level coordinate \( L \) frame relative to nonrotating space \( \omega_{LB}^{(t)} \). The derivation for \( C_{L(n)}^{(0)} \) directly parallels that used to determine \( C_{B(n)}^{(0)} \) in Sec. IV.A.1. The formal definition for \( C_{L(n)}^{(0)} \) is

\[ C_{L(n)}^{(0)} = 1 + \int_{t_{n-1}}^{t_n} \frac{d}{dt} C_{L(n)}^{(0)} \, dt \tag{48} \]

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

The \( C_{L(n)}^{(0)} \) matrix can also be expressed in terms of the rotation vector defining the \( L \) frame relative to \( L_{L(n)}^{(0)} \). Applying Eq. (4) with Taylor series expansion for the coefficient terms obtains

\[ C_{L(n)}^{(0)} = 1 - \sin \frac{\mathbf{\alpha}}{\theta} (\mathbf{\alpha} \times) + (1 - \cos \frac{\mathbf{\alpha}}{\theta}) (\mathbf{\alpha} \times) ) (\mathbf{\alpha} \times) \]

\[ \sin \frac{\mathbf{\alpha}}{\theta} = 1 - \frac{c^2}{2!} + \frac{c^4}{4!} - \cdots \tag{49} \]

where \( \mathbf{\alpha} \) is the rotation vector defining the \( L \) frame attitude at time \( t \), relative to the \( L_{L(n)}^{(0)} \) attitude at time \( t_{n-1} \). Note in Eq. (49) that the term for \( (\sin \frac{\mathbf{\alpha}}{\theta}) (\mathbf{\alpha} \times) ) \) is negative in contrast with the similar term in the Eq. (26) \( C_{B(n)}^{(0)} \) expression.

This is because the \( C_{L(n)}^{(0)} \) matrix has the opposite phase sense from \( C_{B(n)}^{(0)} \) or \( C_{B(n)}^{(0)} \) in Eq. (4) in that \( C_{L(n)}^{(0)} \) transforms vectors from \( L_{L(n)}^{(0)} \) to \( L_{L(n)}^{(0)} \), whereas \( C_{B(n)}^{(0)} \) transforms vectors from \( B_{B(n)}^{(0)} \) to \( B_{B(n)}^{(0)} \). As such, the \( C_{L(n)}^{(0)} \) form in Eq. (49) is the transpose of the Eq. (26) \( C_{B(n)}^{(0)} \) expression form.

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

\[ \mathbf{\alpha} \approx \int_{t_{n-1}}^{t_n} \omega_{LB}^{(t)} \, dt \tag{50} \]

We note in passing that based on the smallness of \( \mathbf{\alpha} \), as already discussed, Eq. (49) for \( C_{L(n)}^{(0)} \) can also be simplified. For example, a second-order version (accurate to second order in \( \mathbf{\alpha} \)) is from Eq. (49).

\[ C_{L(n)}^{(0)} \approx 1 - (\mathbf{\alpha} \times) + \frac{1}{2} (\mathbf{\alpha} \times) (\mathbf{\alpha} \times) \tag{51} \]

The computer memory/throughput advantages of utilizing a simplified form of Eq. (49) for \( C_{L(n)}^{(0)} \) [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 \( C_{L(n)}^{(0)} \) matrix is used to apply tilt updates to \( C_B^{(0)} \) (Refs. 12, Sec. 6.1.2, and 15, pp. 120–121). Initial tilt alignment corrections to \( C_B^{(0)} \) can be fairly large, e.g., 0.1–1.0 deg, which can produce undesirable errors in \( C_B^{(0)} \) 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 \( C_B^{(0)} \); however, it could leave a residual orthogonality/normality error in the \( C_B^{(0)} \) rows (and columns). The result would be the requirement to include an orthogonality/normality correction algorithm (see Sec. IV.A.3) as an outer loop in the \( C_B^{(0)} \) update processing.

A discrete digital algorithm for the Eq. (50) \( \mathbf{\alpha} \) integral can be constructed by first combining Eqs. (13) and (15) to obtain the \( \omega_{LB}^{(t)} \) integrand and then approximating

\[ \omega_{LB}^{(t)} \approx C_L^{(0)} \left[ \omega_{LB}^{(t)} + \rho \omega_{LB}^{(t)} u_L^{(t)} + Fn \right] \tag{52} \]

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

\[ \mathbf{\alpha} \approx \int_{t_{n-1}}^{t_n} \omega_{LB}^{(t)} \, dt \]

where \( \mathbf{\alpha} \) is the rotation rate vector, the \( \omega_{LB}^{(t)} \) is the computer \( \mathbf{\alpha} \) cycle update period \( t_n - t_{n-1} \) and \( j \) is the number of computer \( \mathbf{\alpha} \) cycles over the \( t_{n-1} \) to \( t_n \) \( \mathbf{\alpha} \) cycle computer update period.

The subscript \( n \) can be expressed in terms of the 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 \( \mathbf{\alpha} \) parameters. For example, a linear extrapolation formula using the last two computed values for \( \mathbf{\alpha} \) would be

\[ \mathbf{\alpha}_{n-1} = \mathbf{\alpha}_{n-1} + \mathbf{\alpha}_{n-2} - \mathbf{\alpha}_{n-3} \tag{55} \]

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

\[ \Delta \mathbf{\alpha}_{LB} \approx \frac{1}{2} (\omega_{LB}^{(t)} + \omega_{LB}^{(t-1)}) T_m \tag{56} \]

where \( T_m \) is the computer \( \mathbf{\alpha} \) cycle update period \( t_n - t_{n-1} \).

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

### 3. Normalization and Orthogonalization

From its basic definition in Sec. II.B, the columns (and rows) of \( C_B \) 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 \( C_B \) update algorithms already described, a normalization and orthogonalization algorithm is frequently included to ensure that the \( C_B \) rows and columns remain normal and orthogonal. Factors that cause \( C_B \) orthogonality/normality error include \( C_B \) orthogonality/normality initialization error, software programming error, roundoff error due to insufficient computer wordlength for the total number of \( C_B \) 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 \( C^p_b \) integration 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 \( C^p_b \) 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 \( C^p_b \) 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 \( B^p_m \) 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) \( q^b_L \) 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 \( C^p_b \) updating algorithm derivation in Sec. IV.A. Thus, using the Eq. (7) attitude quaternion chain rule, we write

\[
q_{B(t)} = q_{B(t-1)} q_{B(t-1)}
\]

\[
q_{B(t)} = q_{B(t)} q_{B(t-1)}
\]

where

\[
q_{B(t)} = q_{B} L \text{ at time } t_{m-1} \text{ to the } L \text{ frame at time } t_{m-1}
\]

\[
q_{B(t)} = q_{B} L \text{ at time } t_{m} \text{ to the } L \text{ frame at time } t_{m}
\]

\[
q_{B(t)} = \text{ attitude quaternion that accounts for } B \text{ frame rotation relative to inertial space from its orientation at time } t_{m-1} \text{ to its orientation at time } t_{m}
\]

\[
q_{I(t)} = \text{ attitude quaternion that accounts for } L \text{ frame rotation relative to inertial space from its orientation at time } t_{m-1} \text{ to its orientation at time } t_{m}
\]

The updates for \( q^b_L \) are performed by \( q_{B(t)} \) and \( q_{I(t)} \) 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 \( q_{B(t)} \) to account for angular rotation \( \omega^b_L \) of the strapdown sensor (body) \( B \) frame relative to nonrotating space. The formal definition for \( q_{B(t)} \) is

\[
q_{B(t)} = q_{B(t-1)} q_{B(t-1)}
\]

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

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

\[
q_{B(t)} = q_{1} + \int_{t_{m-1}}^{t_{m}} q_{B(t-1)} \text{ d}t
\]

#### 2. Local Level (L) Frame Rotation

Equation (58) updates the \( q^b_L \) attitude quaternion using \( q_{L(t)} \) to account for angular rate of the local-level coordinate \( L \) frame relative to nonrotating space \( \omega^L_L \). The formal definition for \( q_{L(t)} \) is

\[
q_{L(t)} = q_{1} + \int_{t_{m-1}}^{t_{m}} q_{L(t-1)} \text{ d}t
\]

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

The \( q_{L(t)} \) attitude quaternion can also be expressed in terms of the rotation vector defining the frame \( L(t-1) \) attitude relative to frame \( L(t-1) \). Applying Eq. (5) with Taylor series expansion for the integral terms yields

\[
q_{L(t)} = \left[ \begin{array}{c} \cos 0.5 \zeta \nonumber \end{array} \right]
\]

\[
q_{L(t)} = \left[ \begin{array}{c} \sin 0.5 \zeta \nonumber \end{array} \right]
\]

\[
q_{L(t)} = 0.5 \zeta
\]

\[
q_{L(t)} = 0.5 \zeta
\]

The negative sign on \( \zeta \) accounts for the opposite phase sense of \( q_{L(t)} \), which describes the frame \( L(t-1) \) attitude relative to frame \( L(t-1) \). The \( \zeta \) rotation vector in Eqs. (62) is identical to \( \zeta \), used for \( C^p_b \) direction cosine matrix updating and is calculated using the identical algorithm provided by Eqs. (35), (41), and (42) or Eqs. (35), (46), and (47).

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

\[
q_{L(t)} = \left( 1 - 0.5 (0.5 \zeta)^2 \right)^{0.5 \zeta}
\]

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

<table>
<thead>
<tr>
<th>Algorithm function</th>
<th>Input</th>
<th>Output</th>
<th>Equation number</th>
</tr>
</thead>
<tbody>
<tr>
<td>High-speed calculations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrated $B$ frame angular rate increments</td>
<td>$\Delta \alpha_l$, $\alpha_l$</td>
<td>$\beta_m$</td>
<td>(41) or (46)</td>
</tr>
<tr>
<td>Coning increment</td>
<td>$\Delta \alpha_l$, $\alpha_l$</td>
<td>$\beta_m$</td>
<td>(42) or (47)</td>
</tr>
<tr>
<td>Normal-speed calculations for Earth related parameters</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$N$ frame Earth rate components</td>
<td>$C^E_N$</td>
<td>$\omega^E_{I_N}$</td>
<td>(14)</td>
</tr>
<tr>
<td>Vertical transport rate component</td>
<td>$C^E_N$</td>
<td>$\rho_{ZN}$</td>
<td>Ref. 12, Sec. 4.6</td>
</tr>
<tr>
<td>Curvature matrix</td>
<td>$C^E_{\xi^l}$, $h$</td>
<td>$F_C$</td>
<td>Ref. 12, Sec. 5.3</td>
</tr>
<tr>
<td>Normal-speed velocity calculations</td>
<td></td>
<td>$v^N$</td>
<td>Part 2 (Ref. 13)</td>
</tr>
<tr>
<td>Normal-speed attitude calculations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$B$ frame rotation vector</td>
<td>$\alpha_m$, $\beta_m$</td>
<td>$\phi_m$</td>
<td>(35)</td>
</tr>
<tr>
<td>$B$ frame rotation matrix (for attitude direction cosine matrix updating)</td>
<td></td>
<td>$C_{B^{(m-1)}}$</td>
<td>(26)</td>
</tr>
<tr>
<td>$B$ frame rotation quaternion (for attitude quaternion updating)</td>
<td></td>
<td>$q_{B^{(m-1)}}$</td>
<td>(60)</td>
</tr>
<tr>
<td>Attitude update for $B$ frame rotation (direction cosine matrix form)</td>
<td></td>
<td>$C_{B^{(m-1)}}$, $C_{B^{(m)}}$</td>
<td>(23)</td>
</tr>
<tr>
<td>Attitude update for $B$ frame rotation (quaternion form)</td>
<td></td>
<td>$q_{B^{(m-1)}}$, $q_{B^{(m)}}$</td>
<td>(57)</td>
</tr>
<tr>
<td>$N$ frame position increment</td>
<td>$\psi^N$</td>
<td>$\Delta R^N_m$</td>
<td>(56)</td>
</tr>
<tr>
<td>$L$ frame rotation vector</td>
<td>$\omega^N_{I_E}$, $\rho_{ZN}$, $F_C$, $\Delta R^N_m$</td>
<td>$C_{L^{(m-1)}}$</td>
<td>(53), (55)</td>
</tr>
<tr>
<td>$L$ frame rotation matrix for attitude direction cosine matrix updating (exact form)</td>
<td></td>
<td>$C_{L^{(m-1)}}$, $C_{L^{(m)}}$</td>
<td>(49)</td>
</tr>
<tr>
<td>$L$ frame quaternion for attitude quaternion updating (exact form)</td>
<td></td>
<td>$q_{L^{(m-1)}}$, $q_{L^{(m)}}$</td>
<td>(62)</td>
</tr>
<tr>
<td>Attitude update for $L$ frame rotation (direction cosine matrix form)</td>
<td></td>
<td>$C_{L^{(m-1)}}$, $C_{L^{(m)}}$</td>
<td>(24)</td>
</tr>
<tr>
<td>Attitude update for $L$ frame rotation (quaternion form)</td>
<td></td>
<td>$q_{L^{(m-1)}}$, $q_{L^{(m)}}$</td>
<td>(58)</td>
</tr>
<tr>
<td>Normalization and orthogonalization corrections (for attitude direction cosine matrix)</td>
<td>$C^L_B$</td>
<td>$C^L_B$</td>
<td>Sec. IV.A.3</td>
</tr>
<tr>
<td>Normalization corrections (for attitude quaternion)</td>
<td>$q^L_B$</td>
<td>$q^L_B$</td>
<td>Sec. IV.B.3</td>
</tr>
<tr>
<td>Normal-speed position calculations</td>
<td></td>
<td>$C^E_N$, $h$</td>
<td>Part 2 (Ref. 13)</td>
</tr>
</tbody>
</table>

Sec. IV.A.3 does not apply. If a quaternion normalization algorithm is to be utilized, it is based on comparing the magnitude of $q^E_I$ with unity and using the variation from unity to iteratively update $q^E_I$ 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 and 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 with exposure to 0.01 g rms wideband random linear input vibration. The linear vibration generates a 0.0003-rad multiaxis 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 e.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 in 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 multiaxis 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 effect error 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 multiaxis 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 sculling 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

14Morse, P. M., and Feshbach, H., Methods of Theoretical Physics, McGraw–Hill, New York, 1953.